On the Efficiency of Packet Telephony

On the Efficiency of Packet Telephony

Mario Baldi[*], Davide Bergamasco[†], and Fulvio Risso[‡]
Dipartimento di Automatica e Informatica
Politecnico di Torino
Corso Duca degli Abruzzi 24
10129 Torino - ITALY

January 6, 1999

Abstract

This paper presents a study on the efficiency of packet switching in providing *toll quality* telephone services. Packet switching is appealing for the implementation of a commercial telephone network because it features lower cost and higher manageability than circuit switching, and enables integration of real-time and non real-time services.

This work compares the real-time efficiency of packet switching and circuit switching, i.e., the volume of voice traffic being guaranteed *deterministic* quality related to the amount of network resources used. For this purpose, we developed a call level simulator which allows a general topology network to be studied. The simulator performs call admission control according to the availability of the resources required to provide a *deterministic* delay bound for each call. Statistical data on accepted and rejected calls are the simulation output.

Results show that packet size—possibly constrained by the protocol in use—is a key factor in determining the real-time efficiency. The packet size which maximizes real-time efficiency is devised analytically.

[*] mbaldi@polito.it
[†] bergamasco@polito.it
[‡] risso@athena.polito.it

1 Introduction

The telephone network, that is currently the widest communication network around the world, is revealing its limits in terms of manageability and economicity. Currently research and experimentation are ongoing to replace the technological heart of the telephone network, namely circuit switching, with the promising packet switching.

Two technologies in the packet switching arena are the main candidates for building a telephone network: ATM and IP. ATM was originally conceived within the telecommunications community and engineered for realizing the B-ISDN, i.e. to carry various kinds of traffic with different requirements in terms of delay, bandwidth, reliability. In particular, since ATM comes from the telecommunications world which traditionally deals with voice, it has being designed taking into account this particularly demanding type of traffic. However, the high claims which drove the standardization of ATM, led to a complex protocol stack, that means complex and expensive network devices.

On the other side, the TCP/IP Protocol Suite originates from a reasearch project of the computer community to provide a best effort service for the exchange of data among computers. The simplicity of the protocol stack and of the network access have been the success of the IP protocol and led to its unforeseen and ever growing diffusion. The protocol stack has not been originally conceived to carry real-time traffic and definitely not to support telephony. However, due to its diffusion, it is now a natural choice to build the B-ISDN and several reasearch groups are now working to enhance the IP protocol with multimedia capabilities. This requires changing the basic service paradigm intended to provide a best effort service introducing more complex handling of packets into routers and resource reservation. The most accredited contributions come from the Integrated Services and the Differentiated Services Working Groups of the IETF.

The first working group has moved towards an "ATM-like" solution based on a signalling protocol called Resource ReserVation Protocol (RSVP) through which end-systems notify the network of their traffic and request a service [1]. The network performs call admission control and possibly reserves resources for new flows in order to provide either a guaranteed service [2] or a controlled load service [3] —the service the flow would get on a lightly loaded best effort network. Still the network provides best effort service to the rest of the traffic and to the flows whose request is denied due to resource unavailability.

The Differentiated Services Working Group [4] is looking for a simpler solution based on the definition of a limited number of classes of service. The approach is more scalable because signaling is not required and routers do not have to classify packets according to the flow they belong. Users pay for a selected class of service (e.g., low delay), but no guarantee is given on the QoS since within the same class the packets of all the users are treated in the same way.

This paper focuses on the substitution of the circuit-switched infrastructure in the present telephone network with a packet-switched one, while keeping unchanged the service level perceived by users. Both ATM and IP are considered for the implementation of the network; in order to be comparable to the high quality standard featured by traditional telephony, the IP based solution must implement the guaranteed service model.

End-systems signal to the network the intention to begin a new phone call —User Network Interface (UNI) signalling and RSVP are used on an ATM and IP network, respectively— and the network accepts or rejects it according to the availability of enough resources to guarantee the required quality. The way in which packets are scheduled into nodes impacts on the amount of resources needed to guarantee the QoS to a call and eventually with the total amount of phone calls the network can carry. A call level simulator has been developed to compare the *resource utilization efficiency* in carrying voice traffic of three packet telephone network architectures with respect to the traditional circuit switched architecture. It is worth highlighting that the definition of efficiency adopted in this paper is relevant for the domain of the problem addressed, namely a network designed to carry mainly (or even exclusively) real-time traffic. Maximizing such efficiency is not necessarily the objective of the designer of a network intedend for a different deployment.

The paper is structured as follows. Section 2 discusses the mechanisms used to provide QoS guarantees on a packet network and the indexes used throughout this paper to quantitatively assess resource utilization efficiency. Section 3 describes the simulator developed for this study used to produce the results shown in Section 4. Conclusive remarks and future lines of research are given in Section 5.

2 Guaranteed Services in Packet Switched Networks

In a packet switched network packets are independent data units which are individually carried from a source to a destination. This implies that:

1. data cannot be sent as a continuos bit stream at a given rate, but it must be segmented into packets which are transmitted from node to node at wire speed;

2. each packet must be labeled with an header containing the information needed by the network to properly handle it (i.e., routing informations such as source and destination addresses, desired service level, etc.)

Thus, with respect to a circuit switched network, a packet switched network introduces both an extra delay (due to the packetization process) and a reduction in bandwidth efficiency (due to the additional header).

In a packet switched network there exist also another source of delay. Since packets can be sent by sources without any specific time relationship (i.e., asynchronously), it may happen that many packets can arrive simultaneously at a particular network node and they have to be retransmitted from the same output port. In this situation packet are buffered and are scheduled for transmission one by one. The time spent by packets into a node buffer waiting for transmission is called queuing delay.

The queuing delay is strongly related to the scheduling algorithm exploited by network nodes. The simplest scheduling mechanisms is called *First In First Out* (FIFO). When an output port exploits FIFO scheduling, packets are transmitted exactly in the same order they arrived at the port.

This algorithm is not suitable for controlling queuing delay because it depends on the time packets get into the buffer relatively to other packets. Since this events are asynchronous, the queuing delay is not deterministic.

This could be acceptable in a data network carrying just best effort traffic but not in network which is also intended to transport time-sensitive traffic such as voice and video. Thus, more sophisticated packet scheduling algorithms have been introduced. One of these algorithms is the *Generalized Processor Sharing* (GPS) [5] algorithm, developed by Pareck and Gallagher starting from an original idea of Nagle [6]. GPS is a scheduling algorithm which allows for both fair sharing of the link bandwidth among different flows and a deterministic bound on their queuing delay. GPS is an ideal algorithm because it is intended to work with fluid flows. However, a packet version of it, *Packet-by-packet GPS* (PGPS), has been devised which works with packet streams preserving almost completely the GPS properties. The next section describes in some detail the two algorithms.

2.1 (Packet-by-packet) Generalized Processor Sharing

The GPS algorithm operates with traffic flows having an infinitely fine granularity. This is referred to as the *fluid flow* model. Each active flow feeds a separate buffer and all the back-logged buffers are served concurrently with a rate proportional to a weight ϕ_i. A GPS scheduler guarantees to each flow i a minimum service rate

$$g_i = \frac{\phi_i}{\sum_j \phi_j} \cdot r, \tag{1}$$

where r is the output rate, usually the output link capacity. The service rate associated with a particular flow is independent of the service rate associated with other flows. $\frac{\phi_i}{\sum_j \phi_j}$ represents the fraction of the link capacity associated with flow i.

Moreover, provided that a flow is compliant with the traffic exiting a leaky bucket with an output rate $B_i < g_i$ and token bucket of depth σ_i, GPS guarantees an upper bound to the queuing delay of each flow i given by

$$Q_i^{GPS} = \frac{\sigma_i}{g_i} = \sigma_i \cdot \frac{\sum_j \phi_j}{\phi_i} \cdot r$$

This bound can be intuitively explained by considering that a burst of σ_i bits generated by a source gets buffered at the access node which "fluidly drains" it at the minimum rate g_i, thus introducing a maximum delay σ_i/g_i. Since all the subsequent nodes on the path to the destination serve the flow at exactly the same rate, no further buffering is required in the network, i.e., no extra delay is added.

Unfortunately, the fluid flow model does not exist. In real networks, traffic flows are made of packet streams with an highly variable granularity. PGPS, developed by Demer, Keshav and Shenkar under the name of *Weighted Fair Queuing* [7], extends GPS in order to handle packet-based flows. The basic idea behind PGPS is quite straightforward: incoming packets are scheduled for transmission according to their equivalent GPS service time, i.e., the instant of time in which the last bit of a packet would be sent by GPS.

Assuming that a packet flow is still compliant with the above leaky bucket (i.e., leak rate B_i and bucket depth σ_i), the queuing delay bound is as follows [8, 2]

$$D_i = \frac{\sigma_i}{g_i} + \frac{(h_i - 1) \cdot L_i}{g_i} + \sum_{m=1}^{h_i} \frac{L_{max}}{r_m} \tag{2}$$

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 link capacity m), L_i is the maximum packet size for flow i and P_M is the maximum packet size allowed in the network.

The delay bound provided by Equation 2 is proportional to the burstiness of the source σ_i and the number of traversed nodes $h_i - 1$, and it is inversely proportional to the weight ϕ_i associated with that source. Thus, when a delay requirement is to be met by a flow i, the higher the burstiness of a source, the larger the weight ϕ_i (relatively to the other weights) must be. In other words, as shown by Equation 1, in order to keep the delay below the required bound the larger the burstiness of a source, the larger the share of link capacity (i.e., the amount of resources) assigned to the flow must be. This suggests that PGPS is better suited to constant bit rate flows.

Moreover, Equation 2 shows that the delay bound depends on the number of traversed nodes. Thus, given a delay requirement for flow i, the larger the number of hops on its path, the larger the service rate necessary to satisfy the delay requirement. Equation 2 is applicable when the same service rate is provided to flow i in each node. For the sake of simplicity, the more general case in which the nodes on the path of flow i provide a different service rates is not considered here. In general, however, the maximum delay introduced by each node in the network is inversely proportional to its service rate and the global maximum delay experienced by packets belonging to flow i is the sum of the delay bound associated with each node.

Equation 2 shows that in certain network conditions the service rate of a flow must be larger than the bandwidth necessary to transmit both the data and the protocol overhead. As discussed later, this "over-requirement" can be seen as an extra overhead with respect to the amount of resources necessary for the transmitting the same data on a circuit switched network.

2.2 Call Admission Control

In a packet switched network designed for carrying telephone calls, two flows are generated for each new call: one from the calling party to the called party, and one in the opposite direction. Both of these flows are characterized by specific QoS requirements in terms of minimum bandwidth required and and maximum delay tolerable.

The PGPS scheduling algorithm can meet such requirements if the fraction of the output link capacity used to serve each flow i is large enough. This can be accomplished by properly assigning the weight ϕ_i to a limited number of flows traversing the same link. A mechanism known as *call admission control* (CAC) can be exploited to carry out this task.

The CAC is mechanism which is in charge of processing the signaling requests generated by sources to place a call and decide whether the network can accept the new flows while guaranteeing the QoS requirements of both the new and the already established flows. When a flow is accepted, the fraction of link capacity assigned to it is said to be *reserved* to the flow.

In summary, the CAC mechanism has to (1) determine the amount of bandwidth that needs to be reserved to the new flows in each node in order to keep the overall delay below the given bound, and (2) check whether the appropriate amount of bandwidth is available at each node traversed by the call. If the latter condition is met, the call is accepted, otherwise it is rejected.

If PGPS is deployed into network nodes, the CAC can use Inequality (2) to devise the minimum amount of bandwidth to be reserved in each node to satisfy the flow QoS requirements [1]. The bound on the delay introduced by the network can be obtained by subtracting from the delay acceptable by the user both the time needed for application level processing (i.e., audio or video compression), and the protocol processing time (including the delay introduced by the packetization process).

The CAC checks whether each node on the call path has an amount of available (i.e., not yet reserved) bandwidth larger than $\max(\rho_i, g_i^*)$, where ρ_i is the bandwidth required for the transmission of the i^{th} flow and g_i^* is the minimum g_i value that satisfies Inequality (2) when D_i is the network delay budget for the call. If enough bandwidth is available, the appropriate amount is reserved to the call on every link traversed. When a call is torn down, the bandwidth previously reserved to it is released.

Inequality (2) shows that calls having the same QoS requirements, but routed along paths with a different number of hops, require a different amount of bandwidth to be reserved. In particular, the higher the number of hops, the larger the bandwidth which must be reserved. Thus, in general, the same QoS requirements do not necessarily lead the same resource reservation.

At a first glance, the proposed approach seems to work only with connection oriented protocols. Actually, whenever QoS guarantees are required, an application must ask the network for reserving resources through some sort of signaling protocol, even though the service provided is connectionless. This reservation installs into the network nodes some state information concerning the flow generated by the application which is equivalent to the state information stored for a call in a connection oriented network; the main difference is that in the connection-less network the reservation is usually handled *soft-state*, i.e. it has to be renewed periodically.

2.3 Evaluating the Efficiency of Guaranteed Services over Packet Networks

Since the goal of this work is to study toll quality telephony on a packet switched network, the quality perceived by a user cannot be used as a comparison index because, obviously, it must be almost the same as with a circuit switched network. We compare the two technologies according to the efficiency in carrying real-time and best effort traffic; the efficiency relates the traffic carried to the amount of resources employed.

Considering a given amount of network resources, efficiency can be viewed from two different perspectives:

1. *real-time efficiency* takes into account the amount of real-time traffic

[1]If the CAC is based on Inequality (2), it considers all the nodes on the call path to contribute equally to the whole delay. A more flexible resource allocation can be achieved by considering that each node can contribute differently, and thus reserve resources according to its local availability.

carried by the network;

2. *transport efficiency* refers to the overall amount of traffic carried by the network.

The former focuses on the capability of the network to carry real-time traffic, without careing of other kinds of traffic. Thus, it is relevant when the network is intended to carry mainly real-time traffic, like a commercial telephone network. The latter is relevant when a significant part of the traffic is to be best effort and the provision of the corresponding service is not a marginal issue.

We define a set of four efficiency parameters that are independent from the two definition above and can be used to compare the efficiency of packet switching and circuit switching:

1. The *effective load* is the data rate at the application level. The effective load does not account for the protocol overhead, i.e., it is the bandwidth that would be required to send the data on a circuit switched network.

2. The *real load* is the raw link capacity used by user data; it corresponds to the effective load augmented by the overhead introduced by the various protocol layers.

3. The *apparent load* is the bandwidth reserved to the phone calls (more in general to the real-time sessions) in order to meet their QoS requirements.

4. The *call blocking probability* is the ratio between the number of calls rejected and the total number of calls offered to the network.

The effective load gives an idea of the amount of real-time traffic carried by the network. Comparing effective and real load gives an insight in the transport efficiency, while the comparison between the effective and apparent load shows the real-time efficiency. The call blocking probability can be used by a carrier to engineer the network with packet switched technology.

Throughout the paper the terms effective, real and apparent *bandwidth* are used to refer to the actual data rate, the overall transmission capacity required, and the bandwidth to be reserved, respectively, to a single call in order to meet its QoS requirements. These are indexes of how effectively calls with such characteristics can be carried by the network. For example, the lower the apparent bandwidth of a call, the higher is the amount of such calls the network can carry; while the larger the real bandwidth, the higher the amount of raw transmission capacity required.

3 The Simulation environment

In order to study the feasability of a packet switched telphone network based on the PGPS mechanism, we exploited a simulative approach. In particular, we developed a C++ simulator capable to model a packet switched network composed of nodes, links and users arranged in an arbitrary topology.

Nodes are connected among them by full duplex links in order to form the network. Users are connected to ingress nodes and generate calls with either deterministically or stochastically distributed interarrival times. The destinations of these calls can be chosen on a deterministic or stochastic basis too.

When a source generates a call[2], a *routing module* selects an appropriate forwarding path for the packets belonging to the session[3]. Every node along the path from the source to the destination checks if it has enough free resources to satisfy the call QoS requirements. The amount of resource needed is determined according to a CAC rule which depends both on the scheduling algorithm exploited by network nodes and on the QoS guarantees to be provided to the call. If all the nodes have enough resources to handle the call, it can be accepted; otherwhise it is blocked. A statistical module records the outcome of each call and evaluates both the call blocking probability experienced by each source and the average link utilization.

The rest of this section describes in more detail the simulator modules. The simulation scenario is also introduced.

3.1 Call duration model

Telephone networks are usually dimensioned by considering that the average phone call has a duration of about 3 minutes and the call inter-arrival

[2]If the simulator is used to study a connectionless network, the call can be thought as a real-time session with an associated destination, starting time, and duration. The network is asked to reserve enough resources for the call to provide the required QoS.

[3]At present, the routing module is just able to chose forwaring paths among a set of preconfigured static routes. In future, its capabilities will be improved in order to support also dinamic call routing.

times are exponentially distributed (i.e., phone calls are modeled as a Poisson process). This extremely simple model was devised in the early days of telephone communications and it has been used for almost a century. Lastly, such a model is not a realistic representation of phone calls any more because of new and different traffic patterns. Bolotin [9] proposes a more accurate model in which the call duration is distributed according to a probability distribution obtained by the weighted composition of 3 functions:

$$F(x) = w_s \cdot F_s(x) + (1 - w_s) \cdot [\alpha \cdot F_1(x) + (1 - \alpha) \cdot F_2(x)]$$

$F_s(x)$, weighted from 1 to 3%, takes into account short calls (shorter than 3 seconds). Even though the real probability distribution of short calls is quite complex, $F_s(x)$ approximates it with a uniform probability distribution. $F_1(x)$ and $F_2(x)$ are Gaussian logarithmic distributions and take into account the contributions due to the other types of calls (generated by both residential and business users). Figure 1 shows the probability density of the duration of calls generated by the simulator according to this model; the component probability densities $f_1(x)$ and $f_2(x)$ as produced by the simulator are also plotted.

3.2 Voice Encoding

The bandwidth required by a phone conversation depends essentially on the encoding scheme exploited. Our simulator encompasses five types of encoding techniques:

1. *Pulse Code Modulation* (PCM) is the encoding scheme traditionally used in digital telephone networks. The voice signal is sampled every 125μ and each resulting sample is encode on 8 bits using a non linear compression law. As a result a PCM encoder produces a CBR flow at 64 Kb/s.

2. *ADaptive PCM* (ADPCM) encoders are based on the so called *differential encoding* which exploits the temporal redundancy intrinsecally present in the voice signal to reduce the bit rate of the encoded flow. The ITU-T Recommendations G.726 and G.727 specify the *Embedded ADPCM* encoding for output rates of 40, 32, 24, and 16 Kb/s. Our simulator implements ADPCM32 sources.

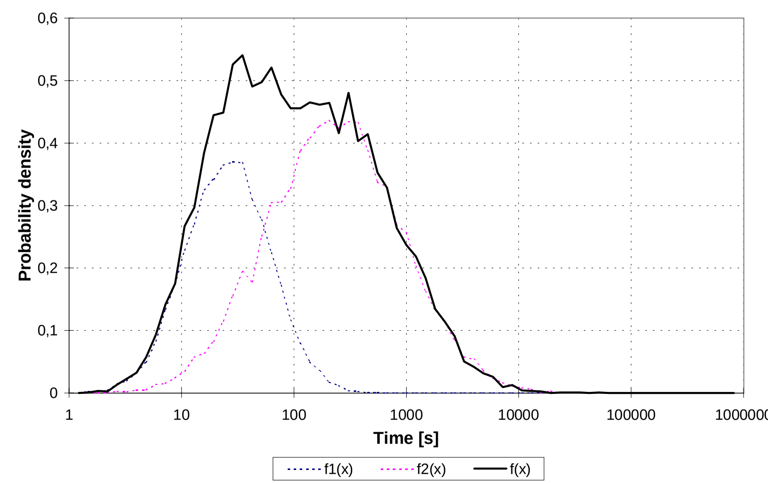

Figure 1: Probability density of call duration as generated by the simulator.

3. *Linear Predictive Coding* (LPC). Differently of the other techniques, an LPC encoder does not encode the voice signal but a set of parameters that represente it. First the voice signal is partitioned in a sequence of 10 to 30 ms long segments. Each segment is then approximated by a linear system having in input a predefined signal. The system is characterized by a set of parameters (from 8 to 12) and a gain constant which are extracted from the voice segment being encoded. These coefficients are finally encoded and trasmitted to the decoder. The decoder uses the parameters received from the encoder to configure a linear system which is fed with the same predefined signal used during the encoding phase in order to obtain a reconstructed voice segment. The bandwith of CBR flow generated by an LCP encoder is about 2.4 Kb/s to 9.6 Kb/s.

4. *Global System for Mobile communications* (GSM) encoding. GSM encoders sample the voice signal at 8 kHz and digitalize each sample on 13 bits. Each group of 160 words (containing 20 ms of voice signal) is independently compressed to 260 bit (i.e. with a compression ration of 1:8) thus producing a 13 Kb/s CBR flow.

5. *Code-Excited Linear Predictive* (CELP) encoding. CELP encoders also devise the parameters of a model of fixed duration segments of the vocal signal. They produce a CBR flow at 4 Kb/s to 16 Kb/s.

Our simulator permits the use of different encoding schemes simultaneously, thus allowing to simulate different kinds of traffic. Once chosen both a particular encoding scheme chosen (i.e., on the bit of the encoded signal) and the maximum tolerable packetization delay, the size of packets generated by souces is determined accordingly. The encoding scheme determines the effective load generated by each call while the real load is determined adding the protocol overhead.

Traffic sources are also characterized by parameters such as the average call inter-arrival time (whose probability distribution can be set to be either a Poisson or a Gauss one) and the average call duration time.

3.3 Link model and Protocol Stack

Currently, the most common physical layer transports in the telecommunications area are the *Plesiochronous Digital Hierachy* (PDH) and the *Synchronous Digital Hierachy* (SDH) hierarchies. We consider these technologies as the basis (the physical layer transport) for both circuit switching and packet switching.

As far as the packet switching technique is concerned, at present there seems to exist just two candidates: the *Internet Protocol* and *Asynchronous Transfer Mode*. IP is the tecnology with the fastest growing pace, while ATM is the one more largely adopted by public carriers. Anyway, in a packet switched telephone network context, this two techniques are not mutually exclusive. At least two scenarios where they can coexist are envisageble:

1. ATM is used as a data link technology to connect IP routers. Both IP routers and ATM switches contribute to the delay experienced by samples which are buffered both inside routers (as IP packets) and inside switches (as cells containing chunks of IP packets).

2. ATM is used to provide end-to-end connectivity while IP is used to provide support the vast amount of applications which exploit the Internet Protocol.

From the point of view of the resource utilization efficiency, the first scenario is very close to the case where IP is used directly over SDH/PDH, except for the (nearly constant) protocol overhead due to the ATM layer. The second scenario, indeed, provides a different efficiency depending on the type of traffic. In fact, in the case of data traffic the efficiency is the same as the first scenario, while in the case of voice traffic the efficiency is larger because voice samples are carried directly into ATM cells.

The purpose of the simulation study is to assess the efficiency provided by the different architectural choices shown in Figure 2 in order to determine how they compare with respect traditional circuit switched telephony.

3.4 Call Admission Control

The CAC mechanism must first determine the amount of resources needed by an incoming call. The apparent load generated by the generica call i is calculated by solving for g_i the following inequality which is derived from Equation 2 by simply adding the propagation delay $Dprop$ and the packetization delay $Dpack$:

$$D_{req} \geq Dpack + Dprop_0 + \frac{\sigma_i + (H-1) \cdot L_i}{g_i} + \sum_{m=1}^{H} (\frac{L_{max}}{r_m} + Dprop_m) \quad (3)$$

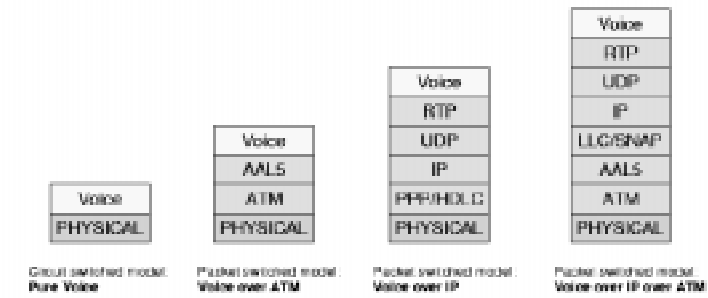

Figure 2: Protocol stacks used in the simulations.

where $Dprop_0$ is the propagation delay of the link from the source to the first node and $Dprop_m$ is the propagation delay on link m. The processing delay to encode and decode the voice signal is not considered here because it depends on the application and the actual encoding scheme. Here only network related delays are taken into account. Sources are supposed to send packets as soon as they have gathered enough samples to fill them. Since we consider only CBR encoders, this results in having sources sending fixed size packets at constant intervals. Moreover, since only omogeneous sources have been considered, L_i is the same for any flow i; since constant bit rate sources are used, $\sigma_i = L_i$ for all packet switching architectures with the exception of IP over ATM. In this case, L_i is the size of ATM cells switched by network nodes, while σ_i is the size of the IP packets sent by the source. L_{max} has been set to 1500 bytes that is the Maximim Transmission Unit for Ethernet networks and can be used by best effort sources.

If we call e_i the real load of call i obtained by increasing the effective load by the protocol overhead, and g_i^* the minimum g_i which satisfies Inequality (3), the apparent load of call i is given by $\max\{e_i, g_i^*\}$. The CAC accepts a call if the sum of the apparent load of all the calls (included call i itself) routed on the links traversed by call i does not exceed the link capacity. So far, this is the only CAC rule implemented in the simulator.

3.5 Statistical module

The efficiency indexes described in Section 2.3 are measured by gathering data during the simulation. A *statistical module* is embedded in the simulator to ensure that the numerical results obtained have statistical relevance.

Data gathered during the initial part of the simulation, called the *initial transient phase*, should not be considered in the evaluation of the performance indexes. In fact, when a simulation run starts, the network is idle, i.e., all the resources are available and all the calls are consequently accepted. Then, as far as calls are generated by users, the network utilization increases and reaches a stable level. The performance indexes should reflect the network situation in this *stable phase*.

Each gathered sample is passed to the statistical module which identifies the end of the initial transient phase. Given an index under statistical evaluation, the statistical module computes the mean over a predefined number n of samples. The transient phase is considered over if the relative difference between each of the last N means and their mean is smaller than a predefined threshold ϵ. The parameters that determine the completion of the transient phase must be chosen empirically. When the transient phase is considered over, the statistical module discards all the data collected in the meantime.

Analogously, the statistical module determines when the simulation can be stopped since the performance indexes have reached a steady state and are not going to change significantly. This is considered to happen when the *confidence interval* of the samples is smaller than a predefined threshold. The confidence interval is the range in which the average over a fixed number of samples falls with given probability. The confidence interval is computed using the *central limit theorem* which states that the average of n samples of a stochastic variable with mean μ and variance σ^2, has a Gaussian probability distribution with the same mean and with variance σ^2/n.

3.6 Network model

We consider a network model in which each telephone is connected to a local exchange through the subscriber loop and local exchanges are in charge of encoding and packetizating. The internal part of the network, i.e., the mash of local offices and toll offices shown in Figure 3, is built by packet switching nodes.

Since local exchanges are not supposed to perform any packet switching function, it is possible to consider them as call sources rather that the individual customers' phone sets.

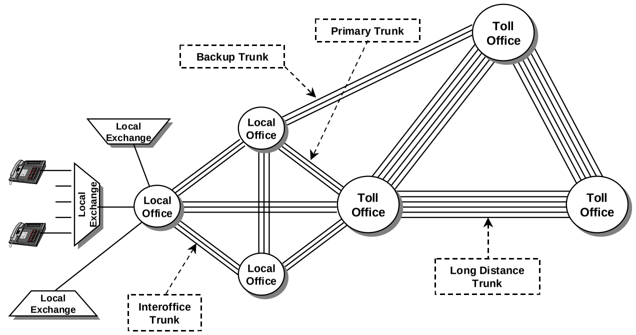

Figure 3: Excerpt from the Topology of a Circuit Switched Telephone Network.

4 Simulation Results

The network topology used in the simulations (see Figure 4) has been modeled after the actual Telecom Italia's telephone network in order to reproduce a quite realistic test environment.

Local and toll offices have been replaced by routers whereas local exchanges have been upgraded with the packetization functionality. The physical length and capacity of links are the same as the real phone network.

The typical domestic long distance call over the Telecom Italia's network crosses at most two local offices and two toll offices. In the scenario depicted in Figure 4, two long distance calls are originated from two different areas (local offices LO_1 and LO_3) toward the same area (LO_2).

Since the packetization process is carried out by local exchanges, from the simulation standpoint they can be assumed as the endpoints of phone calls. These are originated from each local exchange connected to LO_1 and LO_3 and are directed towards every local exchange connected to LO_2. Unless specified differently, the ADPCM32 coding scheme is exploited.

Simulations have been run with an increasing offered load in terms of calls per hour in order to determine the maximum achievable utilization of link $\mathrm{TO}_2 - \mathrm{LO}_2$. Since the actual offered load depends both on the calls duration and frequency, in the rest of the paper it is expressed using a measurement unit known as *Erlang*. The Erlang, i.e., call frequency times average call duration, is the typical measurement unit used in telephony to quantity the offered load.

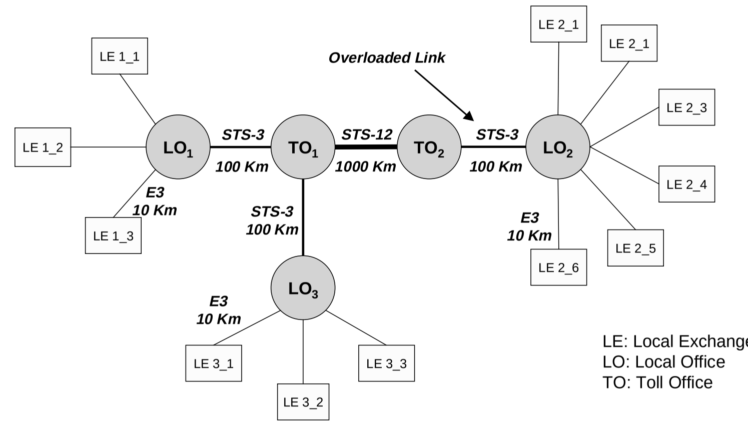

Figure 4: Network topology used in the simulation.

As explained in Section 3.4, the bound on the delay introduced by the network on the packets belonging to a flow depends on the apparent load of that flow. Given the maximun end-to-end delay constraint of 100 ms, the amount of bandwidth that must be reserved to a flow, called the *apparent bandwidth*, can happen to be larger than the minimum amount of bandwidth required to actually transmit the data, referred to as the *real load*. The rest of this section is devoted to identifying the factors that affect the efficiency of packet switching and to determine the trade-off between real-time efficiency and traffic efficiency, whenever possible. Section 4.1 analyzes the difference between apparent and real load on the network and highlight the consequences on network utilization efficiency. Section 4.2 studies the impact of packetization on bandwidth allocation, whereas Section 4.3 discusses the limitation of the current circuit switching technology for transporting compressed voice. Lastly, Section 4.4 explains how to determine the packet size which maximizes real-time efficiency.

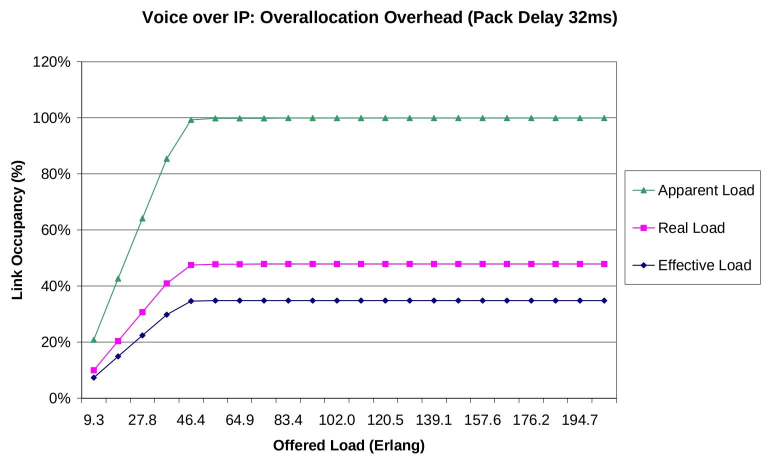

Figure 5: Efficiency indexes on link $TO_2 - LO_2$, with high packetization delay.

4.1 Bandwidth Over-allocation

Figure 5 shows the effective, real and apparent load on link $TO_2 - LO_2$ as a percentage of the link capacity[4]. Voice samples are carried into IP packets transmitted over Plesiochronous Digital Hierarchy (PDH) and Sinchronous Optical NETwork/Synchronous Digital Hierarchy (SONET/SDH) links at various speeds. The packet payload size has been chosen to be 128 bytes, which leads to a packetization delay of 32 ms.

In the leftmost part of the plot, the three loads increase linearly as the traffic offered to the network increases. This means that all the calls are accepted. When the offered traffic becomes large enough to saturate the bottleneck link (i.e., the apparent load reaches 100% of the bottleneck link capacity), the three loads curves flatten, indicating that part of the incoming calls are rejected bye the CAC. The flat part of the curves represents the maximum link utilization achievable in this scenario.

[4]Throughout the paper we often refer to the load on link $TO_2 - LO_2$ as the load on the network. This is motivated by the fact that being $TO_2 - LO_2$ the potential bottleneck link of the reference topology, its utilization is a good representative of the overall load on the network.

The difference between the apparent load and the real load curves is the *bandwidth over-allocation* performed by the CAC based on according to Inequality 3. The over-allocated bandwidth cannot be used to accept further time-sensitive flows on the network because, otherwhise, this would increase the end-to-end delay experienced by telephone calls. However, it can be used to transmit best effort traffic.

The effective load curve allows to compare the packet switched telephone network with the circuit switched one from the efficiency standpoint. Given a call traffic offered to the network, the effective load represents the fraction of link bandwidth that circuit switching would require to carry the same number of phone calls accepted by the packet switched network. The difference between the real load and the effective load curves represent the amount of bandwidth wasted to carry the protocol overhead, i.e., packet headers. The difference between the apparent load and the effective load curves shows how the circuit and packet switched telephone network compare from the real-time efficiency point of view. For example, Figure 5 shows that the same numer of phone calls carried on link $TO_2 - LO_2$ using packet switching can be carried with just approximately 35% of the capacity using circuit switching. In other words, the real-time efficiency of the packet switched telephone network is about one third of the efficiency of the corresponding circuit switched network.

As shown by Figure 5 (and other figures shown later) packet switched telephony is always worse than circuit switching from the real-time efficiency standpoint. The difference between effective and real load (i.e., the overhead due to protocol headers) is unavoidable and can be considered as the fee to be paid in order to exploit "inexpensive" packet switching equipment in place of "costly" circuit switching devices. On the other hand, the real-time efficiency reduction due to the difference between real and apparent load (i.e., the bandwidth overallocation), is less obvious to understand. However, it plays key role since, as shown by Figure 5, overallocation has a significantly stronger impact on real-time efficiency than protocol overhead. Moreover, bandwidth overallocation and protocol overhead are tighly coupled, as shown in the next section.

4.2 Packetization

As stated earlier, the packet payload size affects the overallocation, i.e. the difference between real and apparent load, while the header size (which depends on the particular packet technology deployed) affects the difference

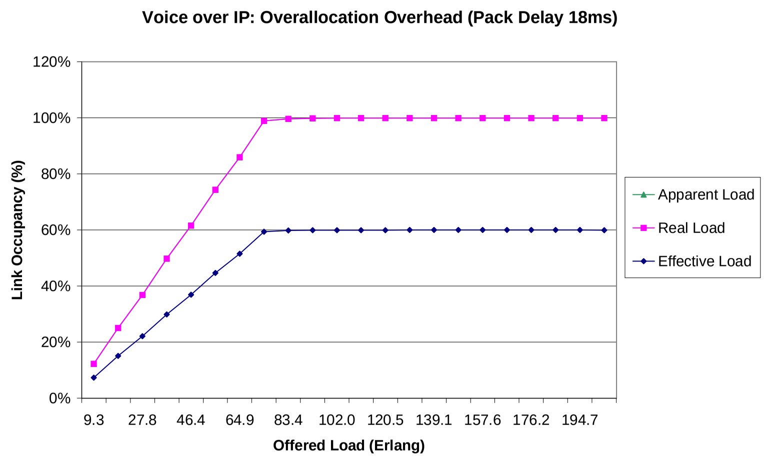

Figure 6: Efficiency indexes on link $TO_2 - LO_2$, with low packetization delay.

between the effective and real loads. This section presents a quantitative assessment of the impact of the two parameters on the real-time network utilization efficiency.

4.2.1 The Payload

A large packet payload allows to minimize the effect of protocol overhead, but requires the transmitter to collect a considerable number of samples before sending a packet. This introduces a significant packetization delay which, in order to meet the delay requirement D_{req}, must be compensated by increasing the the service rate g_i (see Inequality 3). In other words, a larger packet payload translates in a larger apparent load on the network.

This behaviour can be observed by comparing Figure 5 and Figure 6 which show the utilization of link $TO_2 - LO_2$ on an IP network with a packetization delay of 32 ms and 18 ms, respectively. The bandwidth over-allocation —i.e., the distance between the apparent and real load curves— is larger in the former case.

On the other hand, a shorter packetization delay implies a larger relative overhead (since the PPP/IP/UDP/RTP headers have a constant lenght).

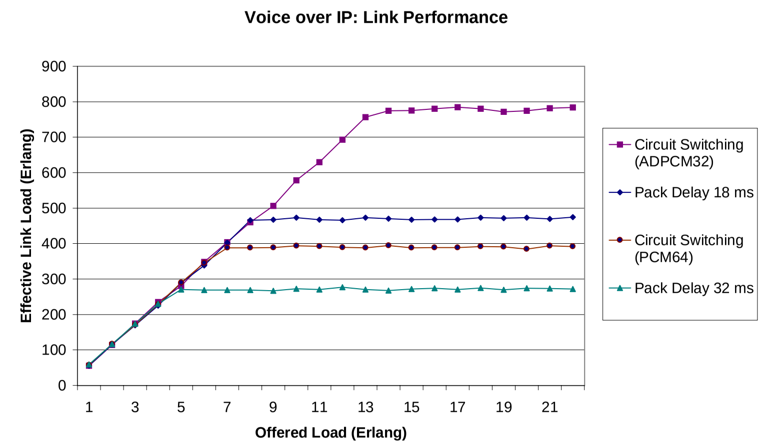

Figure 7: Impact of packet size over the efficiency: effective load.

This can be noted by observing that in Figure 6 the distance between the real and effective load curves is larger that in Figure 5.

A large payload is expected to provide an higher efficiency. However, due to the bandwidth over-allocation, the efficiency is higher when smaller packets are used, as shown by Figure 6 where the effective load is larger than Figure 5. This is confirmed by Figure 7 which plots the effective load (given in Erlang) versus the offered traffic for both a circuit switched and a packet switched network and two different packetization delays (18 ms and 32 ms) As far as maximum volume of traffic accepted on the network, circuit switching outperforms packet switching as expected. As far as the packetization delay is concerned, the 32 ms alternative is the least efficient.

However, there is another issue to consider, namely the amount of network resources available to carry best effort traffic. When a 18 ms packetization delay is exploited, almost all the capacity is used as the number of accepted calls on link $TO_2 - LO_2$ reaches the maximum (note that the apparent load and the real load curves in Figure 6 are overlapped). Instead, when a 32 ms packetization delay is used, the number of phone calls accepted is smaller, but a large fraction of the link capacity is still available to carry best effort traffic (note the significant distance between the apparent

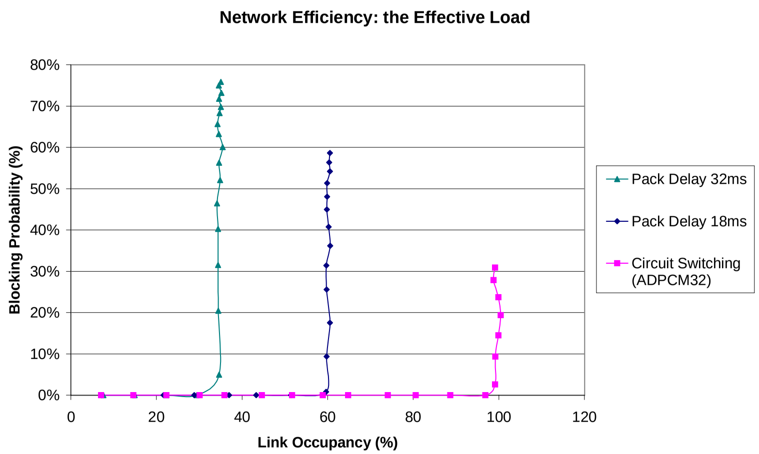

Figure 8: Blocking probability and effective load.

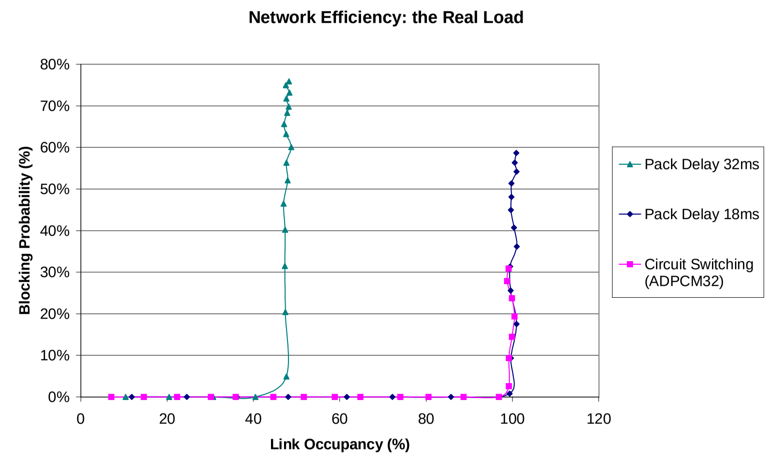

Figure 9: Blocking probability and real load.

load and the real load curves in Figure 5).

The latter situation is the most desirable on a network where the fraction of real-time traffic is small compared to the fraction of best effort traffic (e.g., the Internet and most of today's intranets). This situation is also beneficial on a network engineered to carry mainly real-time traffic but also significant amounts of best effort traffic. In this case, even whether the real-time efficiency is smaller, the overall transport efficiency is significantly large.

The network efficiency can also be analized by studying the call blocking probability versus the effective and real load on the network. A plot of the blocking probability versus the effective load shows the amount of real-time traffic the network is able to accept before rejecting any calls. Instead, a plot of the blocking probability versus the real load shows the network utilization achieved when calls start being rejected. Figure 8 and Figure 9 show the blocking probability of the calls traversing link $TO_2 - LO_2$ versus the effective and real load on that link respectively.

In both of the charts three curves are shown: the first one correspons to a packetization delay of 18 ms, the second one corresponds to a packetization delay of 32 ms, and the third one refers to a circuit switched network. Since the traffic is homogeneous —all the calls require the same bandwidth and delay bound— the blocking probability stays null until the apparent load reaches the link capacity. Then, it jumps to 100% because all incoming calls are rejected. When a call is cleared, a new one is accepted and thus both the effective and real loads remain constant.

Again, Figure 8 shows that circuit switching outperforms packet switching since the effective load reached before calls start to be rejected is the whole link capacity. Instead, packet switching with a packetization delay of 18 ms reaches an effective load slightly larger than half of the link capacity, i.e. it carries half the number of phone calls that would be carried using circuit switching on the same link. A 32 ms packetization delay leads to even lower efficiency. However, Figure 9 shows that using a 18 ms packetization delay leads real-time traffic to use the total link capacity. No extra capacity is left for the transmission of best effort traffic. Thus, if the network is intended to carry comparable fractions of real-time and best effort traffic, a short packetization delay is not necessarily the best solution. In fact, the 18 ms packetization delay allows for high real-time efficiency, but the transport efficiency is considerably low.

4.2.2 The Header

The header size depends on the protocol architecture exploited in the network. In this section we study the effect of varying the packetization delay with different protocol architectures.

Figure 10 shows a plot of the real bandwidth required by an ADPCM32 phone call versus the packetization delay (i.e., the size of the packet payload) for different network technologies. The real bandwidth required on a circuit switched network by both an ADPCM32 and a PCM call are plotted as well[5].

The real bandwidth on an IP network decreases as the packetization delay (and thus the payload size) decreases. This is because of the fixed IP header size. The real bandwidth required by a phone call in an ATM network is smaller than in an IP network because of smaller cell header overhead, and moreover, the packetization delay is considerably smaller (e.g., when ADPCM32 encoding is exploited, about 10 ms are required to fill up a cell payload). When IP packets are encapsulated into ATM cells, the real bandwidth tends to decrease, but discontinuosly. This is due to the fact that when the IP payload size is increased, the IP packet size sometimes exceeds the size of an integral number of cell payloads, so a new cell is needed to carry a fragment of the packet. The real bandwidth is anyway larger than when IP routers are connected directly by SONET/SDH links.

Figure 10 shows that if the packet size is chosen in such a way that the overhead introduced by the header is small enough, a phone call in a packet network can require less bandiwidth then in a circuit switched network exploiting PCM encoding. This means that if the delay requirement is not too tight, the efficiency in a packet telephone network can be larger than in traditional telephone network. This fact is furtherly confirmed by Figure 11 which depicts the apparent bandwidth needed to meet the 100 ms end-to-end delay bound versus various packetization delays with different technologies.

By comparing Figure 10 and Figure 11, it can be noticed that the real bandwidth and the apparent bandwidth are the same. This means that in this scenario no bandwidth overallocation is needed for the various packet technologies except IP. On an IP network, as the packetization delay increases, the delay budget left to queuing shrinks and over-allocation is possibly required in order to keep the end-to-end delay below the bound. Thus,

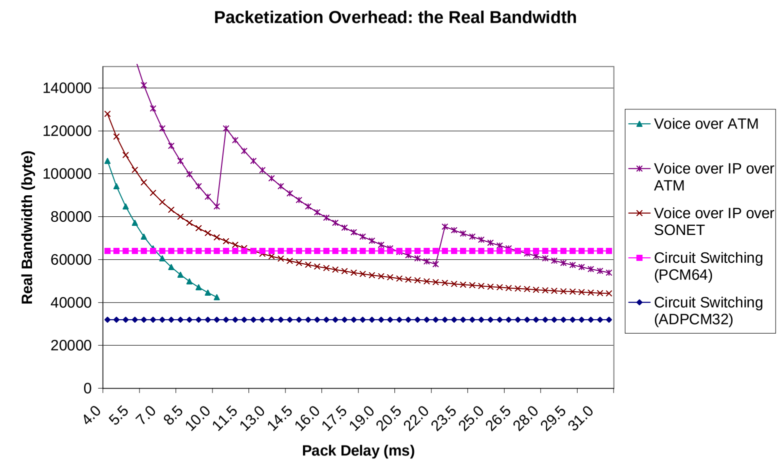

Figure 10: Impact of packetization delay over the real bandwidth of a phone call with various technologies.

there exists an optimal packet size which, by providing minimum apparent bandwidth for a call, maximizes the efficiency of IP telephony.

Viceversa, IP over ATM provides higher efficiency (lower apparent bandwidth) than IP over SONET/SDH for long packetization delays. This stems from the fact that the IP payload size is large enough to generate a low real load, but no bandwidth overallocation is required because the queuing delay experienced by ATM cells in the network is short due to the small cell size.

Among the various packet technologies, ATM is the one characterized by the smallest apparent bandwidth because (1) no overallocation is required due to the low packetization and queuing delay, and (2) the real load is low due to the small cell header.

In general, all the packet technologies require bandwidth overallocation if the user's delay bound is so tight that it cannot be met by allocating the real bandwidth. This is not evident from Figure 11 just because given the network topology and the required delay bound, overallocation is not required[6]. When overallocation is required, the optimal number of samples

[5]In a circuit switched network, the real, apparent and effective bandwidth are coinciding

[6]ATM technology, thanks to the small size of cells, requires over-allocation only in

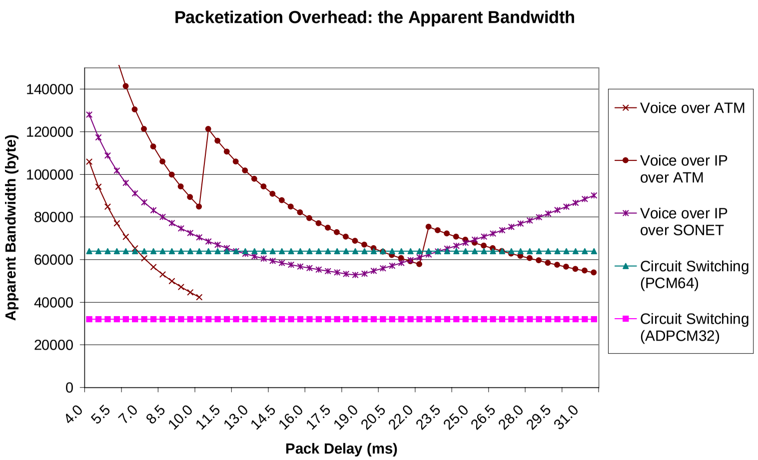

Figure 11: Impact of packetization delay over the apparent bandwidth of a phone call with various technologies.

per packet is a tradeoff between low overhead (i.e., small real load) and low apparent load. If the network is intended to carry also a significant amount of best effort traffic, the former is to be preferred, otherwise the latter is the primary objective.

4.3 SONET/SDH and Voice Compression

Comparing PCM voice calls over circuit switched networks with ADPCM32 calls over packet switched networks seems to be unfair. If a more effective coding scheme, such as ADPCM32, is to be exploited in a packet switched network in order to achieve better efficiency, it could also be used in a circuit switched network leading to even better efficiency. However this is not possible because of the granularity of SONET/SDH channels.

SONET/SDH is the technology intended to replace the old PDH in building circuit switched networks. The main advantage of SONET/SDH over PDH is the possibility to directly isolate a single channel from a carrier of any level of the hierarchy. Only the previous hierarchical level contributors can be derived from a given PDH flow. Thus, isolating a voice call carried over an E2 link (8.192 Mbps) requires first to obtain the four E1 contributors (2.048 Mbps each), then to select the desired channel within the right E1 carrier. SONET/SDH allows a single phone call to be demultiplexed directly from a carrier of whatever level, e.g., from a 155Mbps STM-3 flow.

SONET/SDH, like PDH, assumes a minimum channel capacity of 64 Kbps which makes meaningless the exploitation of voice codecs at lower bit rates, unless multiple phone calls are carried within a single channel. This increases the complexity of the system and prevents the possibility of directly isolating a single phone call from a flow of a higher hierarchical layer.

4.4 The Optimal IP Packet Size

This section analyzed the delay bound formula used to drive the CAC in order to determine an optimal packet size, i.e., the packet size which maximizes real-time efficiency. It is worth to recall that the lower the apparent bandwidth of a voice call, the larger the amount of calls acceptable on the network, and thus the higher the real-time efficiency.

For low packetization delays the apparent bandwidth is equal to the real bandwidth because no bandwith overallocation is required[7]. Increasing the packetization delay, decreases the delay budget left for the network delay and thus bandwidth overallocation is required. As a consequence, the optimal packetization delay is the one for which the allocation of the real bandwith provides exactly the required delay. The real bandwidth is given by:

$$b_{real} = (P_{Headers} + DataRate \cdot D_{pack}) \cdot \frac{1}{D_{pack}} \quad (4)$$

The packet size can be expressed as a function of the packetization delay:

$$L_i = P_{Headers} + DataRate \cdot D_{pack} \quad (5)$$

extreme circumstances, i.e. when the delay required is very tight or when the number of switches on the path of a call is quite large.

[7] If the delay bound required is very tight, then the apparent bandwidth is larger than the real one, no matter how short is the packetization delay. This happens even when delivering empty packets without overallocation requires a time longer than the delay requirement: any packet size requires overallocation and looking for an optimal packet size is meaningless.

Since voice is encoded at constant bit rate burstiness is minimum, i.e. $\sigma_i = L_i$. The optimal packetization delay is the one which, replaced in the second memeber of Inequality 3 with $g_i = b_{real}$ and L_i above, provides a delay bound exactly equal to the required delay $Dreq$. Solving the resulting equation for $Dpack$ we obtain

$$Dpack = \frac{Dreq - Dprop_0 - \sum_{m=1}^{H} \left(\frac{L_{max}}{r_m} + Dprop_m\right)}{H+1} \tag{6}$$

The optimal packet size depends on many parameters. However, a rough estimate can be obtained by an approximated formula which does not take into account some of the delay components in (6):

$$Dpack \simeq \frac{Dreq}{H+1} \tag{7}$$

This equation is a good approximation when the links have high capacity and the network is not very extensive. In our network we are interested in the path between LE_{1_1} - LE_{2_2}: given an end-to-end delay requirement of 100 ms for a call between between LE_{1_1} and LE_{2_2} (i.e., a long distance call), (6) gives an optimal packetization delay $Dpack = 18.7$ ms, while (7) provides $Dpack = 20$ ms. Figure 12, shows the effective load on the IP network versus the packet size, for various level of call traffic offered to the network. The packetization delay which maximizes the effective bandwidth is 18 ms, providing a confirmation of the goodness of the (6) equation.

The number of hops traversed by a phone call is a key factor in determining the optimal packet size, even with the approximate formula in (7). As a consequence, there is no a single optimal packet size for any given network but it must be determined on a call by call basis.

Usually, application programs are not aware of the number of nodes traversed by the traffic they generate. However, the network knows this information at the time it chooses a route towards the destination and possibly reserves resources for the call. The signalling protocol used for resource request and grant could also be used by the network to provide the application with the number of hops or a suggested packet size for the call. This mechanism would give the opportunity to the network provider to dynamically change the traffic mix on its network and adjust the packet size accordingly. If the network is intended to carry a large amount of best effort traffic, a packet size providing a lower effective load will be suggested to the users instead of the optimal one. The efficiency of the network in

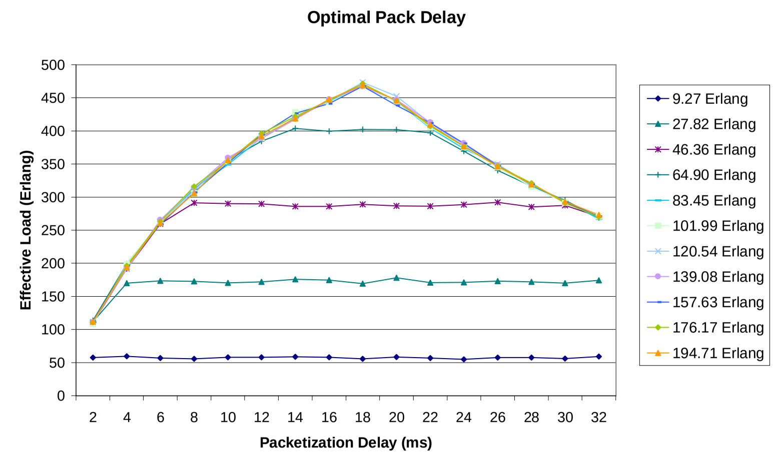

Figure 12: Impact of Packetization Delay on the link efficiency.

carrying real-time traffic will be lower, but the bandwdth available to best effort traffic will be larger.

5 Discussion

This work studies the efficiency of providing toll quality telephony on packet switched networks. When the network is intended to carry mainly real-time traffic, as it will likely be the case in a large scale commercial telephone network, maximizing the efficiency in carrying real-time traffic is crucial. This parameter, called real-time efficiency, has been investigated for various packet technologies and compared to the one of circuit swithing.

A call level simulator has been used across this study. It enables the simulation of a number of call sources generating calls according to specified probability distributions for call arrival and duration over a general topology network. The simulator performs call admission control according to the availability of the resources required to provide a *deterministic* delay bound to each call. The needed amount of resources is determined assuming that the Packet-by-Packet Generalized Processor Sharing queue management scheme is implemented in network nodes. Statistical data on accepted and rejected calls are the simulation output.

In order to simplify the interpretation of the results, simulations have been performed on simple topologies which are however representative of the structure large scale telephony networks usually have nowadays.

Some of the results of this paper could have been devised analytically by calculating the apparent bandwidth of a call and then figuring out how many calls could be accepted on a link, along the line of what is done in [10]. Similarly, the blocking probability could have been devised through the Erlang-B formula. Neverteless, this analytical approach does not apply to etherogeneous call traffic.

The main conclusion we can draw from the simulation results are:

- Deterministic delay guarantees usually require resource overallocation, i.e. each phone call must be reserved more transmission capacity than the minimum required to transmit voice samples and packet overhead.
- The real-time efficiency heavily depends on the packet size. Thus, it is particularly important to carefully choose the size of packets used for the transmission of voice samples. The optimal packet size depends on call specific parameters, i.e. it should be chosen on a call-by-call basis taking into account information provided by the network like the number of hops traversed.
- Low efficiency (obtained with long packet size) corresponds to large capacity spared for best effort traffic. Thus, in case the network is not inteded to carry mainly phone calls, real-time efficiency can be traded for available bandwidth according to the mix of traffic to be carried on the network.
- Circuit swithing features a higher real-time efficiency in any considered scenario.

Given the last point, should we rethink over the whole packet telephony issue and stay with the circuit switching for providing real-time services?

Resource utilization is not the only comparison criteria in the choice of the technology to be used to provide real-time services and possibly integrate them with data services. Among the others, the lower cost of packet switches is to be taken into account. This stems from the simpler technology, but also the lower reliability of packet switches with respect to circuit switches which feature extremely low probability and duration outages. However, if the user is satisfied with the anyway high level of reliability featured by packet switches, there is no reason to invest in the provision of higher reliability.

Along the same line, if the user is satisfied with a lousier quality, a deterministic delay bound is not necessary. As a consequence, allocation of resources in the network can be reduced, thus increasing the real-time efficiency. The efficiency improvement stems from reducing or possibly avoiding over allocation and from taking advantage of the statistical multiplexing of phone calls in wich silence suppression is modeled. The evaluation of real-time efficiency with probablistic quality guarantees is the subject of ongoing work.

Acknowledgments

This work has been partially supported by Centro Studi e Laboratori Telecomunicazioni S.p.A. (CSELT), Italy. The authors wish to thank Luca Fantolino from CSELT for his insightful comments during the development of the work described in this paper. The authors thank also Simone Martini e Vincenzo Frappietro for their work on the implementation of the simulator.

References

[1] J. Wroclawski. The use of RSVP with IETF integrated services. Standard Track RFC 2210, Internet Engineering Task Force, September 1997.

[2] S. Shenker, C. Partridge, and R. Guerin. Specification of guaranteed quality of service. Standard Track RFC 2212, Internet Engineering Task Force, September 1997.

[3] J. Wroclawski. Specification of the controlled-load network element service. Standard Track RFC 2211, Internet Engineering Task Force, September 1997.

[4] IETF. Differentiated Services (diffserv). URL=`http://www.ietf.org/html.charters/diffserv-charter.html`.

[5] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The siglenode case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[6] J. Nagle. On packet switches with infinite storage. *IEEE Transactions on Communications*, 35(4):435–438, April 1987.

[7] 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.

[8] 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.

[9] V. Bolotin. Modeling call holding time distributions for CCS network design and performance analysis. *IEEE Journal on Selected Areas in Communications*, 12(3), April 1994.

[10] M. Baldi, D. Bergamasco, and E. Guarene. Architectural choices for packet switched telephone networks. In *International Switching Symposium (ISS '97)*, September 1997.