Efficiency of Packet Voice with Deterministic Delay

Mario Baldi and Fulvio Risso, Politecnico di Torino

ABSTRACT

Packet switching is appealing for carrying real-time traffic because it can benefit from (possibly variable bit rate) compression schemes and statistical multiplexing to more efficiently exploit network resources. This work explores the efficiency of IP telephony in terms of the volume of voice traffic carried with deterministically guaranteed quality related to the amount of network resources used. An IP network carrying compressed voice is compared to circuit switching carrying PCM (64 kb/s) encoded voice, and some design choices affecting IP telephony efficiency are discussed.

INTRODUCTION

Circuit switching is particularly suitable for providing 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 taking advantage of low-bit-rate voice encoding, unless multiple phone calls are aggregated in a single flow, significantly increasing the complexity of the network and of call handling.

Packet switching is appealing for carrying real-time traffic because it can benefit from high compression encoding schemes, variable bit rate traffic, and real-time and best-effort multiplexing in order to more efficiently exploit network resources. Moreover, packet switching devices are cheaper than circuit switching ones.

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. 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 to carry users' requests to the network, and policing functions to ensure that the actual traffic generated by users complies with their requests.

Whenever a new phone conversation is to be started, the needed QoS is signaled to the network through some sort of signaling protocol, such as the Resource Reservation Protocol (RSVP) [1] on IP networks.

The described approach to QoS provision is conformant to the model for integrated services (IntServ) over the Internet [2], which has been recognized as having scalability problems. A differentiated services (DiffServ) model [3] has been proposed as a more scalable solution because signaling, call admission control, packet scheduling, and policing are performed with a coarser granularity than the call level. The DiffServ effort is devoted to the definition of single-node-level services (per-hop behaviors). The end-to-end service provided to users — determined by the concatenation of per-hop behaviors of traversed nodes, network dimensioning, and network access control — is not part of the DiffServ framework. Recent proposals suggest combining the IntServ and DiffServ approaches in order to provide some sort of guaranteed service on an end-to-end path while taking advantage of flow aggregation. In this case the IntServ model can be successfully deployed in the edge part of the network, without compromising scalability.

This work explores the *real-time efficiency* of IP telephony, that is, the volume of voice traffic with *deterministically* guaranteed quality related to the amount of network resources used. Since this article focuses on the user-perceived quality guaranteed for each call, the IntServ model is adopted. One of the QoS objectives for a toll-quality phone call is a deterministic bound of about 200 ms on the round-trip delay perceived by users in order to enable nonannoying interaction. Unless differently specified, this is the round-trip delay set in the simulations reported throughout the article.

IP is taken into consideration as packet switching technology for carrying compressed voice, and is compared to circuit switching carrying pulse code modulation (PCM) (64 kb/s) encoded voice. Adaptive differential PCM-32 (ADPCM32) is the voice encoding scheme considered throughout most of the article; the deployment of other encoding schemes is also taken into consideration, highlighting their relative benefits and drawbacks. This work also

Series Editors: K. Elsayed and M. Lerner.

0163-6804/00/$10.00 © 2000 IEEE

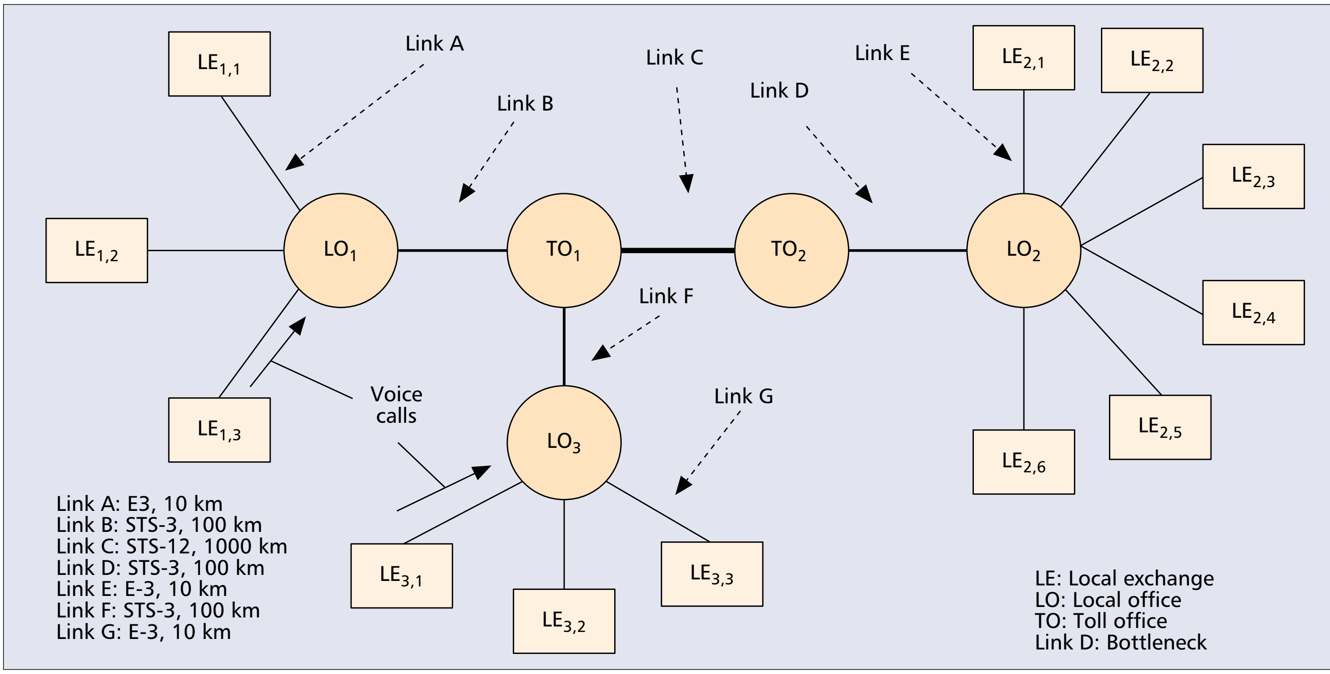


Figure 1. *The network topology used in simulations.*

points out the advantages of advanced resource allocation mechanisms, showing how they improve the efficiency of the network.

Results are obtained through a simulation study on the network shown in Fig. 1; the topology has been designed after that of a domestic telephone network. The deployed call-level simulator [4] assumes that the Packet-by-Packet Generalized Processor Sharing (PGPS) [5, 6] scheduling algorithm is used in network nodes.

The article is structured as follow. We discuss how connection admission control (CAC) is performed when PGPS is used to manage queues in network nodes. Indices, used throughout the article to evaluate the efficiency in utilizing network resources and the main factors affecting them, are introduced. Another section studies the effects of using various voice encoding techniques. We show the results obtained with different resource allocation criteria. Finally, conclusions are drawn.

Call Admission Control

PGPS is derived from the Generalized Processor Sharing (GPS) algorithm, which assumes the *fluid flow* model of traffic: each active flow feeds a separate buffer, and all 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* for flow i.

Provided a flow is compliant with the traffic exiting a leaky bucket with an output rate $\rho_i < g_i$ and depth σ_i, GPS guarantees an upper bound on the queuing delay of each flow i equal to $Q_i^{GPS} = \sigma_i/g_i$.

PGPS, also named *Weighted Fair Queuing* (WFQ) [7], extends GPS in order to handle packet-based flows. The basic idea behind PGPS is that 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 compliant with the above leaky bucket (i.e., leak rate ρ_i and bucket depth σ_i), the queuing delay is deterministically bound [8, Eq. 12.1]. The delay bound is a function of the number of hops on the path of the flow, the service rate of each node (usually the capacity of the output link), the maximum packet size for the flow, and the maximum packet size allowed in the network.

The delay bound is proportional to the burstiness of the source ρ_i and the number of traversed nodes h_i, and inversely proportional to the bandwidth g_i allocated to that source. Thus, when a delay requirement is to be met by a flow i, the higher the burstiness of a source and the number of traversed nodes, the larger the bandwidth g_i must be.

The queuing delay is only a component of the overall end-to-end delay. The CAC is provided with a delay requirement D_{req}, which is the network delay budget for the call obtained by subtracting from the delay acceptable to the user both the time needed for application-level processing (i.e., audio or video compression) and the protocol processing time, not including the delay introduced by the packetization process. The CAC uses the following inequality to determine the amount of network resources needed to guarantee the required QoS to a flow and decide whether to accept it or not:

$$D_{req} \geq D_{pack} + D_{prop_0} + \frac{\sigma_i + (h_i - 1)\cdot L_i}{g_i} + \sum_{m=1}^{h_i}\left(\frac{L_{\max}}{r_m} + D_{prop_m}\right). \quad (1)$$

The inequality takes into consideration the propagation delay D_{prop_m} on the mth link of the path and the packetization delay D_{pack}.

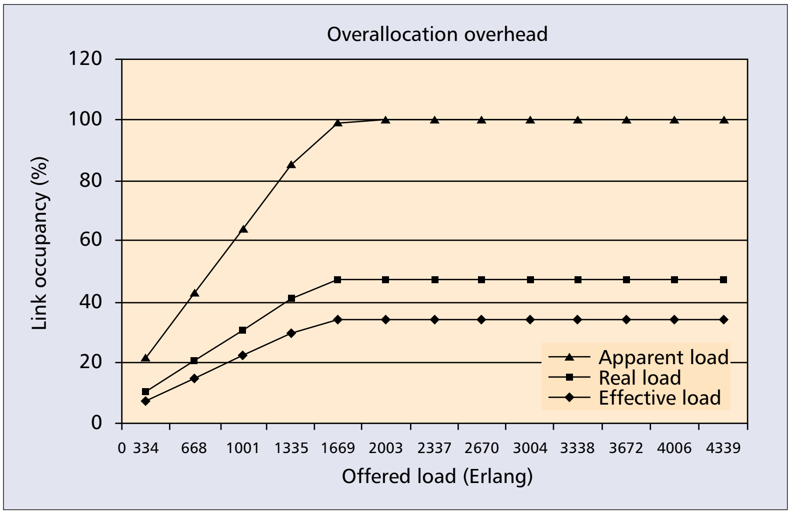


Figure 2. *Voice over IP: efficiency indices on link D.*

The CAC checks whether each link 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 ith flow and g_i^* is the minimum g_i value that satisfies the inequality of Eq. 1. If enough bandwidth is available, the appropriate amount is reserved for the call on every link traversed.

When the amount of bandwidth g_i^* needed to meet the QoS requirement of a flow is larger than the amount ρ_i required to transmit flow i, including protocol overheads, we call it *bandwidth overallocation*. This "over-requirement" can be seen as an extra overhead which possibly adds to the protocol overhead introduced to transmit packet headers. When a call is torn down, the bandwidth previously reserved for it is released.

Efficiency of Guaranteed Services over Packet Networks

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

- *Real-time efficiency* is given by the amount of real-time traffic carried by the network with respect to the amount of resources (e.g., transmission capacity) reserved. The real-time efficiency is relevant when the network is intended to carry mainly real-time traffic, such as a commercial telephone network.
- *Transport efficiency* is given by the overall amount of traffic (real-time and best effort) carried by the network with respect to the amount of resources reserved. Transport efficiency is relevant when a significant part of the traffic is to be best-effort and provision of the corresponding service is not a marginal issue.

This study uses the following set of efficiency indices that are orthogonal to the two definitions above and can be used to compare the efficiency of packet switching and circuit switching [4]:

- The *effective load*[1] is the data rate at the application level and gives an idea of the amount of real-time traffic carried by the network. The effective load does not account for protocol overhead, so it is the capacity that would be required to send the data on a circuit-switched network.
- 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.
- The *apparent load* is the bandwidth reserved for phone calls (more in general to the real-time sessions) in order to meet their QoS requirements and is equal to $\max(\rho_i, g_i^*)$.
- The *network load* represents the number of (accepted) calls active on the network. In analogy with telephone networks, it has been measured in Erlangs, 1 Erlang being the number of circuits (calls) continuously used (active) for 1 hr.

These indices provide a measure of how effectively calls with real-time guarantees can be carried by the network. For example, the lower the apparent bandwidth of a call, the higher the amount of such calls the network can carry; the larger the real bandwidth, the higher the amount of raw transmission capacity required.

The effective load represents the fraction of link bandwidth circuit switching would require to carry the same number of phone calls as accepted by the packet-switched network. Thus, effective load enables the comparison between the packet-switched telephone network and the circuit-switched one from the efficiency standpoint.

Figure 2 shows the effective, real, and apparent load on link D as a percentage of link capacity.[2] Voice samples are carried in Real-Time Transport Protocol (RTP) packets, so the standard encapsulation (RTP, UDP, IP, PPP) results in a 48-byte- header. 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 and all the calls are accepted. When the offered traffic becomes large enough to saturate the bottleneck link (i.e., the apparent load reaches 100 percent of the bottleneck link capacity), the three load curves flatten, indicating that some of the incoming calls are rejected by the CAC. The flat part of the curves represents the maximum link utilization achievable in this scenario.

The difference between the apparent and real load curves is the bandwidth overallocation performed by the CAC. However, this overallocated bandwidth is not really wasted since it can be used to transmit best-effort traffic which has no delay requirements.

The difference between the real and effective load curves represents the amount of bandwidth wasted to carry the protocol overhead (i.e., packet headers). This waste is unavoidable and can be considered the price paid to benefit from the advantages of packet switching.

The difference between the apparent and effective load curves shows how circuit- and packet-switched telephone networks compare

[1] *When referring to a single call instead of the overall network occupancy, the term* bandwidth *is used instead of* load*.*

[2] *Throughout the article we often refer to the load on link D as the load on the network. This is motivated by the fact that because D is the potential bottleneck link of the considered topology, its utilization is a good representative of the overall load on the network.*

from the real-time efficiency point of view. For example, Fig. 2 shows that the same number of phone calls carried on link D using packet switching can be carried with only approximately 35 percent of the capacity on a circuit-switched network carrying ADPCM32 voice calls.[3] In other words, in the considered scenario the real-time efficiency of the packet-switched telephone network is about a third that of a corresponding circuit-switched network.

Bandwidth overallocation plays a key role since, as shown by Fig. 2, it can have a significantly stronger impact on real-time efficiency than protocol overhead. Bandwidth overallocation and protocol overhead are tightly coupled, as shown in the next section.

HEADER AND PACKET SIZE

The header size depends on the protocol architecture deployed in the network; the packet size depends on the packetization delay introduced by the sender.

As shown in Fig. 3, increasing the packetization delay decreases the real bandwidth. Moreover, if the relative overhead introduced by the header is small enough, a phone call on a packet network can require less bandwidth than on a circuit-switched network exploiting PCM encoding. Thus, the real-time efficiency in a packet telephone network can be larger than in a traditional telephone network.

Figure 3 shows different values for the real and apparent bandwidth; the apparent bandwidth curve has a minimum at 18 ms, and then increases with packetization delay. This means that, with the considered topology and delay requirement, bandwidth overallocation is required for packetization delays larger than 18 ms. In fact, as the packetization delay increases, the delay budget left to queuing shrinks, and overallocation is possibly required in order to keep the end-to-end delay below the QoS requirement. The optimal packet size (i.e., the last packetization delay that does not require overallocation) can be devised analytically [4] and intuitively seen in Fig. 3 when the apparent bandwidth curve reaches its minimum value. Increasing the packetization delay reduces the real bandwidth of calls, and the number of accepted calls (i.e., the network load) increases accordingly. However, further increasing the packetization delay beyond the optimal value (18 ms in Fig. 3) leads to overallocation and to a consequent decrease of network load. These phenomena can be observed only when the offered call load is high enough to require all the link capacity.

HOPS

The network topology shown in Fig. 1 has been modified with a variable number of toll offices, and link C has been set to 15,000 km in order to evaluate the impact of the number of nodes traversed by calls. Simulations take in account two alternative delay requirements: a tighter one (400 ms round-trip) and a looser one (600 ms).[4] The IP packet size is fit to one of two scenarios:

- The network is intended to carry mainly real-time traffic; therefore, real-time efficiency is maximized. The IP packet size is chosen in order to minimize bandwidth overallocation; therefore, the incoming calls have the optimal packetization delay.
- The network is intended to allocate half the bandwidth to carry real-time traffic; the remainder is dedicated to transport best-effort traffic; therefore, the transport efficiency is maximized.

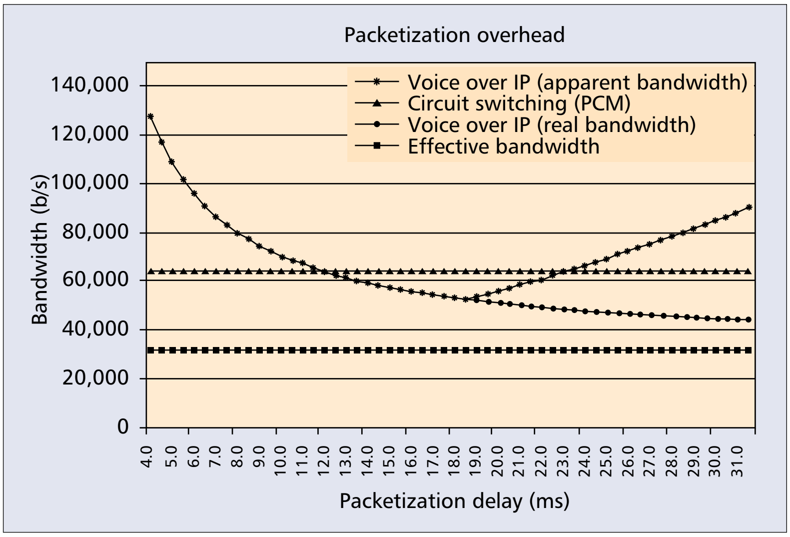


Figure 3. *The impact of packetization delay on the real and apparent bandwidth of a phone call with various technologies.*

In the second case the real-time traffic can take advantage of overallocating bandwidth. Since overallocated bandwidth is "reserved" but not "used," the 50 percent of the link bandwidth that has to be dedicated to best-effort data can be exploited by overallocation. In other words, overallocation is free, unless the percentage of the network bandwidth used by overallocation is larger than the percentage dedicated to best-effort traffic. This permits smaller IP packets so that the real bandwidth of each call can be decreased, improving the transport efficiency of the network. Therefore, the IP packet size is chosen in order to create such an amount of overallocation.

Figure 4 plots the maximum call load accepted by the network vs. the number of nodes on the path of calls and shows that the real-time efficiency is low across a large number of nodes. In fact (Fig. 5), the corresponding packetization delay is becoming smaller and smaller, thus making the header overhead prevail.

The topology of an IP network intended to carry telephony must be designed with this result in mind, and the number of hops should be kept as small as possible on any path. Since the Internet usually features a large number of routers on long distance paths, it could be concluded that PGPS schedulers are not the optimal choice for carrying toll-quality telephony in the present Internet.

It can be noted that the network in Fig. 1 extended for long distance paths has a maximum load of 1450 Erlang when it is intended to carry only real-time traffic (a path with 20 intermediate nodes and 400 ms round-trip delay), against 1100 Erlang obtainable when the network is dedicate to carry 50 percent best-effort traffic. This shows that a high percentage of best-effort traffic enables high transport efficiency.

[3] *Note that currently deployed circuit-switched networks can transport only PCM encoded voice. In this case the bandwidth needed to carry the same amount of voice is approximately 70 percent of the link capacity.*

[4] *The provider could be willing to offer a low-cost long distance service for which the user is required to tolerate higher round-trip delays.*

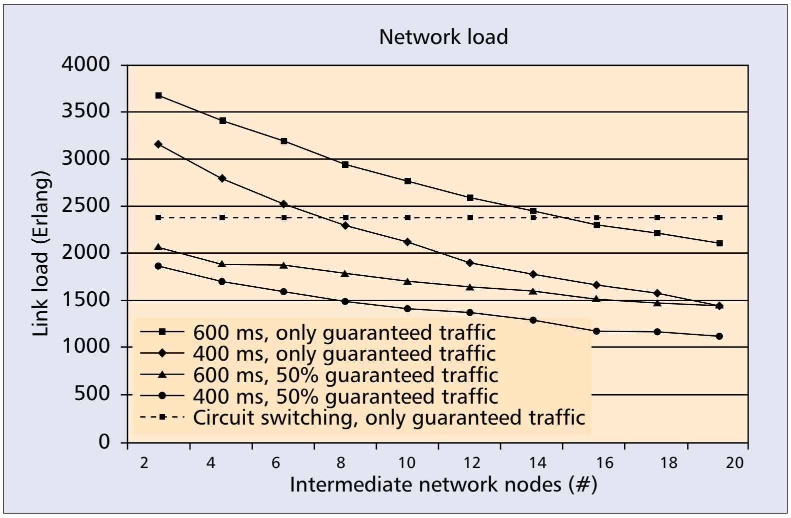


Figure 4. *Long distance calls: maximum load accepted by the network.*

In the foreseeable future best effort will make up most Internet traffic. As long as voice traffic is negligible, the overallocation is no longer a problem because the bandwidth can be exploited by best-effort traffic; therefore, PGPS can be successfully deployed to create networks that offer guaranteed-quality services.

Maximizing Transport Efficiency in the Presence of Best-Effort Traffic

When the network is to be dedicated to carry a certain percentage d of data traffic, the optimal efficiency point can easily be obtained by extending Eq. 6 in [4]. In fact the optimal point is reached when the ratio between the "occupied" and "reserved" bandwidth is exactly equal to the percentage that has to be dedicated to real-time traffic (i.e., $B_{real} = (1 - d) \cdot B_{app}$).

Substituting this optimal bandwidth in Eq. 1 and expanding the term B_{real} with the proper value (Eq. 4 in [4]), the optimal packetization delay results:

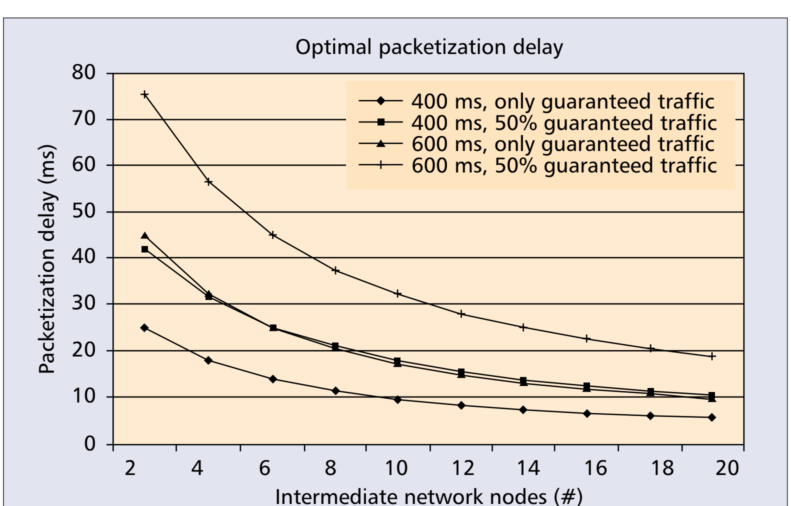


Figure 5. *Long distance calls: optimal packetization delay in correspondence with the maximum load accepted by the network.*

$$D_{pack} = \frac{D_{req} - D_{prop_0} - \sum_{m=1}^{h_i}\left(\frac{L_{\max}}{r_m} + D_{prop_m}\right)}{h_i \cdot (1-d)+1} \cong \frac{D_{req}}{h_i \cdot (1-d)+1}. \qquad (2)$$

The above approximation holds on paths with limited number of nodes and fast links.

Equation 2 can be used to derive the optimal packetization point (i.e., the point that maximizes the transport efficiency of the network) given the percentage of best-effort traffic the network is supposed to carry. They show that the optimal packetization delay depends on such a percentage, thus affecting transport efficiency.

Since the optimal packetization delay depends on many parameters, it is likely that users will operate with a packetization delay other than the optimal one, although close to it. A longer packetization delay requires larger bandwidth overallocation, and a smaller amount of real-time traffic is accepted by the network. As a result, the service provider accommodates a smaller amount of high-cost QoS connections, some users see their calls rejected, and more capacity is left to cheap best effort traffic. If the packetization delay is shorter than the optimal one, real-time traffic produces a larger protocol overhead, which wastes part of the capacity that is intended to carry best effort traffic. To avoid degrading the service provided to best effort traffic, the packetization delay should be chosen longer, rather than shorter, than the optimal value.

The Codec

The possibility to use codecs with different compression factors is among the advantages of packet telephony. A high number of codecs which produce flows ranging from 5.3 to 64 kb/s (traditional PCM) and more (high-quality codecs) have been developed. Voice transmission is based on either encoding voice samples or building a mathematical model of voice and sending the parameters of such a model (i.e., on the mathematical synthesis of voice). Traditional schemes use the former technique, while the most efficient ones (G. 723, CS-ACELP, GSM, LD-CELP) use the latter.

Some encoders operate on multiple voice samples, and their packetization delay can be varied with a fairly coarse granularity. For example, each GSM encoded frame is 260 bits, and the *granularity* of the packetization delay with GSM encoding is 20 ms.

Figure 6 shows the apparent bandwidth of a call according to the codec used. Obviously, the apparent bandwidth grows as packetization delay increases, resulting in a small number of phone calls being accepted on the network. However, a small packetization delay may end up with the same result due to the high overhead introduced. Due to the coarse granularity of high-gain codecs, it may be impossible for the network administrator to choose the real-time efficiency best suited to maximize the utilization of the network according to the traffic mix (namely, the ratio between real-time and best-effort traffic). In the considered network, CS-ACELP is the coding scheme which provides the best trade-off between output bit rate (8 kb/s) and granularity of the packetization delay (10 ms).

RESOURCE ALLOCATION

Traditional telephone networks allocate resources with the granularity of a synchronous optical network (SONET) channel (64 kb/s); the same reservation is performed on each link on the path of the call. Packet technologies enable more flexible allocation which can benefit from tailoring the reservation on each link to the amount of resources available on that link. The *slack term* introduced by the IntServ working group in RSVP [9] can be used to exploit this potential.

In order to evaluate the impact of allocating different amounts of resources on the links along the path, we rewrite Inequality 1 separating the delay contribution of each hop. Moreover, we factor as D_{fixed} the delay components independent of the allocation, thus obtaining

$$D_{req} \geq D_{fixed} + \frac{\sigma_i}{\min_{\{1 \leq m \leq h_i\}} g_{i,m}} + \sum_{m=2}^{h_i} \frac{L_i}{g_{i,m}}. \quad (3)$$

A simple criterion to differentiate allocation among links is to reserve resources proportional to the link *capacity* r_m. Thus, a coefficient K can be introduced so that $g_{i,m} = K \cdot r_m$. The amount of bandwidth to be allocated can be devised by finding the minimum value of K which satisfies Inequality 3. However, on low-speed links the amount $K \cdot r_m$ can be less than the real bandwidth (i.e., the minimum amount of bandwidth required for transmission of the voice samples). In this case $K \cdot r_m$ will be substituted with the real bandwidth, and a new (smaller) K' will be determined for the whole path. The process is repeated until the bandwidth reserved on each link is at least the real bandwidth of the phone call.

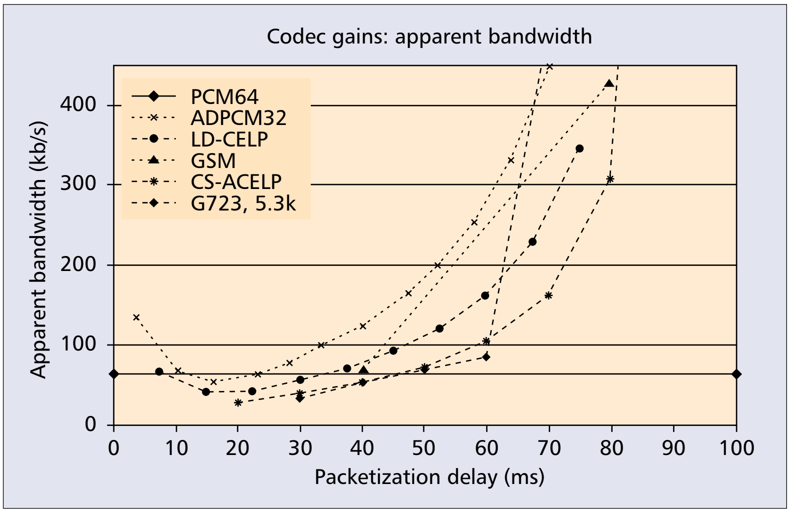
The above described resource allocation criterion can easily be extended to become proportional to the bandwidth *available* on the traversed links. This can be beneficial because high-capacity links are usually located in the backbone where traffic is more intense; thus, high-capacity links are likely to be the most heavily loaded ones.

Figure 7 compares the different allocation criteria with respect to the packetization delay on the network depicted in Fig. 1. The solid lines plot the call load accepted on the network, while the dashed lines depict the network load gain over the maximum load achievable with the flat allocation criterion. The *capacity allocation* shows a maximum gain of 6.5 percent over the flat allocation, while the *available allocation* shows a gain of 10 percent. The relative performance of these allocation criteria strongly depends on how the network has been engineered with respect to the actual pattern of calls.

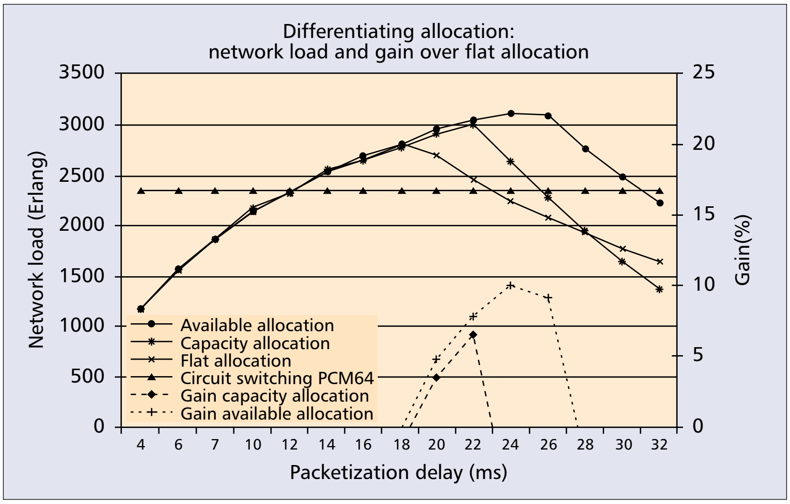
The plot shows the benefit stemming from distributing in a different way the apparent bandwidth allocated on a path. In fact, as far as phone calls have no overallocation, all of the criteria perform the same because the bandwidth allocated on each link is always the minimum possible (the real one). Differences arise when phone calls need overallocation: for example, the *available allocation* criterion tends to allocate the minimum bandwidth on the most congested links and more bandwidth on free links. As a consequence, the delay on the former can be quite high, while that on the latter is reduced to satisfy the end-to-end requirement.

Figure 8 shows the amount of resources reserved on the links according to the various allocation criteria; each plot refers to a different value of the packetization delay. Since an 18 ms packetization delay allows the 200 ms round-trip delay requirement to be met without bandwidth overallocation, the bars of the first graph show that the same amount of resources is reserved on each link.

Higher packetization delays require bandwidth overallocation; the flat allocation criterion distributes the overallocation evenly over all the links. As a consequence, the bandwidth of link D is completely reserved, while only a percentage of the resource is reserved on other links. Instead, the other allocation criterion show a different distribution of the overallocation on the various



■ **Figure 6.** *The apparent bandwidth of a phone call with different codecs.*



■ **Figure 7.** *A comparison of different allocation criteria: accepted call load (left axis) and gain over flat resource allocation (right axis).*

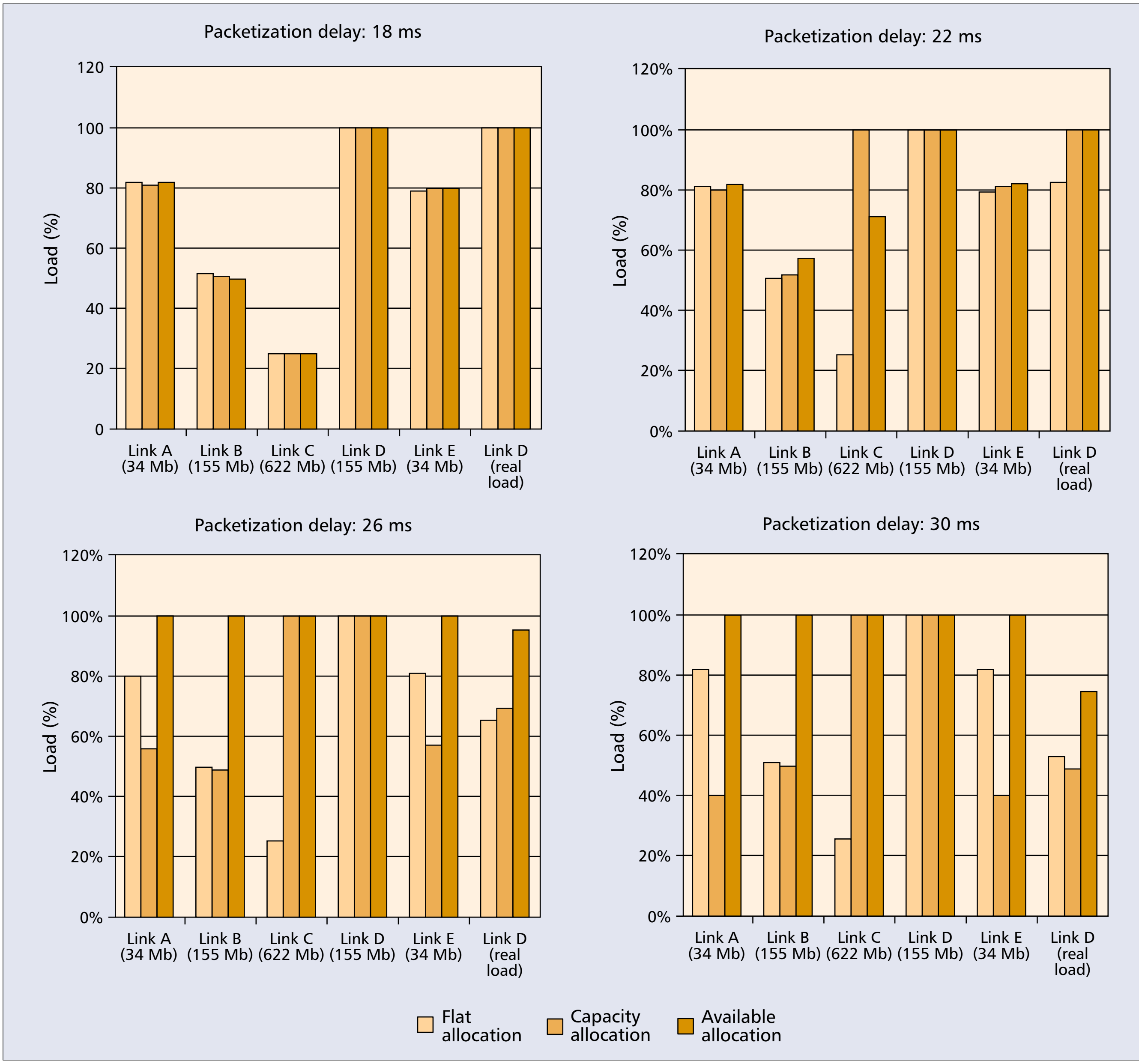


Figure 8. *Bandwidth allocation on each link, varying the packetization delay.*

links. With a 26 and 30 ms packetization delay, the available allocation criterion uses the bandwidth of all the links. As can be noticed by the real load on the bottleneck link D, the available allocation outperforms the others in terms of volume of voice traffic accepted by the network.

When the capacity allocation and available allocation criteria are used, it is harder to determine the optimal packetization delay, that is, the packet size which maximizes the amount of phone calls carried by the network. As the packetization delay increases, the real bandwidth is reduced at the expense of a certain overallocation; the criterion used to distribute the overallocation on the links adds a new dimension to the problem of finding the optimal packetization delay.

While using the optimal packetization delay in a network with flat allocation guarantees that the network is able to transport the desired percentage of best effort, this is no longer true when advanced allocation criteria are deployed. Since some links tend to have less overallocated bandwidth than others, the CAC has to make sure that there will be enough bandwidth left for best effort traffic. This makes the CAC more complicated.

Discussion

Packet telephony features many advantages over traditional circuit-switched telephony: both data traffic and voice traffic are carried on the same network, cheap packet switches are deployed in place of circuit switches, and high-performance codecs can be exploited to produce voice flows at a very low bit rate.

In this article we study, through simulation, the efficiency of IP telephony and the design choices affecting it. The *overallocation* that might be required in order to keep user-perceived delay low reduces the maximum amount of voice traffic the network is able to carry (i.e., the *real-time efficiency* of the network). Therefore, we derived a way to calculate the point that maximizes the efficiency of the network in the presence of best-effort traffic. Moreover, we show that best performances can be obtained when the percentage of best-effort traffic prevails and the number of nodes on the path of voice calls is small.

Despite the common belief, deployment of high-gain codecs might not be so beneficial since some of them prevent the optimization of the network for carrying the actual mix of real-time and best-effort traffic. The implementation of allocation criteria which differentiate resource allocation on the various links can substantially increase the number of phone calls carried by the network. These criteria can be based on mechanisms like Integrated Services' slack term.

Acknowledgments

The authors wish to thank Hewlett Packard for partially supporting this work.

References

[1] R. Braden *et al.*, "Resource ReServation Protocol (RSVP) — Version 1 Functional Specification," RFC 2205, Sept. 1997.

[2] S. Shenker R. Braden, and D. Clark, "Integrated Service in the Internet Architecture: An Overview," (RFC 1633), July 1994.

[3] S. Shenker R. Braden, and D. Clark, "An Architecture for Differentiated Services," RFC 2475, Dec. 1998.

[4] M. Baldi, D. Bergamasco, and F. Risso, "On the Efficiency of Packet Telephony," *7th IFIP Int'l. Conf. Telecommun. Sys.*, Mar. 1999.

[5] 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 Trans. Net.*, vol. 1, no. 3, June 1993, pp. 344–57.

[6] 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 Trans. Net.*, vol. 2, no. 2, Apr. 1994, pp. 137–50.

[7] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of A Fair Queuing Algorithm," *ACM Comp. Commun. Rev. (SIGCOMM '89)*, pp. 3–12, 1989.

[8] C. Partridge, *Gigabit Networking*, Addison Wesley, Oct. 1993.

[9] S. Shenker, C. Partridge, and R. Guerin, "Specification of Guaranteed Quality of Service," RFC 2212, Sept. 1997.

Biographies

Mario Baldi (mbaldi@polito.it) is an assistant professor at the Department of Control and Computer Engineering of Politecnico di Torino. He got his Ph.D. in computer and system engineering from Politecnico di Torino in 1997. His research activity focuses on provision of real-time services over packet-switched networks. He has been a guest researcher at IBM T. J. Watson Research Center, Yorktown Heights, New York; Columbia University, New York, New York; International Computer Science Institute (ICSI), Berkeley, California; and Synchrodyne, Inc., New York, New York.

Fulvio Risso (risso@polito.it) is a Ph.D. student in Computer and System Engineering at the Department of Control and Computer Engineering of Politecnico di Torino. He spent one year at University College London under the supervision of Professor Jon Crowcroft, working on quality of service on packet-switched networks. His research activities focus on advanced scheduler mechanisms.

Best performances can be obtained when the percentage of best-effort traffic prevails and the number of nodes on the path of voice calls is small.