Scalable QoS: state-of-the-art architectural solutions and developments

Technical Report, FTW-TR-2004-003

Authors in alphabetical order:

Ivan Gojmerac

Florian Hammer

Fabio Ricciato

Hung Tuan Tran

Thomas Ziegler

Contents

Abbreviations

AF-PHB	Assured Forwarding Per Hop Behavior
AQM	Active Queue Management
AS	Autonomous System
ATM	Asynchronous Transfer Mode
BB	Bandwidth Broker
BGRP-P	Border Gateway Resource Protocol Plus
BGP	Border Gateway Protocol
CAC	Connection Admission Control
CBR	Constant Bit Rate
CE	Customer Edge
CELP	Code Excited Linear Prediction
DSCP	DiffServ Code Point
ECN	Explicit Congestion Notification
EF-PHB	Expedited Forwarding Per Hop Behavior
ER-LSP	Explicitly Routed Label Switched Path
FEC	Forward Error Correction
GIPS	Global IP Sound
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
iLBC	Internet Low Bitrate Codec
ISP	Internet Service Provider
IS-IS	Intermediate System – Intermediate System
LDP	Label Distribution Protocol
LSP	Label Switched Path
MPLS	Multi Protocol Label Switching
MRTG	Multi Router Traffic Grapher
OC	Optical Carrier
OSPF	Open Shortest Path First
PE	Provider Edge
PHB	Per Hop Behavior
POP	Point of Presence
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RED	Random Early Detection
REM	Random Exponential Marking
RIO	RED with In and Out
RSVP	Resource Reservation Protocol

RSVP-TE	Resource Reservation Protocol- Traffic Engineering
RTCP	Real-time Transport Control Protocol
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SON	Service Overlay Network
SRLG	Shared Risk Link Groups
TE	Traffic Engineering
TCP	Transport Control Protocol
UDP	User Datagram Protocol
VBR	Variant Bit Rate
VoIP	Voice over IP
VPN	Virtual Private Network
WFQ	Weighted Fair Queuing

1 Introduction

1.1 Why another report on QoS?

In the context of modern packet switching networks, the QoS topic has attracted great attention over the last 20 years, both in the industrial and academical fora. Historically, the QoS topic developed in the X.25 and ATM context, and then migrated to the IP world.

Several approaches have been proposed to cope with QoS, and several mechanisms have been made available in commercial equipments in support of QoS techniques. Despite the massive academical and commercial speculations on QoS, to most network operators it is still not clear which is the best way to practically inject QoS in their operational networks. This is partially due to the fact that there is not a single solution universally optimal over all operators. Rather, the best strategy among the available ones should be selected by evaluating the pros and cons of each approach in the light of several operator-specific conditions.

A broad range of mechanisms have been defined to address the QoS commitments, and there is a general trade-off between network capacity and system complexity. And fatally, higher system complexity leads to lower scalability. The optimal choice strictly depends on the relative cost of capacity vs. complexity / scalability. In turn, these costs are related to factors like network size, level of traffic aggregation, traffic variability, targeted services and so forth.

In this framework, the nature of this report is analytical rather than descriptive. Beyond giving a brief, descriptive survey of QoS approaches, this report focuses on the analytical evaluation of the advantages and costs of different QoS approaches, enlightening the criticality along with the points of effectiveness. Our scope is to provide guidelines to a network operator about the more convenient strategy to implement QoS in a practical network. Note that due to the broad spectrum of the topic, this report does not encompass all related aspects. The QoS topic is discussed mainly at the layer 3, and not all the aspects are treated with the same level of detail. For a report on QoS, a good starting point is the very fundamental question: What is Quality of Service ? We will clarify this point in the rest of this section.

1.2 Services and Quality

Before targeting the QoS dilemma, it is helpful to elaborate on the usage of word *service*. We distinguish between **Application Services** and **Transport Services**. The Application Services deal with the *nature* and the *utilization* of the information carried by the network (e.g., telephony, web browsing, file transfer, video-

conference, video streaming, remote gaming, database transactions, etc.). The Transport Services define the characteristics of the connectivity service offered by the network to its users (e.g., bounded delay, bounded loss, guaranteed throughput, etc.). They can be characterized in terms of several metrics, for example

- transfer delay (mean, max)
- packet loss probability
- throughput

These characterize the performance of the transport service once it is in place, implicitly assuming that the connectivity itself is available. Additionally, one might include in the definition of transport service other metrics relevant to the degree of *availability* of the service (or connectivity) itself. These other metrics can be defined in different ways, depending on the particular context. For example, if the service availability is impacted by failures, an appropriate metric is the percentage of out-of-service time, or the probability of connectivity disruption, etc.

A transport service bounded to some performance requirements is said to deliver a certain *quality of service* (QoS).

1.3 Strict-sense-QoS and Availability

Historically, the proposals named after QoS architectures (Intserv, Diffserv, see Sections 2.1, 2.2) have focused on per-flow parameters (packet delay and loss), implicitly assuming that the connectivity is in place. We will say that these proposals are oriented to *strict-sense QoS*, and present them in Section 2.1–2.6. Independently from such a track, several solutions were proposed to improve the degree of service *availability*, typically by improving the network reliability against failures by means of schemes such as rerouting, restoration, protection, etc. An overview of such schemes is given in Section 2.7.

Despite strict-sense-QoS and fault-recovery schemes were treated as orthogonal components by the networking community, the design of a network architecture must necessarily include both components, and take into account the mutual relationships between the solutions provided on the two sides.

Remarkably, MPLS might be considered as an exception to the orthogonal approach. In fact, the MPLS platform can be exploited to enforce QoS control and differentiation along with fast fault-recovery scheme. Nevertheless, the MPLS features that are oriented to strict-sense-QoS are rather independent from the MPLS features addressing fault-recovery. Section 2.5 reports on possible combined scenarios based on MPLS.

1.4 QoS assurances and QoS differentiation

Any application service involves certain requirements on the underlying transport service, e.g., telephony and video-conference require bounded-delay transport service, as well as a high degree of availability. In principle, the set of transport services delivered by the network should somehow match the supported application services. Therefore, one component in the QoS topic regards the *assurance* of target quality levels for the most demanding flows, that is the support of transport service with guaranteed delay / loss / throughput. On the other hand, in most practical cases, the packet network shall not be dedicated to a single application service, but rather support a multiplicity of application services with different performance requirements. In this case, the network might support the full set of applications on top of a single transport service, tailored to the most strict transport requirements over the whole set. This approach in general requires a very large volume of resources (bandwidth), and in most cases this is not an economically viable approach. In such a case, one alternative solution would be to *differentiate* on the transport services, exploiting the difference in the performance requirements to save network resources.

Therefore, any QoS architecture must deliver both QoS assurances and QoS differentiation, and this applies on both sides of strict-sense-QoS an fault-recovery schemes.

1.4.1 QoS assurances

In order to deliver some degree of QoS assurances (or QoS guarantees), the network should somehow control the relative amount of traffic with respect to available resources. Depending on the relative abundance of resources (e.g., bandwidth) such control can be more or less sophisticated, more or less accurate.

At one extreme, one might adopt the *Over-provisioning* approach, i.e. over-dimensioning the network capacity, and avoid any active control scheme. This approach is convenient when i) the cost of network capacity is low (or at least cheaper than the cost of any other active control schemes); ii) the traffic dynamics and volumes can be safely estimated; and iii) the risk of anomalous traffic concentration is very low.

In order to save on network capacity with regard to the over-provisioning scheme, one might adopt some active mechanism aimed at a better utilization of the given network capacity, for example by injecting some degree of adaptivity into the routing (*Traffic Engineering* schemes). Alternatively, instead of triggering a network adaptation, one might implement schemes that trigger the sources adaptation (e.g., *Active Queue Management (AQM)+rate adaptation* and similar).

The adaptation mechanisms (on the network or on the sources) can coexist with some degree of over-provisioning. For example, in an intermediate scenario the network is over-dimensioned with respect to nominal traffic condition, so that in normal operation the adaptation mechanisms are silent. On the other hand, in case of anomalous or unexpected traffic volume or concentration, the adaptation mechanisms protect the system from collapse.

Since in such approaches there is no explicit blocking of traffic, their success is limited to those scenarios where the risk of anomalous traffic concentration that can not be absorbed by the adaptation mechanism is very low. If this is not the case, one should include some mechanisms into the system to explicitly control traffic access to the network. These typically means implementing some *Reservation* schemes, enforced through Admission Control. The granularity and the accuracy of the reservation schemes depend again on the availability of network capacity, and on the characteristics of the traffic.

1.4.2 QoS differentiation

Any QoS architecture designed for a multi-service network must cope with different kinds of traffic. In fact, every application generates traffic with different characteristics in terms of, e.g.,

- source characteristics (CBR, VBR, adaptive),
- transport protocol (TCP, UDP),
- interactivity,
- flow and packet size.

Additionally, each application is associated to different QoS requirements.

The differences can be exploited to achieve effective sharing of the global network capacity. For example, packets belonging to real-time traffic can be queued in higher-priority buffers at the router interfaces, so that a larger amount of non-real-time traffic can be sent through the link than were possible with a single buffer for real-time and non-real-time packets. In fact, in the latter case the strict delay requirements of real-time packets would impose a severe restriction on the amount of total traffic on the link. In other words, packet prioritization at the scheduler is helpful to preserve the strict delay requirements of real-time traffic, and at the same time allows for a better use of the global bandwidth. Differentiation in packet handling can be achieved by several scheduling and queue management schemes. In any case it is important to remark that *packet prioritization does not necessarily imply service prioritization*: to stay with the above example, real-time packets are

prioritized because they are more sensible to delay, NOT because real-time users are more important than non-real-time ones.

To this point, we suggested that differentiation in traffic handling can emerge as a means to exploit differences in QoS requirements to achieve a better network utilization. Additionally, packet-level differentiation mechanisms can be exploited to enforce some level of *separation* between different traffic components. This is useful whenever some traffic component may impact another one. As an example, the separation of UDP from TCP aggregates (e.g., by weighted fair queuing (WFQ) scheduling) prevents the first one to starve the latter, since UDP does not implement congestion-control feedback. Analogously, differentiation of short-lived and long-lived TCP connections may be considered a useful feature.

Despite differentiation is an attractive component of QoS architectures, it should not be abused since it also has its cost. First, any differentiation implies that some packet classification schemes are in place. Secondly, scheduling and queuing differentiation are associated to parameters that are often difficult to tune.

1.5 Organization of the report

In line with the beforementioned aims, this QoS report is organized as follows. We begin with Section 2, describing briefly the evolution of QoS architectural solutions concerning both strict-sense-QoS and service availability aspects. Afterward, in Section 3, the main part of this report, we discuss several timely issues, which emerge in the context of how and where to achieve scalable QoS with today's networking technologies. Note that the focus of this report is put mainly on the backbone network environment. Therefore, we do not go into the details of QoS solutions in the access and wireless network environments. Finally, we provide our conclusions in Section 4, stimulating further interesting, open discussions on the QoS topic.

2 QoS solutions and aspects

In this section, we describe the state-of-the-art architectural solutions for QoS (in a sense of both QoS assurances and QoS differentiation). In addition, we devote special attention to the service availability aspect and its solutions.

2.1 Intserv

IntServ [1] was the first architectural approach developed by the Internet Engineering Task Force (IETF) towards QoS provisioning in IP networks. The heart principle of the IntServ architecture is to ensure per-flow based QoS guarantees by

means of the signaling protocol RSVP (Resource Reservation Protocol). A new flow with its QoS requirements is served only if the network side has enough available resources yielding the required QoS. This ensures that with RSVP the resource reservation for the new flow is feasible, satisfying two following objects at once. The strict QoS requirements of the new flow are guaranteed and the inclusion of the new flow would not deteriorate the QoS of other, currently active flows in the network. The resource availability is checked by performing admission control for each entering traffic flow.

Services provided by IntServ are flexible and highly appealing in the sense that strict QoS requirements are assured for each micro traffic flow crossing the network domain. However, regarding the implementation, the whole concept seems not to be scalable because of the necessity of per-flow signaling elaboration in routers. Maintaining and processing per-flow control information it each router might not be realistic and makes routers' performance very poor, since the number of flows the router accommodates may be extremely large in a core network. As a consequence, the deployment of IntServ is only advisable in an access network environment, where scalability is not the main issue due to the relatively small number of traffic flows a router has to maintain.

2.2 DiffServ

Recognizing the strongly limited scaling properties of IntServ, the IETF has introduced the Differentiated Services (DiffServ) architecture [2] to overcome this scalability shortage. In the DiffServ architecture, a concept of *service differentiation* has been worked out by defining a few number of traffic classes. QoS is rendered to the classes or, in other words, to the aggregations of individual traffic flows. In an early version, beyond best-effort services, DiffServ can offer premium services and assured services. The premium service is treated with Expedited Forwarding per hop behavior (EF-PHB), while the assured service is treated with assured forwarding per-hop behavior (AF-PHB). The assured service provides *qualitative differentiation* between traffic classes by setting different *drop precedence levels* for the traffic classes. In the last years, further refinements concerning the qualitative service models have been developed. For example, the *proportional differentiated service* model [3] defines a service model in which the ratios of loss rates and packet delays between successive priority classes remain roughly constant. Another example is the *quantitative assured forwarding* service [4] that is able to ensure both absolute guarantees for traffic classes and proportional guarantees between them.

In the DiffServ architecture, per-flow states are maintained only at the edge routers of the domain, while the core routers only have to treat with a few traffic

classes by performing aggregate scheduling. As a consequence, QoS guarantees are delivered only to the aggregate traffic (or classes). The main technical features of Diffserv are the following (for a more complete description, we refer the readers to our technical report [5]):

- **Packet classification at the edge routers**: When entering the network domain, a packet will first be classified. This is done by checking the DS (Differentiated Service) byte in the packet header or checking this DS codepoint together with other header fields (source address, destination address, protocol ID etc.).

- **Packet conditioning at edge routers**: after being classified, the entering packet is subject to further conditioning actions, which depend on how the packet conforms to the traffic profile (e.g., the peak rate, the burstiness of the traffic) predefined for the class it belongs to. The conditioning actions comprise marking and policing. It means, for example, that if the packet is out of profile (i.e. it violates the given traffic constraints of the profile), it may be a subject of shaping or dropping.

- **Packet forwarding at core routers**: Once the packet has been marked at the edge router, it will receive the treatment associated with its class at each core router in its path. For example, if the non-preemptive priority scheduling is implemented in routers, a packet belonging to the high priority class will get the EF PHB and will be directed to the high priority queue, meaning that it will be served before packets of all other classes. In case WFQ scheduling is implemented, a packet will be directed to the queue associated with its class which has a predefined share of bandwidth.

Later, in Section 2.6 we argue that classifying and handling traffic in some classes according to the DiffServ paradigm, coupled with over-provisioning the bandwidth provides a highly reasonable QoS solution. In case over-provisioning is not deployed (e.g., in a relatively small ISP's network), DiffServ raises the concern about per-flow and per-class QoS interrelation. In more detail, DiffServ has resolved the scalability inside the network domain, but it does not assure strict per-flow QoS, i.e. per-class QoS guarantees do not imply that a flow belonging to the given class experiences the same level of QoS. In fact, various work (see e.g., [6, 7]) have demonstrated that the per-flow QoS and per-class QoS may differ from each other leading to cases in which per-class QoS is fulfilled but per-flow QoS is not. This phenomenon has again necessitated the involvement of per-flow admission control. Admission control in the DiffServ architecture is performed mostly by an automated resource management entity called BB (Bandwidth Broker) that has a complete view on the actual resource usage in the network domain.

An alternative way is a family of measurement based, end-point admission control, in which edge routers of the domain are responsible for the decision of accepting or rejecting the new flow with regard to the overall QoS picture in the network. We refer the interested readers to [8] for the detailed overview of the applied CAC mechanisms.

2.3 User-Network interaction based architecture

One of the plausible QoS solutions is to better exploit the QoS supporting features of the current best-effort architecture. They are, for example, the *congestion avoidance* feature or, more precisely, AQM mechanisms deployed at routers and the *source rate control* feature deployed at the traffic sources.

Generally speaking, the user-network interaction based architecture assures QoS by combining the co-operation between the end-user side and the network side. *The principle is that the network side provides the users with the QoS evolution picture inside the network, based on which the user side takes some certain reactions in favor of QoS*. In today's practice, the first task is mostly realized by performing (enhanced) *congestion avoidance*, which comprises smart packet dropping (or packet marking) at routers and the notification of the dropping (marking) rate to the end users. The second task is overwhelmingly done by implementing strategies for the sending rate adaptivity. For voice applications, additional alternatives of the second task may be the adaptation of the FEC (forward error correction) functionalities, or dimensioning the jitter buffer size.

Packet dropping or marking at routers in fact represents the router ability on congestion observability, which can be resolved by applying AQM mechanisms such as RED (Random Early Detection) [9] and its enhanced versions (e.g., [10, 11]), or probably some other, newly envisaged mechanisms (e.g., BLUE [12], REM [13], GREEN [14]). While in the traditional AQM mechanisms (RED and its versions), the congestion state is indicated by the packet loss or packet marking rate, some recent solutions (e.g., GREEN or REM) use the arrival rate as a measures of congestion severity. The deployment of GREEN or REM, for example, thus yields very low queuing delay and negligible packet loss, hence is able to ensure the strict QoS requirements of real-time applications.

The congestion state is signalled back to the sender. Basically, the feedback can be arranged by two approaches. We can either exploit the acknowledgement capability of the underlying transport protocol like TCP, or we can use the explicit congestion notification (ECN) solution. In case of UDP traffic of voice and video applications, since there is no such feedback information implemented at the UDP level, we can rely on the Real-time Transport Control Protocol (RTCP) for sending feedback from the receiver to the sender.

Once the sender has obtained the information on the congestion state in the network, it acts in his own interest to adjust the sending rate. In case TCP capability is exploited, the sender simply reduces the congestion window by half, resulting ultimately in a decrease in sending rate. However, we can design more sophisticated schemes, in which the basic principle that drives the adaptation of the sending rate is to maximize the *net benefit* of the user. Two fundamental functions must be defined for the strict definition of the net benefit:

- Utility function $U(x)$: this function gives the level of usefulness the users could earn by having a sending rate x. In other words, the utility function reflects the perceived QoS of the users. It is constructed based on the features of the application.

- Congestion-price function $C(x)$: this function gives the price of an amount bandwidth x with regards to the congestion level in the network. The price is controlled by network operators. Intuitively, the more severe the congestion state in the network, the higher the price assigned to the bandwidth units.

Given the above functions, the net benefit of a given user is defined as $U(x) - C(x)$. From the economic point of view, the whole phenomenon resembles to a supply-demand pricing process. The network side has bandwidth to sell at the price reflecting the congestion level inside the network. The user side has a freedom to decide how much bandwidth it will buy, i.e. it controls its own *willingness to pay* based on the net benefit at a given bandwidth price.

Since an individual user is always prone to maximize its net benefit independently of other users, the phenomenon is akin to game theory at the network scale, where many users are simultaneously present. Because each user can react in an individual manner to the congestion feedback, i.e. the users can employ different rate adaptation strategies, the differentiation between users' perceived QoS is implicitly achieved. As a simplest example, consider the case where some users, who really have a willingness to pay the price, continue their transmission. Other users, who find the transmission unworthy will stop themselves, leaving the bandwidth for other contenders. In this way, QoS assurance is granted to the first group of users, and this QoS assurance is better than that the latter user group can get, leading to QoS differentiation.

Certainly, the most important design issue in this architecture is the proper choice of the functions $U(x)$, $C(x)$. If we rely purely in the features of the currently used TCP without any (enhanced) design, then what we get is exactly the current best-effort paradigm[1]. Thus, a step toward QoS is a careful application-

[1]The well-known inverse square root relation between the sending rate x and the packet loss rate

oriented specification of $U(x)$ and resource-oriented specification of $C(x)$, as well as manager entities accomplishing rate controller actions.

As a specific example of this user-network based architecture, let's take a closer look on how it is realized in case of voice transmissions, widely known as voice over IP (VoIP) applications. With regard to the network architecture, the perceptual QoS of VoIP applications can be improved by facilitating receiver feedback on the QoS parameters of the IP network. The structure of the signal processing parts of a VoIP system is depicted in Figure 1.

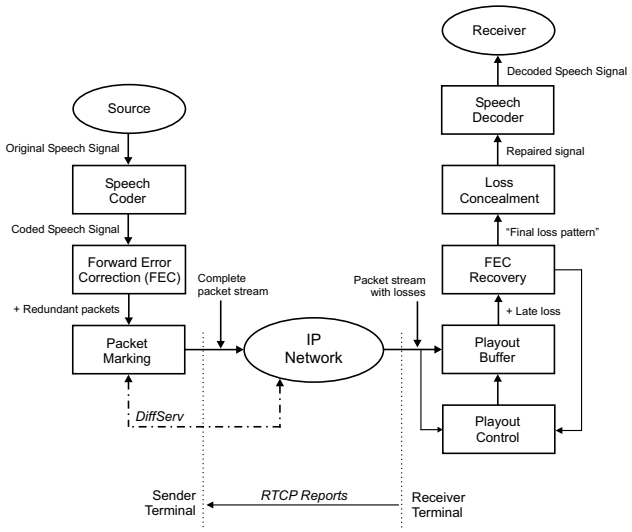

Figure 1: Application layer focused VoIP network structure

Network parameters can be monitored by the sender using RTCP receiver re-

p of a TCP connection ($x \sim 1/\sqrt{p}$) will maximize the objective function $U(x) - C(x)$ with the choice $U(x) = K - 2/T^2 x$, (K is an arbitrary constant, T is the round trip time of the connection), and $C(x) = px$. In other words, TCP defines a priori the functions $U(x)$ and $C(x)$ for all TCP connections. Consequently, the chance for QoS differentiation is really limited, because TCP connections with the same RTT sharing a bottleneck link are expected to have the same sending rate.

ports [15] or extended reports [16] are intended to provide information on packet loss and discard metrics, delay metrics, signal related metrics, call quality or transmission quality metrics, configuration metrics, and jitter buffer parameters. Such a mechanism allows for the sender to adapt the technical properties of the connection. The following parameters can be tuned:

- *Codec selection.* If provided by both terminals, the speech coding algorithm may be changed during the connection according to the available bitrate (e.g., G.711 (64 kbps) and G.729 (8 kpbs)), and according to the packet loss robustness. The Internet Low Bitrate Codec iLBC [17] takes a bandwidth of 15.2 kbps, but provides more graceful speech quality degradation in case of packet losses than G.729. The iLBC is about to get an IETF Proposed Standard (see Issue 1 in Section 3 of this report).

- *FEC.* Redundant packets may be added using forward error correction (FEC, [18, 19, 20].

- *Packet Marking.* Apply some kind of packet marking algorithm at the sender side, e.g., perceptual packet marking (see below).

Packet marking may be used in different ways. Aside from using some arbitrary marking pattern like tagging every second packet to potentially improve the performance in case of burst losses, speech properties can be used to distinguish levels of perceptual importance. Not every voice packet is equal with regard to the impact of its loss on the perceptual speech quality. This is due to the effect of "phonemic restauration" on which, from a linguistic point of view, the codec's packet loss concealment methods are based on. Some parts of the speech can be concealed well, but others can not. A discrimination of importance levels can, e.g., be used in combination with DiffServ, marking voice packets on two or more levels, and handling them in different ways like dropping unimportant packets or forwarding important packets corresponding to their mark. In such a way, the speech quality would degrade more graceful than by ignoring the perceptual importance. Related work on this topic has been done by Sanneck [21], De Martin [22], and Hőne [23].

2.4 Flow-aware architecture

Flow-aware architecture is a research QoS proposal in which all the associated mechanisms are carried out on the abstraction of flows [24]. Briefly speaking, the building blocks of the flow-aware architecture are

- the flow-level, bufferless multiplexing concept, and

- on-the-fly flow-level admission control.

The proposal advocates flow-level resource dimensioning with respect to QoS, since traffic analysis performed at the packet level is very complicated due to the manifestation of long range dependence. Furthermore, in order to derive the necessary resource (i.e. bandwidth) for a target quality criteria, the bufferless multiplexing concept is adopted. That means that no buffer is provided for the exceeding traffic in rate overload situations, i.e. situations when the input traffic rate is larger than the output rate of the multiplexing stage. The criterion for resource dimensioning is thus to keep the rate overload probability below a predefined threshold. This bufferless multiplexing scheme, naturally, reduces somewhat the achievable resource utilization compared to the case of using the buffered scheme, where a buffer is used to absorb the exceeding traffic. Fortunately, this reduction has been shown not to be significant. Moreover, another important argument why not to use the buffered multiplexing scheme is that the buffer size (or the buffer overflow probability) very much depends on the variation structure of the incoming rate, and is therefore hardly controllable.

In parallel to having a good flow-level resource dimensioning, admission control is performed at flow level to prevent overload congestions. A new flow could be blocked if there is not sufficient resource. Expressing this more precisely, in case we deal with streaming and elastic traffic, an integrated admission control mechanism would only accept a new flow if two following conditions hold at once [24]

- the current load is less than a given threshold. This is to ensure the quality for streaming (e.g., voice) traffic flows,

- the available bandwidth (the bandwidth a new flow would attain assuming fair sharing) is greater than another threshold. This is to ensure the throughput guarantee for elastic flows.

To obtain knowledge about the current load and the available bandwidth, load measurement and bandwidth estimation are to be employed, i.e. measurement based admission control needs to be achieved.

From the practical point of view, this architecture would require flow-awareness at each router in the network. Routers have to keep track of the number of active flows, and to make admissibility decisions. To exclude the need of signaling, flows are identified in an ”on-the-fly” manner, meaning that the flow ID in each packet header is compared with the list of flows in progress. If the flow exists, the packet is forwarded, otherwise admission control test must be performed. In case the admissibility is affirmative, the new flow ID is added to the list, and the packet is

forwarded. In case of rejection, the packet will be dropped, which the user generating this flow should interpret as flow rejection. An active flow is deleted from the list if a preset time-out elapses from the registration of its last packet.

Up to date, we are not aware of any prototype or testing environment of this architecture. We also see some hurdles of this architecture. As mentioned earlier, in order to deliver QoS and at the same time to avoid elaborations in the basic of per-flow states, "on-the-fly" admission control has to be deployed. The admission control, as being done "on-the-fly", requires flow identification matching *for each packet* in routers. In backbone routers, despite that the number of active flows might remain in a range of some hundreds thousands, per-packet elaboration would mean significant processing overhead. Another issue is that for services like VoIP, QoS is often construed at the packet level, i.e. is expressed in packet loss, packet delay and jitter. Until now, we are not aware of a good mapping between flow level QoS (e.g., flow blocking probability, throughput) and packet level QoS (i.e. packet loss, delay and jitter). A preliminary investigation in this topic is available, e.g., from [25], but the results therein are only valid with a quite special approximation of assimilated Poisson traffic, which does not hold for the general cases. In our opinion, to derive the packet level QoS from the abstraction of flow-level bufferless multiplexing is still an open issue. In overall, more research and practical validations are required to appraise the feasibility and practicality of this flow-aware QoS solution.

2.5 MPLS

2.5.1 Introduction to MPLS

Several providers have already injected MPLS technology into their networks. This was done primarily because MPLS provides an easy and scalable support of Virtual Private Networks (VPN) with minimal management overhead.

In the most simple scenario, labels are dynamically allocated through the domain along the default IGP (Interior Gateway Protocol) paths, in such a way that a full mesh of LSPs (Label Switched Paths) is installed between edge routers. The attractiveness of this capability is that the label allocation process is based on a standard protocol LDP [26], it is fully distributed (LDP message exchange only applies between IGP neighbors), scalable, and most importantly it is completely dynamic. The labels allocated in this way are often referred to as "hop-by-hop LSP". Noticeably, since such labels are *not* associated to ingress-/egress-router pair, but rather to egress-router only, the number of allocated labels scales only as N (= number of edge routers).

The full mesh of hop-by-hop LSPs offer a packet forwarding platform which

is independent from IP addresses. This features is exploited by the so called BGP/MPLS (Border Gateway Protocol/Multi Protocol Label Switching) architecture [27], which integrates it with a fully dynamic mechanism for distribution of private routes based on BGP. In summary, both functionality - namely the IPaddress-free forwarding based on MPLS, and route-distribution based on BGP - are achieved in a fully dynamic manner, which allows for an easy and fast deployment of large VPNs with minimal manual configuration.

The success of the BGP/MPLS VPN model has driven the success of MPLS, and is deployed by many ISPs. From a strictly technical perspective, in such VPN model the necessary component is not MPLS itself, but rather the IPaddress-free forwarding mechanism. MPLS is a possible way to implement it, but other alternative schemes might be used in conjunction with BGP to achieve dynamic VPN model, for instance IP-over-IP encapsulation. The point of VPN is further covered in Issue 7 in Section 3 of this report.

The rest of the traffic (i.e. public, non-VPN traffic) can be either carried with native IP forwarding, or encapsulated in MPLS headers as well. In the latter case, all the traffic is carried in MPLS LSP, so that the routing decision is concentrated at the ingress edge router. A potential advantage of this scheme is that it is possible to switch-off the BGP processes at the internal routers. On the other hand, the main disadvantage is that a full mesh of LSPs has to be created between *all edge routers* - not only the PEs (Provider Edges) - and that also non-VPN edge routers must be MPLS-capable.

At the first stage of MPLS deployment, LDP is activated and a full mesh of hop-by-hop LSPs are dynamically established. Physical paths do not depart from default IGP routing, and no bandwidth reservation is enforced. Once deployed, MPLS can be further exploited to improve network resilience by means of Fast Restoration capabilities. This can be achieved by local-protection or path-protection, as discussed in Section 2.7. Additionally, MPLS can offer support to traffic engineering and strict-sense-QoS: in the rest of this section we discuss how this can be achieved.

It is just important to mention here that in order to implement path-based protection and/or traffic engineering schemes, one needs to route LSPs onto non-default paths. This can be achieved by installing the so called "Explicitly Routed LSP" (ER-LSP). This requires a more sophisticated signaling protocol than LDP, namely RSVP-TE (Resource Reservation Protocol- Traffic Engineering) [28]. With ER-LSP labels are allocated on a per ingress-/egress-node pair basis. For those applications involving an exchange of traffic between any pair of PEs this would require a full mesh of ER-LSP, which would lead to a number of labels and signaling sessions scaling as N^2, where N is the number of PEs. It is questionable whether this scenario is scalable to realistic values of N, today in the order of

several hundreds.

On the other hand it is possible to envisage some practical scenarios where ER-LSP only applies to selected PEs pairs, without requirin a full mesh of ER-LPSs. For instance, in some application scenarios ER-LSP are used only for some selected traffic collected at a reduced number of PEs (e.g., telephony traffic collected at ToIP gateways). Also, some applications would not involve any-to-any traffic exchange between all PEs, as for example VPNs involving a reduced number of sites. Additionally, for traffic engineering purposes it is possible to envisage mixed scenarios where the usage of ER-LSP is restricted to a small number of traffic streams - tipically those between dominant POPs (Point of Presence), which are likely to carry a large portion of the global traffic - leaving the rest of the traffic being supported by hop-by-hop LSPs.

As a last remark, in current architectures the LSPs that are seen by the network are always associated to Provider-Edge routers (PE, in the terminology of RFC2547 [27], not to single Customer-Edge routers (CE) nor customer interfaces. In fact, CE-to-CE LSPs are always tunneled into PE-to-PE LSP. Any model oriented to exploit direct CE-to-CE LSP through the network, each with an associated RSVP-TE signaling session, might rise serious scalability concerns, similar to those recognized for IntServ.

As an example for an MPLS architecture allowing CE-CE LSPs, an ISP can be mentioned that allows a national wide company having sites in various cities to interconnect these sites with a full mesh of LSPs.

2.5.2 MPLS and QoS

There are at least two points of contacts between MPLS and QoS:

- Packet prioritization
- Traffic Engineering

Diffserv over MPLS

The first point regards the mechanisms that are related to traffic prioritization / differentiation at the packet level, namely scheduling, queue management, classification, policing etc. Basically, these mechanism are the same for Diffserv, the only difference being in that MPLS packets are handled in place of native IP packets. Therefore, the specification of such mechanisms is usually referred to as ”Diffserv over MPLS” techniques. The packet marking must be done in the MPLS shim header rather than in the IP header (i.e. DSCP field) as in the original Diffserv model, and two possibilities exist: mark the LABEL field (L-LSP) or the EXP field (E-LSP). In both cases it is possible to separate different types of traffic into separate LSPs and handle them independently.

There exist two options to implement QoS enabled MPLS LSPs using WFQ scheduling in routers and DiffServ. Either an MPLS LSP can be mapped to a dedicated WFQ queue and differentiation between EF, AF, and best effort traffic is performed ”inside” the LSP. Such an implementation would run into severe problems of scalability and management complexity as the numbers of LSPs and thus WFQ queues to be scheduled would be high. Thus we do not recommend such a solution. Alternatively, each router can have one WFQ queue per traffic class (e.g., a total of three queues for EF, AF, and best effort). An LSP solely transports packets of a single traffic class and is mapped into the corresponding WFQ queue. Edge routers classify arriving packets into LSPs according to their DiffServ codepoints, perform traffic policing, and are connected to other edge routers via a full mesh of LSPs per traffic class. Such an architecture would exhibit less problems in terms of scalability.

MPLS and Traffic Engineering

At the first stage of its deployment, MPLS is used with so called hop-by-hop LSPs. A full mesh of hop-by-hop LSPs can be automatically created with simple LDP message exchange between neighboring routers. These LSPs always map the default paths determined by the running IGP (OSPF or IS-IS).

In a second deployment stage, RSVP-TE protocol is activated in the routers. RSVP-TE is basically a signaling protocol to install the so called ”Explicitly-Routed LSPs” (ER-LSP). Remarkably, an ER-LSP can be i) source-routed over a non-default path, and ii) associated to a bandwidth reservation. These two capabilities are independent from each other. For example, one can use ER-LSP to optimize the routing and pursue better load balancing, without any associated bandwidth reservation. In fact, with RSVP-TE it is possible to route ER-LSPs over non-default paths which have been selected according to some bandwidth availability constraints (constraint based routing) and/or according to some load-balancing and optimization criteria. If only bandwidth optimization objectives are pursued, bandwidth reservation can be avoided at all. Bandwidth reservation associated to ER-LSPs is required in the first case, while is optional in the latter one.

Such techniques are collectively called ”MPLS Traffic Engineering” (TE). MPLS-TE can be done statically or dynamically, depending on whether non-default paths are determined by human operators - eventually with support of some external tool - or by the network elements themselves. In turn, dynamic TE can be implemented in a centralized or distributed fashion.

It is clear that TE techniques are not needed if a generous over-provisioning scheme is adopted. On the other hand, the adoption of more or less sophisticated TE strategies might be helpful to mitigate the level of over-provisioning. First, by introducing the possibility to exploit alternative non-shortest paths, TE intrinsically increases the potential network capacity that is available between remote

node pairs. Second, dynamic TE provides the network with an additional degree of adaptivity to absorb peaks in the spatial traffic distribution (hot-spots), allowing for less conservative capacity provisioning. Additionally, dynamic TE can be helpful in absorbing unanticipated changes in the traffic volume or distribution. In other words, MPLS-TE can provide the network with an additional degree of *elasticity*, thus mitigating the level of required over-provisioning.

In practice, the potential benefit of such elasticity versus rigid over-provisioning largely depends on the characteristics of the traffic at the macroscopic level – spatial distribution, variability in time – and on its predictability. Clearly, the potential benefit of MPLS-TE techniques must be compared with the cost of such techniques in terms of system complexity.

A major concern of network operators regarding network operation in the core section is about *scalability*. RSVP-TE definitely improved over RSVP with respect to scalability in the number of traffic flows: in fact RSVP-TE aggregates all reservations between a pair of PEs onto a single reservation. On the other hand RSVP-TE might still suffer problems of scalability in the number of edge routers. This is the case in those applications involving any-to-any exchange of traffic between all PEs pair, since full mesh of RSVP-TE sessions in support of a full mesh of ER-LSPs might not be sustainable by current equipments.

On the other hand, for such applications it might be expected that in practice it is possible to achieve quasi-optimal TE performances by diverting only a minor part of LSPs to non-default paths, for example the ones carrying more traffic. This is consistent with recent results in [29] about the characteristics of the traffic, that show that very few POP-to-POP flows account for a large portion of the global traffic. Alternatively, traffic engineering techniques might be restricted to those selected traffic components that do not involve full mesh of connectivity between all PE pairs.

MPLS traffic engineering and packet differentiation - or MPLS-TE and MPLS-Diffserv to stay with the popular jargon - should be regarded as orthogonal. In fact, TE can be used without packet differentiation to improve the performances of legacy best-effort traffic. In turn, packet differentiation can be adopted to protect QoS traffic in a pure ”Diffserv over MPLS” fashion. There are also several potential mixed scenarios: in a MPLS/Diffserv domain, TE techniques might be applied to QoS traffic only, while best-effort traffic is handled in hop-by-hop LSPs or simply as native IP. Again, the decision about the more appropriate scenario must be taken considering the particular profile of the network operator.

2.6 Over-provisioning and service differentiation

According to the best-effort nature of today's Internet, user applications can start their data transmission whenever they want, independently of the current network state. Evidently, in such a scenario there is no way to assure hard guarantees for the quality of the started connections. On the other hand, most of the QoS solutions (e.g., IntServ, DiffServ) we have surveyed so far are equipped with add-on traffic control mechanisms, particularly with CAC to ensure QoS. In these architectures, therefore, a user application might get blocked against his/her willingness while trying to enter the network domain.

In today's practice of backbone QoS assurance, the feature that users have a total freedom for the decision whether they should start the connection or not, is retained. That means that there is no connection rejection from the network side. Instead, the users can start the transmission whenever they want. The issue of QoS assurance is resolved by the service providers who are committed to provide sufficient bandwidth by applying over-provisioning coupled with the use of proper, class-based scheduling in routers. Note that traffic classification can be done according to the DiffServ paradigm, i.e. packets carry a header codepoint indicating their traffic class. In effect, it leads to a QoS architecture with the following main entities:

- over-provision of the link bandwidth,
- deploying class-based scheduling to achieve service differentiation and protection for the high priority, QoS-critical traffic class,
- updating the link bandwidth in response of the traffic dynamics.

ISPs often deploy a certain range of over-dimensioning, meaning that the bandwidth utilization in any given link is always kept below a safe threshold (e.g., 50%). By doing this, the network get transparent to the users, i.e. the QoS in the strict sense is highly assured.

In addition, service differentiation can be achieved by traffic classification and by exploiting the scheduling capability in routers. Output ports of today's routers can support more traffic classes, by using proper schedulers. The simplest case is when two queues are implemented, supporting two traffic classes. A non-preemptive, priority queue for the high priority, real time (voice and video) traffic, and another queue for the best effort traffic. Another scheduler type that can support three traffic classes comprises a priority queue for the high priority (e.g., voice) traffic, and a queue managed by the RIO (RED with In and Out) algorithm supporting further two traffic classes. This RIO queue is shared by the best effort traffic and the assured (e.g., premium web-surfing) traffic. Lower dropping thresholds are assigned

to the best effort traffic and the out-of-profile assured traffic, while the in-profile assured traffic has higher RIO dropping thresholds. More sophisticated schedulers can support more than 3 traffic classes by implementing, e.g., an absolute non-preemptive priority queue for the high priority traffic and weighted fair queues for the rest of traffic classes.

A remaining issue of this architecture is the question of resilience, where anomalous events, e.g., link failures might induce QoS degradations. We argue that this architecture efficiently protects the high priority, QoS-critical traffic class from any degradations. Later, in Section 3 we will discuss this issue in more detail.

To accommodate the overwhelming growth of traffic, or more precisely, to follow closely the dynamics of customer traffic, ISPs should have an ability to update the link bandwidth over time scales if necessary. From the technical point of view, the decision on bandwidth updates is a result of two consecutive tasks as described below.

- **Task 1**: measuring and monitoring the traffic load and/or packet losses, and delay in the link. This measurement task should be continually performed as informally indicated in Figure 2.

- **Task 2**: based on the analysis of the collected measurement data, to define the bandwidth amount needed to keep the pre-desired link utilization. The relation between this bandwidth amount and the currently deployed bandwidth will dictate the update decision. This task should periodically take place at an adequately chosen time scale.

The first task can be achieved by exploiting and/or extending the monitoring capability of routers (e.g., using add-on monitoring tools like Cisco Netflow [30], MRTG (Multi Router Traffic Grapher) [31], MRTG++ [32] or using SNMP (Simple Network Management Protocol) to get the traffic load samples in router ports). The preliminary purpose of the second task is to get an as detailed picture as possible about the dynamics of the traffic, enabling a reasonable forecast for dimensioning. To this end, several statistical techniques, e.g., time series, Wavelet multiresolution analysis, forecasting methods etc., can be involved, see e.g., [33]. Concerning physical capacity dimensioning, the relevant timescale of updates is normally in the order of several months. This update time scale is much smaller (in the order of hours or minutes) in case of logical links like MPLS and VPN pipes. In the next subsection we detail one possible provisioning scheme, which derives explicitly the required link utilization based on the traffic dynamics and QoS requirements.

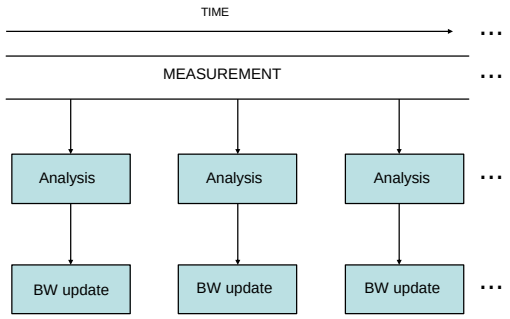

Figure 2: Dynamic resource provisioning procedure

2.6.1 Provisioning schemes

Recall that the simple, but at the same time, quite rough way of provisioning is to adjust the link bandwidth so that the safe link utilization threshold is kept. Nevertheless, when more explicit respect should be paid to QoS, we need a provisioning strategy clarifying the relation between the required QoS and the corresponding link utilization. It will help the providers to have a practical view on the necessary degree of over-provisioning.

To be more specific, suppose that the target QoS we have to keep is the packet level constraint $P(delay > D) < \epsilon$, where D and ϵ are the given delay bound and violation probability bound, respectively. The question is what is the corresponding link utilization threshold assuring this QoS requirement? A tangible solution is to keep the link utilization at a quite low level (e.g., below 50%). With some engineering considerations, however, we can definitely achieve much better resource usage as our recent work [34] has pointed out.

The main contribution of the before-mentioned work is an efficient bandwidth update algorithm used in the analysis module (referred to Figure 2). The approach is based on an adequate traffic model providing a fair mapping between the desirable QoS and the associated link bandwidth, thus specifying explicitly the required link utilization threshold.

Being in the backbone network environment, where the link accommodates traffic with high degree of aggregation, the Gaussian process appears to be a suitable model for traffic description as proposed in several research work [35, 36]. The parameters of the Gaussian model can be deduced from measurement data. Specifically, the aggregate rate of the incoming traffic is periodically collected in

consecutive time slots with fixed length. The dynamics of this trace (mean and covariance functions) are used to define the Gaussian process' parameters.

Exploiting some features of the Gaussian traffic model, we can estimate the evolution of statistical performance parameters like packet delay and packet loss. Based on this estimation, with a simple binary search, we can determine a bandwidth yielding the required statistical QoS with high precision. Thus, with a non-significant add-on analysis, we can perform more efficiently the provisioning task.

As an illustration, trace driven simulation results in Figure 3 show the QoS achieved by different provisioning schemes. In this figure, PS1, PS1*, PS1** are the versions of the Gaussian traffic model based provisioning approach with the increasing enhancements for the involved traffic prediction. The utilization based scheme is the provisioning approach keeping the link utilization at 80%. The Cisco scheme allocates the bandwidth identical to the maximum traffic rate measured in the last period of time. The traffic aggregate is the Ethernet traffic trace [37] and the target QoS is $Pr(delay > 10ms) < 10^{-4}$. It is clearly discernible that the Gaussian model based approach provides QoS closest to the desirable value, while other approaches do not. This is because in case of the utilization based and Cisco schemes we do not have a way to control the QoS, or in other words, there is no explicit relation between the target QoS and the allocated bandwidth, which can be exploited. Also note that the better QoS achievement of the Gaussian model based schemes are not at the expense of significant over provisioning. We refer the readers to [34] for the complete investigations.

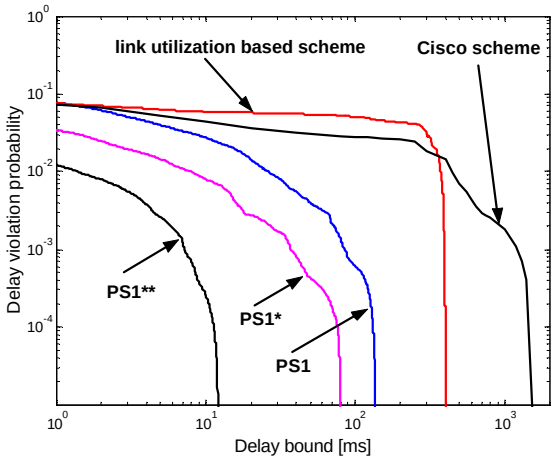

Figure 3: Achieved QoS of different provisioning schemes

An appealing feature of this provisioning approach is that its application is not limited only to the high priority traffic class (e.g., streaming traffic). In fact, we can also use it for the bandwidth allocation of the lower traffic classes (e.g., elastic traffic) as well. Note that for elastic traffic, the perceived quality of service is often construed as the mean file transfer time (or equivalently the mean throughput), and not as the packet level QoS we use in our provisioning scheme. However, note that elastic traffic is mostly carried by TCP protocol stack and there is an explicit relation between the packet loss probability p and the achievable throughput B, such that $B(p) = K/RTT\sqrt{p}$, where K is the constant depending on the second-order statistics of the loss process (e.g., $K = \sqrt{3/2}$ for periodic losses). This means that the throughput-related QoS can easily be conducted from the original packet level target QoS we use for dimensioning. When different packet loss targets are defined for allocations, we can in fact achieve a throughput differentiation between traffic classes.

We can even use the provisioning approach for the mixture of all traffic classes[2] and perform bandwidth provisioning with regard to the most stringent QoS class. At the first sight, such provisioning might cause poor resource utilization, because not all the classes need such a strict QoS. Fortunately, by *exploiting the high degree of multiplexing*, it seems that a quite high utilization can still be reached, thus there are no disadvantages of doing in this way.

A prerequisite for the adequate operation of this provisioning approach is that traffic load measurements must be done at fine granularity (less than 100ms). The standard 5 minute SNMP data is thus not suitable and other measurement procedures should be deployed in routers.

If we would like to apply the provisioning scheme to adaptive resizing of logical pipes (like LSP in MPLS networks), i.e. to achieve *dynamic provisioning*, we need to signal the bandwidth reassignment to the routers along the logical pipe of interest. Each router along this logical pipe should deploy per-pipe based weighted fair queuing scheduling in order to assure the per-pipe bandwidth reservation. Note that if each pipe conveys one certain traffic class or a proper aggregation of some traffic classes then the number of queues is low and remains completely in accordance with today's router facility. Achieving a bandwidth reassignment thus practically means that the scheduling weights assigned to each pipe-queue is correspondingly changed.

In case of MPLS, for the signaling purpose we can exploit the refresh messages in the soft-state RSVP-TE [28], the current signaling protocol of MPLS[3]. Therefore, we do not need additional signaling overhead compared to the current MPLS

[2] In this case, traffic classification and priority-based scheduling become superfluous.

[3] The default period between refreshing messages is 30s but this value can be configurable.

architecture to achieve QoS-aware adaptive provisioning.

It may happen that the bandwidth re-assignment is not possible, i.e. the required bandwidth volume can not be granted due to the lack of link capacity at one or more routers along the logical pipe. In this case, we have to resort to additional solutions in order to keep the QoS of the pipe at the desired level. We can either perform blocking all the new traffic flows that would joint this logical pipe, or we can perform rerouting to find another path with enough bandwidth. Note that the rerouting may concerned only to the newly incoming, and a properly chosen part of existing traffic in the logical pipe.

The application of the proposed provisioning scheme can also be extended to the case of physical capacity planning. With the knowledge about the network topology, the traffic matrix, and the underlying routing procedure, we can dimension each physical link inside the network such that the end-to-end statistical delay requirements that are targeted between each pair of edge nodes are fulfilled. Currently, work is being in progress in this direction.

2.7 Service availability

Network equipment failures or outages caused by maintenance activities usually cause interruptions in connectivity between individual pairs of hosts for a certain period of time. These interruptions are mostly not critical for elastic traffic like TCP, but they prove to be vitally important for services with stringent QoS and availability requirements, e.g., Voice over IP (VoIP), video conferencing, e-commerce, etc. Therefore, irrespective of the paradigm used for achieving *QoS in the strict sense* (refer to the Introduction section), ISPs should implement a comprehensive resilience strategy in order to ensure that all QoS requirements concerning network availability (i.e. *QoS in the wide sense*) can be met.

When deciding on failure recovery strategies and on the concrete mechanisms, ISPs must first choose in which layer these mechanisms should be implemented. The choice here is following [38]:

- *Implementation of resilience mechanisms only in layers below IP.*

  With this strategy very short network recovery times can be achieved, as connection oriented networks usually have sophisticated network management which allows a fast and efficient detection and isolation of failures. The drawback of lower layer resilience mechanisms is that they can only operate at a very coarse granularity: it is not possible to protect individual IP-layer services, but rather only individual links, which inadvertently results in high costs of resilience. Additionally, lower layer mechanisms cannot cover IP layer equipment failures, like e.g., router outages.

- *Implementation of resilience mechanisms only in the IP layer.*

  In the IP layer, resilience is usually assured by routing protocols, the main task of which is keeping all routers in the network up to date with the current network topology. Forwarding tables in individual routers are derived from the available topology information, and therefore it is straightforward that the performance of recovery mechanisms will mostly depend on the propagation speed of topology changes throughout the network, as well as on the timely processing of this information. Additional technologies like MPLS enable very low recovery times, which are comparable to those of lower layer mechanisms, and they also enable selective protection of individual (critical) services, leading to a cost advantage compared to lower layer technologies. In a scenario of selective protection, unprotected services with lower resilience requirements are rerouted by standard routing protocols. A drawback of all IP layer solutions is that even in the case of single link failure, a large number of individual IP flows must be recovered, whereas the lower layers could handle such simple failures much more easily.

- *Combined multilayer approach.*

  Resilience mechanisms may also operate in multiple layers, i.e. both in the IP layer and in the lower layers. The presence of resilience mechanisms in multiple layers may lead to race conditions, meaning that multiple mechanisms may simultaneously attempt to solve the same network fault. This might lead to transient network-wide instabilities in the presence of failures, which may significantly prolong overall network convergence. Furthermore, implementing resilience mechanisms in multiple layers usually also leads to inefficient ”overprovisioning” of backup capacity. Multilayer resilience strategies therefore require interactions between the layers, or at least, awareness of the different resilience mechanisms present in multiple layers.

Apart from delegating the issue of resilience to a particular layer, the efficiency of network recovery in the presence of network faults or maintenance activities also heavily depends on the ISP’s choice of resilience strategies. Basically, there are two different types of strategies: protection and restoration. Protection assumes pre-computing the network’s reactions to individual failures, and usually also assumes making reservations of spare capacity for the diverted traffic in advance. Protection may therefore be seen as an *a priori* resilience strategy. On the other hand, restoration only assumes *a posteriori* reactions to failures, i.e. after that a failure has occurred, without making explicit reservations of spare capacity in advance.
The proactive nature of the protection approach brings some potential advantages

over restoration, which is purely reactive. For instance, it can be expected that protection schemes generally involve shorter recovery delay. On the other hand, some recent works [39] suggest that a well-designed restoration scheme might reach satisfactory recovery delay. The main drawback of any protection approach - admittedly also those introduced in next section 2.7.2 - is that they all rely on the *a priori* knowledge of the full set of potential failures that may occur at the IP layer. In turn, this knowledge should be based on the accurate information about of the deployment at the physical layer, and of the mapping between packet- and physical-layer connectivity. Unfortunately, in some practical cases such knowledge is not assured. This might be due for instance to lack of coordination between different departments within the ISP company (typically, packet-layer and physical-layer staff), or to the fact that lower layer connectivity is provided by a different carrier. In these cases, it is not possible to anticipate the full set of potential failures, therefore it makes no sense to apply proactive strategies (i.e. protection), and purely reactive mechanisms are the only viable solution (i.e. restoration).

In the following subsection we will discuss restoration mechanism at the IP layer, then we will expand on protection mechanisms which are being proposed for MPLS.

2.7.1 Enhanced resilience mechanisms with native IP

In the rest of this chapter we will focus on resilience mechanisms in the IP layer, and we will first examine the standard IP routing protocols. The most widely used routing protocols today like OSPF and IS-IS belong to the group of ”link-state” protocols, meaning that each router conveys information about its ”local piece” of the global topology (i.e. all links attached to the router) to all other routers in the network using a message flooding mechanism [40, 41]. After receiving partial topology information from all routers in the network, each router has got information about the entire global topology. In the case of OSPF and IS-IS, failure recovery (or more precise, *failure restoration*) is basically a three stage process consisting of:

- *Failure detection* – After a link or a node failure has occurred, the router must first detect the failure in order to react to it. If there is no interworking between the IP layer and the lower layers such that the IP layer is immediately notified by the lower layers about the failure, the IP layer will have to rely on the very slow mechanism of HELLO messages exchange between neighbor routers. In the case of OSPF, failure detection usually lasts around 40 seconds.

- *Failure advertisement* – As mentioned above, after failure detection, routers use the message flooding mechanism to convey this information to all other routers in the network. If the routing protocol is implemented well this process should be completed very fast, i.e. in the order of tens of milliseconds.

- *Recomputation of forwarding tables* – After receiving topology information updates, routers recompute their forwarding tables using the *Dijkstra* shortest path algorithm [42].

In [43], it has been demonstrated that for large ISP networks running the IS-IS routing protocols with standard parameter settings (i.e. IS-IS specific timer intervals), the re-computation of forwarding tables takes 5.1 to 5.9 seconds, provided that fast failure notification from lower layers is implemented. Around 1.5 seconds should further be added to this convergence time, which corresponds to the time required for entering the recalculated routing information into the router's line-cards, meaning that IS-IS routing is typically restored in 6.6 to 7.4 seconds in the Sprint backbone network. It is important to stress that the largest part of the forwarding tables' recovery is consumed neither by failure communication through message flooding, nor by the re-computation of the forwarding tables, but rather by IS-IS specific timers which delay individual steps of the recovery process. The main reason these timers have been introduced is dampening the frequency of routing event generation, as originally the processing power of routers was very scarce, such that it was crucial not to overload the routers with too much signaling overhead. Overall, we can conclude that 6 or 7 seconds is definitely a too long period of outage time for services like VoIP or video conferencing, where a large number of calls may be lost due to a single link failure. Better resilience mechanisms for IP networks are therefore needed if very stringent VoIP-enabling Service Level Agreements (SLAs) must be supported. Recently, a very attractive solution for improving native IP routing protocol's resilience mechanisms has been implemented in the Sprint backbone IP network [44]. Essentially, the IS-IS *notification timer, LSP generation timer, and shortest path computation timer* are set to minimal purposeful values (1-10ms), which dramatically reduces network convergence times. It has been shown that with such parameter settings less than one second is needed for full routing recovery in the Sprint IP backbone network after a fault has occurred.

In [45], a local restoration technique which operates within the framework of the standard IP routing is introduced. The fundamental idea is to exploit the presence of multiple viable paths in each node by reacting strictly locally to individual links outages instead of launching the slow and resource consuming global routing recovery. Note that such a recovery process only requires the deletion of one

next hop per destination in the router's forwarding table, which can be performed practically instantaneously upon the detection of the failure. In order to enable the operation of such a scheme for all possible cases of individual link failure, the network topology (i.e. the network graph including link weights) must provide the following property: *in each node, at least two different next hops must be available for reaching each destination in the network.* It has been shown that this property can easily be fulfilled if at least two paths of equal hop-lengths exist between all individual pairs of nodes in the network, provided that very simple rules are followed when setting link weights [45]. It is well known that ISP networks are normally built with a very high degree of redundancy, such that all mentioned criteria can easily be fulfilled in most networks. This raises the attractiveness of this approach compared to the standard IP re-routing, and its efficiency in terms of reaction time and signaling further make it a strong competitor to more complicated solutions, which will be presented in the following paragraphs.

2.7.2 Resilience mechanisms with MPLS

An alternative solution would be to offload the IGP protocol from the task of recovering failure. This can be done for example in IP/MPLS network, by exploiting the MPLS Fast Restoration capabilities. The easiest way to implement MPLS Fast Restoration is in the form of link-protection (also called local-protection). Consider a generic link from router A to B. A detour backup LSP is preliminarily installed between A and B. Upon failure of this link, node A changes its forwarding tables and sends the packets into the backup LSP. This method is already operatively used by some operators in their MPLS networks. The recovery delays are reported to be in the order of few hundreds of milliseconds. The exact value depends on the size of the forwarding table, which is roughly proportional to the number of LSP sharing the physical interface.

The advantages of this scheme are following:

- *Accuracy* – The operator can decide which link to protect at the MPLS layer, and which not (as for example some links might already be protected at lower layer).
- *Simplicity* – It does not require far-end failure notification, since the switching node is local to the failure.
- *Scalability* – The number of backup LSP scales as the number of physical links.

The disadvantage is mainly that this scheme does not protect against router failures, since the link end points are in common with its backup LSP. Another

point of weakness is that it is not possible to deliver resilience differentiation in a simple manner. Resilience differentiation means that only a subset of LSPs access strict survivability guarantees through the FR mechanism, while other LSPs are left unprotected or depend on other mechanisms for restoration.

In order to achieve fast recovery in presence of router failure, more sophisticated techniques such as segment-based or path-based protection are required.

With path-based an end-to-end disjoint LSP is associated to each working LSP. Upon occurrence of a failure, the detecting node must notify the event to the ingress edge nodes, which switch packets from the interrupted working LSP onto the associated backup one.

The main advantages of path-based protection are: i) protection against node-failures, ii) Fine-grained resilience differentiation on a LSP-by-LSP basis [46, 47], iii) potential bandwidth saving by means of sharing backup bandwidth.

On the other hand, path-based protection requires notification mechanism to notify the head-end node about the failure, since the switching node in general differ from the detecting node. Additionally, more advanced route-selection capabilities are required for setting up disjoint LSPs. In fact, working and backup LSP must meet complex fault-disjointedness requirements that generally do not reduce to link-disjointedness due to the presence of Shared Risk Link Groups (SRLG). A SRLG is a set of links that can be potentially interrupted by a single failure event. This might be due for esample to non-complete spatial diversity in the underlying physical deployment. Most critically, since the working and the backup LSP can not share a same path, they can not be built hop-by-hop with LDP, but rather must be explicitly routed and installed by RSVP-TE. As discussed above, in those applications involving any-to-any traffic exchange between all PE pairs this requires a full mesh of ER-LSP and associated signaling sessions. This scenario, as stated above, might raise scalability concerns.

An intermediate solution between link- and path-based protection has also been considered, namely segment-based protection [48, 49]. Such techniques have gained considerable interest by the research community and by IETF, particularly path-based protection, but to the best of our knowledge only link-based protection has been operatively deployed in commercial networks to date.

3 Analytical discussions on the QoS solutions

Given the QoS solutions described in the preceding sections, we discuss now some emerging issues concerning their scope, deployability and vision. Without stating that the completeness is covered, we strive here to tackle some most frequently raising issues and arguments.

Issue 1: Today's Internet as a sufficient QoS architecture for voice communications

Today's Internet structure exhibits deficiencies at the packet level. Packet loss and jitter hamper the delivery of decent QoS of VoIP at the user level *in an Interdomain structure*. Still, these deficiencies can be concealed by employing appropriate signal processing of the data to be delivered and the data that is received.

Global IP Sound (*http://www.globalipsound.com*) has developed an integrated solution to deal with lost information and varying delay. Their codecs provide graceful degradation in case of packet losses and the NetEQ system adapts the jitter buffer constantly in order to reduce the number of lost late packets and reduce the end-to-end delay.

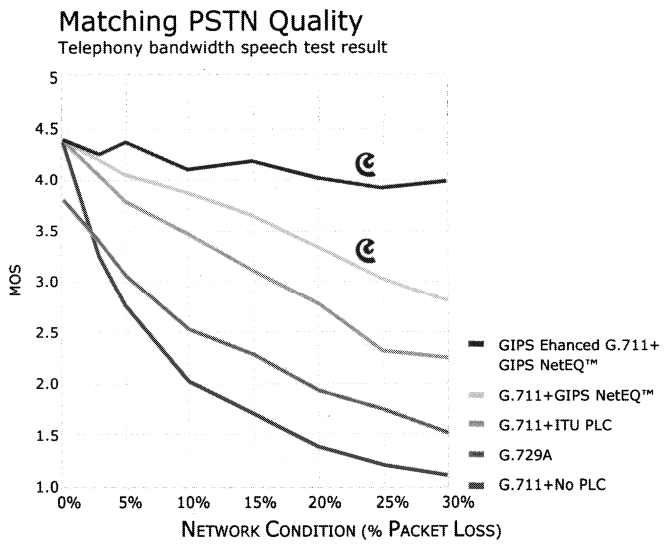

Figure 4: Perceived Quality of GIPS VoIP signal processing solutions (Source: Lockheed Martin Global Telecommunication (GIPS/COMSAT)).

The performance of such a system for random packet loss is depicted in Fig-

ure 4. The perceived quality of the G.711 codec (64 kbps, deployed in our digital, circuit-switched telephone network) without any packet loss concealment is compared with a low bitrate codec (G.729, 8 kbps) and with variations incorporating signal processing for receiver-based loss concealment (and play-out buffering).

These results clearly show that high speech quality can be gained even at very high packet loss rates up to 30%. The perception of end-to-end delay highly depends on the level of echo, either resulting from acoustic coupling at the user terminal or electric coupling at a potential IP/PSTN gateway. Echo cancellers are able to suppress these artifacts and thus may reduce the impact of the end-to-end delay.

For low bit-rate IP-links exhibiting packet losses, the Internet Low Bit-rate Codec (iLBC), working at 13.3 kpbs and 15.2 kbps for 30ms and 20ms speech frames, respectively, has been developed and is about to get a proposed standard of the IETF [17]. Code Excited Linear Prediction (CELP)-like speech codecs working at low bit-rates usually maintain an internal state that results in an interdependency of consecutive speech packets. The iLBC uses block-independent linear predictive coding, so the impact of a lost packet is greatly reduced. Thus, this codec enables graceful degradation of the speech quality in case of packet losses. A performance comparison of low bit-rate codecs used for VoIP is presented in Figure 5.

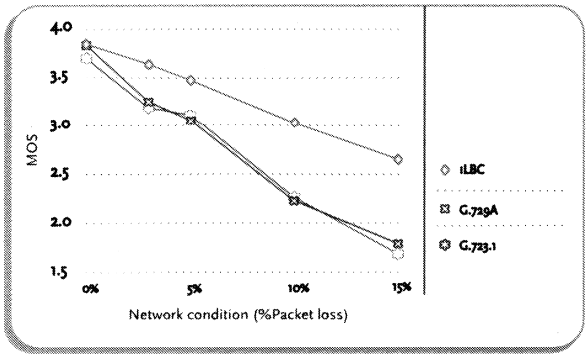

Figure 5: Comparison of the perceived speech quality of low bit-rate VoIP codecs (Source: GIPS/DYNASTAT).

An issue that has to be clarified is the proper working of signaling at high packet loss rates. It is expected that the call-establishment would take longer at such conditions, which would influence the quality perceived by the user to a certain extent. Furthermore, signaling translation at IP-PSTN gateways may become troublesome if the protocols do not provide robust mechanisms. The impact of signaling problems on the perceived quality needs to be determined in subjective conversational tests.

In conclusion, we can say that the technology for the deployment of good VoIP QoS in an Inter-domain Internet structure is at hand.

Issue 2: Should over-provisioning be a more viable and more preferable QoS solution than other alternatives?

If the cost for having abundance of resources is negligible, then the answer is definitely yes. With this solution no significant router-upgrades and router-developments concerning both control plane (e.g., admission control) and data plane (e.g., sophisticated scheduling schemes) are needed as in the case of resorting to IntServ or DiffServ. With the developments of new transmission technologies (e.g., Dense Wavelength Division Multiplexing), it might be expected that the bandwidth will become cheaper, encouraging the use of this architecture. Another advantage is that for the specific case of QoS sensitive, voice traffic, the over-provisioning degree assuring a target QoS is relatively easy to be planned, because the traffic characteristics of voice applications have been quite well-understood and analyzed.

Over-provisioning is the simplest option for QoS provisioning. This is well reflected in today's practice that most of large-scale ISPs vote for over-provisioning solution (see for example the Sprint's network [50]). To see that QoS is indeed well achieved, let's consider the case of the QoS-critical real time traffic (e.g., voice and interactive video) class. This traffic class is often treated as the high priority class because it requires very low packet delay, jitter and packet loss rate. It is well know that this traffic class constitutes only a small (around 10% or even less) fraction of the total traffic. Thus, if the total over-provisioning degree in a given link is less than 50%, the over-provisioning degree for the real time traffic class is much higher. Even in case sudden congestion events take place because of, e.g., hot spots or link failures, such a high over-provisioning degree is far sufficient to deliver strict QoS assurances. In fact, many studies in the operating network, e.g., [51, 36], show that the end-to-end quality requirements are excellently fulfilled if the link load along the path does not exceed a well-defined level. Extensive measurement results have also shown that the per-node packet delay at an OC3 link is about only 1-2ms when the link utilization is kept under 50% [52]. In case of connectivity outages due to link or node failure, efficient resilience mechanisms will assure the QoS

satisfaction of QoS sensitive traffic as discussed in Issue 5 in this section.

Naturally, from the perspective of providers, it is better to keep the degree of over-provisioning under control. Therefore, there is a scope for developing efficient bandwidth provisioning schemes, helping ISPs to economize better their resource treatment. The scope has a broad vision and embodies a lot of open issues. Our provisioning concept presented earlier in Section 2.6.1 is in fact one step toward the solutions of the emerging issues of such efficient resource management framework.

Issue 3: QoS solutions in the access environment

In some situations, the incentive for having bandwidth over-provisioning is limited. This is typically true for the access network environment and for small ISPs. In the former case, the customers may own the access link connecting its premises with the edge router of the core network. Clearly, although they would like to get QoS, they don't want to pay for the over-provisioned link. In the latter case, the small ISPs may have to buy the bandwidth from the larger ISP partners and thus, for the profitable operation, these small ISPs still have to think about the bandwidth investment they can afford.

When an access link is not over-provisioned, class based scheduling itself in routers is not enough for providing QoS. For example, even with the strict non-preemptive scheduling, lower priority traffic has considerably detrimental effects on the quality of high priority traffic (see e.g., [53]). This in turn means that in order to deliver QoS assurances with non-plentiful network resources, addition networking mechanisms are needed. IntServ, for example, with its resource reservation and explicit admission control mechanisms, offers a good QoS solution in a small network scale. For larger network scales, DiffServ means an alternative solution, including not only traffic classification and class based scheduling, but also traffic policing and admission control.

Issue 4: Do we really need MPLS for QoS and TE?

Before trying to answering to this issue, one should clearly have in mind that different flavors of MPLS can be envisaged as discussed in Section 2.5. For instance, it is possible to exploit MPLS just as a means of encapsulation in support of VPN, with a full mesh of hop-by-hop LSP between all PEs (Provider Edges, i.e. edge routers with VPN capabilities and VPN customers attached), or eventually between all edge routers. Such a full-mesh can be dynamically created by LDP message exchange between neighbor routers, and does not involve RSVP-TE signaling sessions. On top of such platform, one can apply packet-level prioritization in a pretty similar manner as it can be done with pure DiffServ. More precisely, one is ap-

plying the Diffserv approach on top of a MPLS cloud - or more likely on top of a mixed native IP / MPLS cloud in case that public traffic keeps to be carried as native IP. It is clear that in this simple scenario, MPLS was not at all introduced *because* of QoS.

While in the scenario depicted above MPLS has an ancillary role in support of VPN, there are more sophisticated scenarios in which MPLS absorbs more and more functionality, and assumes a central role in the network operation. For instance, one can introduce Explicitely-Routed LSPs (ER-LSP) in support of some smart Traffic Engineering scheme and / or bandwidth reservation capability. This approach comes along with some potential advantages and drawbacks. The drawbacks are that ER-LSP are point-to-point in nature and scale as N^2 (N is the number of involved edge routers), while hop-by-hop LSPs created by LDP are instead destination-oriented and scale as N. Also, ER-LSPs require RSVP-TE signaling, which is a far more complex protocol than LDP: it introduces more state in the routers, and has a more complex state-machine. Concerning LSPs with bandwidth reservation capability the loss of statistical multiplexing can be mentioned as an additional drawback. Today's connectionless Internet multiplexes all flows sharing a link, in other words there exists no fine grained division of a link's capacity into small edge-to-edge or customer pipes. As many flows are multiplexed on high capacity links, and as the temporal characteristics of the flows are stochastic in nature, bursty Internet traffic is accommodated minimizing the probability of packet loss. This important advantage of a connectionless Internet is clearly lost in case of an MPLS architecture employing LSPs with bandwidth reservation.

In front of such drawbacks, the introduction of ER-LSP brings in some *potential* advantages. Case by case, depending on the particular conditions of the network operator, these advantages may or may not be important. If not, they probably do not pay off the loss of scalability and additional complexity.

Among such potential advantages there is the possibility to introduce some advanced forms of end-to-end protection to contrast failures. This point is discussed in Issue 5.

A QoS-related potential advantage is the possibility to optimize routing, so as to achieve some savings of bandwidth with respect to overprovisioning schemes. This point may be important for those operators that find bandwidth in the network core expensive, typically because they buy it from third-parties, but this is unlikely to be the case for bigger carrier and ex-incumbent. And even in that case, some alternative solutions based on the connection-less paradigm that are being recently developed might become fierce competitors of MPLS-based solutions (e.g., tweaking IGP weights [54, 55], or perform adaptive load balancing [56, 57]). In fact, the connection-less approaches to traffic engineering might be recognized more scalable than traditional connection-oriented schemes, while at

the same time equally perform [58]. On the other hand, as a matter of fact, these proposals for connection-less traffic engineering (denote by cl-TE) are rather novel to the networking community, and network operators are definitely more familiar with connection-oriented optimization techniques (denote by co-TE). At a more abstract level, the know-how for co-TE has been inherited from the past, since c.o. technology has been long dominating in the core network (Sonet/SDH, ATM), while know-how for cl-TE is growing up in the very last days [54].

Issue 5: Which paradigm for network resilience should ISPs pursue?

Several different strategies for resilience have been presented in Section 2.7. In the IP layer, which is in the focus of this paper, we can subdivide the available restoration and protection approaches into two major groups:

- *Traditional pure-IP restoration.* These approaches are based upon the traditional IP network architecture, i.e. upon path calculation based on link weights and routing protocols for the distribution of topology information throughout the network domain. The standard IP re-routing approach and approaches which focus on local restoration of IP routing are representative of this group.

- *MPLS-based protection.* Many different mechanisms have been developed, like e.g., *link-, path-, and segment-protection.* All these approaches have in common that concrete network reactions to expected failure events have to be pre-computed in advance.

In recent years, MPLS based solutions have been strongly pushed by equipment vendors. One of their main arguments was that MPLS offers the potential for immediate and optimal protection in case of failures. This has lead to extensive research and the development of various MPLS-based protection mechanisms, shifting the focus of research and engineering away from traditional pure-IP based solutions. However, recently in [39, 43] the potential of achieving restoration on the time-scale of milliseconds by employing only pure-IP techniques has been opened, which again moves the traditional technologies into the spotlight of interest. Additionally, [45] has demonstrated that for many realistic IP networks novel schemes are feasible, which enable a local restoration of routing within the framework of the traditional IP architecture. The proposed local restoration scheme is especially well suited for the case of transient link failures, as it does not necessitate a global re-computation of routing tables twice (i.e. in the event of failure, and after the repair), but instead keeps the reaction local, i.e. in proximity of the failure.

In general, we identify following arguments in favor of pure IP-based solutions:

- It is impossible to pre-compute network reactions to all realistic faults scenarios for MPLS-based solutions, as the underlying problem is combinatorial, and therefore NP-hard. For example, a commonly occurring event like a single fiber cut may cause multiple link failures in layer 3, which might correspond to thousand os LSPs. Such problems can normally not be addressed by protection schemes, as the topology of the layer 2 network might change dynamically, making the computation of network reactions to all possible combinations of correlated link failures impossible. Therefore, most deployed MPLS-based protection schemes do not provide a consistent solution for resilience.

- Protection schemes introduce a significant state-increase in the network. This makes the network more prone to errors and inconsistencies, and it significantly increases efforts required for network management.

- IP-based schemes do not require the deployment of any new protocols and technologies which might become additional sources of errors and failures. Instead, they only require a smart tuning of current routing protocols and/or link weights.

When considering different solutions for resilience, a very important metric is also the amount of spare capacity required. Advocates of MPLS-based solutions, like e.g., path-protection switching, claim that typically only around 20% extra capacity is needed for ensuring resilience against all single link or node failures [59]. Being able to re-route the traffic far ahead of the point of failure, i.e. at the entry into the MPLS cloud, path protection switching certainly has got more potential for minimizing the amount of spare capacity than link protection or local re-routing - but as previously explained, this of course only applies to a small number of tractable failure scenarios.

We believe that the importance of ensuring full bandwidth availability in the presence of failures is not the critical requirement in today's networks. In our view, the most important performance metric is the network recovery time, and possible degradations of available bandwidth are only a secondary issue. The main rationale behind this is that the vast majority of Internet users currently requires only best effort Internet service (today's Internet traffic is mostly comprised of TCP flows), such that it is very unlikely that they will be annoyed by slight bandwidth degradations - in most cases users will just observe slightly longer Web page or file download times. In contrast to that, the same users might become irritated if they experience long outages of Internet service of, e.g., 40 seconds or even longer. This advances network recovery time to become the most important metric for resilience, and increases the attractiveness of local resilience schemes, because they

offer the greatest potential for ensuring fast network recovery. Furthermore, it is well known that links are usually loaded less than 30% in typical ISP networks, which means that in the case of single link failure we should never observe link utilizations exceeding 60% in these networks. Supporters of MPLS often argue that, in contrast to plain IP routing, MPLS opens the potential of protecting only individual, high-priority services, like e.g., *Virtual Private Networks (VPNs), VoIP calls, video-conferences, e-commerce applications*, etc. In contrast to that, we believe that IP routing offers an even more simple strategy for service differentiation: basically it suffices to introduce only two *DiffServ* classes of service: high priority and best effort. The high priority class should include real-time sensitive services, while the best effort class should encompass the rest of the traffic which is elastic.

If we consider realistic scenarios in which the high priority traffic class comprises only a fraction of the total traffic, it is clear that high priority traffic will retain a very high degree of over-provisioning even for extremely malicious cases of correlated network faults.

Issue 6: Mechanisms and applications of internet traffic monitoring

Provisioning Internet connectivity with assured QoS is recognized as an attractive source of revenue by many ISPs. However, irrespective of the QoS paradigm employed, *precise evidence of compliance with active SLAs* is required in order to charge customers for QoS. SLAs usually specify QoS in terms of technical constraints like end-to-end delay, delay jitter (i.e. delay variation), and packet loss probability. On the other hand, activities like *traffic engineering and network capacity planning* also require very detailed knowledge about the current traffic. For example, the majority of traffic engineering approaches require information about link utilization and packet loss probability on individual links and paths. Overall, we can conclude that extensive measurements of different traffic parameters are required in order to manage modern ISP networks.

There are many different methodologies for measuring Internet traffic. They can basically be subdivided into the following three groups:

- *Passive, hardware-based measurements.* With this type of measurements, optical splitters are usually installed on OC-48 or OC-192 links (2.5 and 10 Gbit/s, respectively) between individual POPs, and traffic is traced such that each packet's headers (in the case of native IP networks usually IP + transport layer) and the first few bytes of application data are stored on a hard disk. Traffic traces are then collected from different measurement points in the network, after which they are evaluated using sophisticated post-processing procedures. The following data can be precisely derived from such traffic measurements:

  - *Multi time-scale load information for the measured link,*
  - *Spatial and temporal distribution of traffic.* Please note that in order to obtain information about the spatial distribution of traffic, measurement data must be processed together with routing data. Normally, this has to be performed manually, as hardly any generic software solutions are available [29].
  - *Consistency of IP traffic.* Precise information about the absolute and relative volumes of traffic can be derived by protocol type and applications.

- *Software measurements in native IP networks.* An alternative to passive measurements is to measure traffic using software solutions integrated into network equipment, like e.g., *SNMP link utilization data* or *NetFlow* in Cisco's IP routers [60]. However, these solutions usually provide only coarse grained analyses, as processing power in routers limits the possibilities for real-time traffic evaluation. For example, Cisco NetFlow can only derive statistics of IP traffic on a per flow basis. Nevertheless, it has been shown that if used with care, NetFlow can be a valuable tool for network measurement [61].

- *Software measurements in MPLS networks.* Unfortunately, it is a common belief that deploying a full mesh of Label Switched Paths (LSPs) in MPLS networks increases the computational efficiency of real-time software measurement compared to native IP networks. In contrast to that, we believe that software traffic measurements of LSPs are just as (in)efficient as measurements of native IP traffic. Arguments in support of this are:

  - In both approaches, routers basically have to update a minimum of one traffic counter per egress POP for each incoming packet. The source address of packets should additionally be checked in the native IP approach in order to discriminate between traffic originated locally from the POP, and the transit traffic.
  - Similarly, both native IP and MPLS routing require that the IP address of each packet is matched to an egress POP. In the native IP approach this procedure is necessary for choosing the next hop of the packet, whereas in the MPLS approach it is required for placing the IP packet into the appropriate LSP.

In contrast to this, we do believe that it is easier to derive traffic matrices with MPLS when passive measurements are performed, as POP to POP traffic can easily be identified by the packet label. This is of course only true if LSP labels remain static (or at least tractable) during the entire measurement period.

Issue 7: Virtual Private Networks (VPNs)

Several operators are interested in MPLS mainly as a means to provide VPNs for their customers. The reasons for this are:

- *MPLS-based VPNs are highly scalable*, as they do not necessitate the establishment of site-to-site peering. Additionally, emerging LAN branches can easily be integrated into existing VPNs.
- *VPN management can be outsourced from customers to ISPs*, which is particularly attractive for service providers, as it might increase their revenues. It might also be interesting for customers with low in-house IT resources or skills.
- *Precise intra-domain QoS and bandwidth configurations can be provisioned*, which opens the potential of service and pricing differentiation.

However, there are attractive alternatives to using MPLS for VPNs, like e.g., native IP solutions based on the IPsec technology. The basic difference to MPLS is that IPsec VPN-related activities are usually not outsourced to ISPs. Instead, traffic is encapsulated into additional IP headers and encrypted at the customer's site, after which it is forwarded in native IP tunnels across the network to another site of the same customer. There it is decapsulated, decrypted, and forwarded into the remote LAN [62, 63]. IPsec-based VPNs have the following properties:

- *Modern IPsec-based VPNs are very scalable*. Originally, IPsec tunnels had to be manually configured between different customer sites, which required extensive planning and coordination for large scale IPsec-based VPN deployment. In recent years, many solutions have appeared which automate the process of VPN establishment and maintenance, enabling this type of VPN to scale more easily. However, due to management overhead and the requirements imposed on customer premises equipment, there might still be practical limitations concerning the deployment of fully meshed IPsec VPNs with over 1000 sites [63, 64].
- *Authentication and confidentiality is left up to the customers.* No special trust relationship between the customers and the ISPs is required, as all security related issues are resolved at the customer's site. The ISP's role is limited to the provisioning of Internet connectivity.
- *Customers can easily change ISPs, as their VPN system is ISP independent.* Additionally, they can also employ multi-homing to multiple ISPs in order to increase redundancy and network performance. In the case of MPLS VPNs,

customers are firmly bound to their ISPs, as their VPN system is outsourced to the ISP. This means that the entire VPN would have to be re-engineered if the ISP is changed. Even more importantly, for global companies with branches scattered across many different countries around the world it should be almost impossible to find an ISP with such a global reach. Therefore, companies with large scale global operations cannot employ MPLS for constructing their VPNs, as inter-domain MPLS solutions are not yet available.

- *In most realistic cases, IPsec-based VPNs offer QoS which is just as good as that of MPLS VPNs.* If we consider today's ISPs, whose networks are usually overprovisioned, it is clear that switching LSPs with bandwidth reservations will not lead to an increase of QoS. Hence, in those networks intra-domain VPN solutions based on MPLS will not be able to offer higher QoS than IPsec-based VPNs. Furthermore, as there are still no standardized solutions for establishing inter-domain MPLS networks with QoS guarantees, the use of best effort IPsec tunnels is practically obligatory for inter-domain VPNs.

Issue 8: How could the inter-domain, end-to-end QoS be achieved?

So far we have mainly dealt with intra-domain QoS, i.e. QoS inside a single administrative system AS. Within one AS, all the resource management rules, actions are in charge and under control of one concerted operator. In today's Internet, however, user applications are often materialized in an inter-domain manner, i.e. their traffic traverses through several AS-es. As a consequence, inter-domain QoS provisioning becomes a challenge because resources are handled by diverse management systems. Thus, making the question how could end-to-end QoS be assured in such a multi-AS scenario is really pertinent.

Up to date, it is technically envisaged that end-to-end QoS would be offered by using either one of the two following architectural solutions:

- the peer-to-peer concept, or
- the overlay network concept.

The peer-to-peer concept exploits the cooperation between the peer AS-es crossed by the application connection. It introduces a two tier management paradigm as follows (see e.g., [65]). Within a given AS, QoS assurances are in charge of its own operator. The way a given operator achieves QoS guarantees in his AS is independent of that other operators do in their AS-es. This means that the QoS solution may change from AS to AS. For example, the operator in one domain can choose the DiffServ architecture as a scalable QoS solution and deploy in addition

a bandwidth broker entity for further resource management tasks. Another operator in another domain, however, can choose over-provisioning to achieve QoS inside his network.

In order to exert control over the inter-domain QoS evolution, there is a need for the service agreement between the peer AS-es' operators. This agreement is made with regard to the QoS and is usually specified in the form of bilateral SLAs between neighbor domains. Staying with the example that each domain deploys DiffServ, then each autonomous domain has its own Bandwidth Broker (BB). The bandwidth brokers keep the SLA negotiated a priori between their AS-es and peers. In this way, the bandwidth brokers can collaborate in resource management tasks. Namely, they can perform admission control for traffic flows crossing the AS-es. Each BB checks the resource availability in its domain and contacts the neighbor BB to checks the acceptance based on the preset service agreement contract between the two domains.

The cooperation between AS-es for QoS-aware inter-domain resource control can also be resolved with specific scalable solutions, rather than that based on BB involvements described above. An example is the BGRP-P (Border Gateway Resource Protocol Plus) framework [66]. In this architecture, resource management is made in a scalable manner relied on the destination based aggregation principle, i.e. the so called sink tree concept. The operation makes use of the route aggregation property of the inter-domain routing protocol BGP. Scalability is obtained by the beforementioned aggregation strategy and is further enhanced by the mechanism called Quiet Grafting, which enables an early completion of resource request processing [66].

The main technical challenges of the peer-to-peer concept stem from the heterogeneity and scalability. In fact, heterogeneous management rules of distinct providers make the coordination and cooperation hard. As always expected when dealing with a large network and traffic scale, scalability emerges as one of the most important perspective in the inter-domain administrative framework. These aspects mean hurdles on the specification of peer-to-peer SLAs. To overcome these hurdles needs certainly more time and efforts, which is the main reason slowing down the end-to-end QoS materialization in the worldwide Internet.

An alternative way for end-to-end QoS relies on the concept of *service overlay network, SON* (see e.g., [67, 68]). The main idea is that there is an additional ISP on top of several existing network domains. The SON is a network of *service gateways* which perform service-specific data forwarding and control functions. The SON purchases resources ensuring certain QoS guarantees from the underlying domains to form its own *virtual links* between the service gateways. This is done by specifying bilateral SLAs between the SON and the underlying domains. Users now have a direct contract with the SON to obtain services with QoS guarantees.

Note that one advantage of SON is that it bypasses the peering points between the network domains, and thus avoids the beforementioned heterogeneity problems.

The principle of having a SON network, however, necessities further work items. Among others are how to determine the optimal topology of the SON, what is the proper bandwidth the SON should purchase from the underlying networks to make its operation profitable, what about implementation and technical feasibility of the SON (e.g., its gateways), how to perform efficient routing in the SON etc. All of these issues have not been completely well elaborated and standardized, keeping the development process going on.

Ultimately, the goal of having end-to-end QoS seems to be still far from the current stage. The complexity of this task, understood in both technical and economic senses, only allows progress to take place in a step-by-step manner. General guidelines and practical specifications of technical solutions for inter-domain QoS and the related inter-domain traffic engineering task are basically work in progress, see for example [69, 70]. Within this process, we are optimistic that we can contribute our efforts to surmount certain emerging difficult subtleties. For example, we are extensively discussing the role of the inter-domain routing protocol BGP in optimal traffic engineering.

4 Conclusions

The deployment of quality of service is clearly not a technical problem anymore. A rich set of QoS mechanisms like advanced scheduling, packet marking and dropping, signaling protocols like RSVP-TE are implemented in commercially available routers. So given this technical feasibility, why are network QoS mechanisms still rarely activated?

As a partial answer to this question one should mention that due to the global nature of Internet traffic and because a QoS deployment's value increases with its scale any local deployment of QoS is of limited interest. Global QoS, however, suffers from a deadlock situation: for any single ISP it is of limited interest to start implementing QoS because other ISPs don't have QoS as well. So how can a critical mass of ISPs be convinced to offer service differentiation and QoS? The last years have shown that the peer-to-peer model (see Issue 8 in Section 3) has failed in giving ISPs an incentive for QoS. On the other hand, given the success of current peer-to-peer overlay networks, one might be slightly more optimistic for the overlay model for future implementations of global QoS services.

Another answer may be based on the fact that ISPs have consider issues like availability and resilience in the presence of link or node failures as more important than guaranteeing end-to-end QoS parameters like delay. This point has been

reasonable in the past given the asynchronous nature of major Internet Applications like WWW or file sharing. Today VoIP becomes more and more deployed, thus both, resilience and QoS guarantees are important. In Section 2.7, this report provides an overview on various techniques towards fast network convergence. Among those connectionless approaches based on tweaking of IGP parameters and connection-oriented approaches like MPLS local and global protection and restoration can be mentioned. Both approaches are approximately equally balanced concerning their pros and cons, as explained in Issue 5, of Section 3. Obviously, their state in deployment is different. In recent years a lot of research has been devoted to MPLS resilience resulting in commercially available solutions. Pure IP based solutions have been neglected and are thus currently an important topic in research.

A third answer to the above question is based on the fact that bandwidth is extremely cheap in the core network. So why implement sophisticated QoS mechanisms in the core implying high costs in terms of management, more failures due to higher system complexity, and higher load on core routers which are the real bottlenecks in optical networks? Measurement studies have shown that edge-to-edge QoS can indeed be achieved by pure overprovisioning based on concise measurements and network planning (see Issue 2 in Section 3). This argument is additionally supported by the tremendous advances made in the area of enhanced codecs for VoIP (see Issue 1 in Section 3). If there exist codecs which are capable of handling 30% packet loss without significantly degrading the perceived quality of voice, the most QoS demanding application, why then bother about QoS mechanisms in the core network?

Combing the problem of QoS provisioning with the problem of availability might give an answer to the latter question. Pure overprovisioning without any additional mechanisms might indeed be a sufficient solution in case the network operates in its normal state. However, link and router failures occur frequently. In such cases QoS sensitive applications demand most importantly fast re-convergence of the network to ensure minimum outages in connectivity (see Section 2.7 and above paragraph). A second order, but still very important issue, is the guarantee of SLAs (for instance packet loss probability per time window) during periods of non-recovery from link or node failures. In such cases, a combination of DiffServ-like packet marking, overprovisioning, high degree of statistical multiplexing, and priority scheduling for QoS sensitive traffic offers a simple, but yet effective solution. Given the fact that QoS sensitive traffic only accounts for a small percentage of a link's total traffic during normal state, it can be expected that the link's load due to QoS sensitive traffic in times of failure is still smaller than its capacity. Thus simple priority queuing suffices to make SLAs hold and hide the failure from the perspective of QoS sensitive traffic. Additionally, SLAs (and thus traffic characteristics of QoS sensitive traffic) are known in advance. Thus, on the contrary to best

effort traffic, it is possible to engineer networks for the transport of QoS sensitive data even in the presence of link failures. How this engineering is done in an optimum and connectionless manner is currently a hot topic in Internet research. We refer to Section 2.6 and Issue 2 in Section 3 for further details.

Based on IP and Tag switching proposals, MPLS was originally intended to offload core routers by replacing forwarding table lookups by label swapping. However, successful research on efficient algorithms for table lookups has dropped this argument in favor of MPLS. Subsequently, several myths concerning the abilities of MPLS to support QoS and traffic engineering arise, which have been de-mystified to the biggest extent. It is widely known today that MPLS per-se is not a sufficient QoS architecture as edge-to-edge pipes by no means guarantee per end-to-end flow SLAs. Only a combination of MPLS with QoS technologies like DiffServ and provisioning based on concise measurements can guarantee per flow QoS. Such an architecture could be implemented with MPLS only in support of load balancing due to explicitly routed LSPs or with MPLS in support of QoS by assigning fixed capacities to LSPs. In both cases, RSVP-TE is required, raising concerns on scalability. As explained in Section 3, Issue 4 identifying a few high capacity LSPs to be re-routed in order to solve this scalability problem seems to be a promising field of research in the domain of MPLS in support of load balancing. In any case, assigning bandwidth guarantees to LSPs imposes severe headaches concerning loss of statistical multiplexing, one of the most desirable features of today's best effort Internet. Additionally, bandwidth guaranteed LSPs might have problems in holding their guarantee in the presence of interdomain link failures, often requiring to change the egress POP. For reasons of non-scalability and management complexity we clearly do not recommend any flavor of MPLS allowing numerous per customer LSPs.

As a consequence of arguments as summarized in the above paragraph, the research community has always been sceptical concerning MPLS. So what is the case for the success of MPLS in terms of deployment in Autonomous Systems? To some extent major equipment manufacturers did a good job in exploiting MPLS myths to the benefit of their own income. Another reason may be based on the mentality of engineers in European Telcos, who might still prefer to think in terms of "strings" (or in the connection oriented paradigm as well known from PSTN) instead of "clouds" (or in the connection-less paradigm as advertised by pure IP). A third argument often raised in favor of MPLS is the need for VPNs. Scalable MPLS VPNs supporting Security and intra-domain QoS have been on the market very early and have definitely created a strong impulse for ISPs to go for MPLS. In the meantime, as shown in Issue 7, Section 3, scalable IPSec-based VPNs have been successfully commercialized and compete against MPLS-based VPNs. The main difference in the two approaches lies in the assignment of responsibilities. In

case of MPLS-based VPNs, the first tier ISP is responsible for the security and QoS of pipes rented to a large customer owning several sites at various locations. In case of IPSec-based VPNs, the customer itself manages its VPN. Thus, the provider can easily be changed without having to re-configure the VPN, and no trust relationship with the ISP is required, which might be considered as advantages. IPSec based VPNs traditionally do not care too much about QoS. However, for multinational companies this is not an issue because MPLS based inter-domain QoS has not standardized anyway. For national companies with several sites, an ISP offering full meshes of customer edge LSPs with bandwidth assignments for creating VPNs would run into the scalability and loss of statistical multiplexing problems explained above.

Although the topic of QoS has been researched for many years, network support for QoS sensitive applications and services is still rather the exception than the rule. Undoubtedly, a major part of QoS-related research has been too far off the practical requirements of ISPs, as demonstrated by the failure of ATM and IntServ. Some important research directions might have failed in reflecting real customer needs. For instance, it is questionable whether VoIP customers would prefer to be denied service in case of overload by some admission control algorithms or rather get connected and experience a somewhat degraded speech quality. In any case, there still exist many promising topics in QoS research, among which the combination of QoS with resilience, the evaluation of user perceived QoS, and QoS provisioning in capacity restricted domains like wireless networks should be emphasized.

References

[1] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC1633, June 1994.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. RFC2475, December 1998.

[3] C. Dovrolis and P. Ramanathan. A Case for Relative Differentiated Services and the Proportional Differentiation Model. *IEEE Networks*, 13(5):26–34, September-October 1999.

[4] N. Christin and J. Liebrherr. A QoS Architecture for Quantitative Service Differentiation. *IEEE Communications Magazine*, pages 38–45, June 2003.

[5] A0 Deliverable No. 1: State of the Art Report. Technical report, Telecommunication Research Center Wien, ftw, May 2001.

[6] Y. Xu and R. Guerin. Individual QoS versus Aggregate QoS: A Loss Performance Study. In *Proceedings of IEEE INFOCOM*, volume 3, pages 1170–1179, 2002.

[7] P. Siripongwutikorn and S. Banerjee. Per-flow delay Performance in traffic Aggregates. In *Proceedings of GLOBECOM*, volume 3, pages 2634–2638, 2002.

[8] H. T. Tran. A Short Review on QoS Architectures and Applied End-to-End CAC Mechanisms. Technical Report, 2002.

[9] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

[10] T. Ott, T. Lakshman, and L. Wong. SRED: stabilized RED . In *Proceedings of INFOCOM*, volume 3, pages 1346 –1355, 1999.

[11] S. Floyd, R. Gummadi, and S. Shenkei. Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management. ACIRI Technical Report, 2001.

[12] W. Feng, D. Kandlur, D. Saha, and K. Shin. BLUE: A New Class of Active Queue Management Algorithms. Technical report CSE-TR-387-99, U. Michigan, April 1999.

[13] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin. REM: Active Queue Management. *IEEE Network*, 15(3):48–53, May/June 2001.

[14] B. Wydrowsky and M. Zukerman. GREEN: An Active Queue Management Algorithm for a Self Managed Internet. In *Proceedings of IEEE International Conference on Communications, ICC*, pages 2368–2372, 2002.

[15] H. Schulzrinne et al. RTP: A Transport Protocol for Real-Time Applications. RFC3550, July 2003.

[16] T. Friedman, R. Caceres, and A. Clark. RTP Control Protocol Extended Reports RTCP XR. Technical report, November 2003.

[17] Søren V. Andersen et al. Internet low bit rate codec. Internet draft, draft-ietf-avt-ilbc-codec-04.txt, 2003.

[18] C. Perkins, I. Kouvelas, O. Hodson, V. Hardman, M. Handley, J. C. Bolot, A. Vega-Garcia, and S. Fosse-Parisis. RTP Payload for Redundant Audio Data. RFC2198, September 1997.

[19] J. Rosenberg and H. Schulzrinne. An RTP Payload Format for Generic Forward Error Correction. RFC2733, December 1999.

[20] C. Perkins and O. Hodson. Options for Repair of Streaming Media. RFC2354, June 1998.

[21] H. Sanneck and A. Wolisz. Intra-flow loss recovery and control for VoIP. In *9th ACM Int. Conf. Multimedia*, Ottawa, Ontario, Canada, 2001.

[22] J. C. De Martin. Source-driven packet marking for speech transmission over differentiated-services networks. In *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, volume 2, pages 753–756, Salt Lake City, Utah, 2001.

[23] C. Hoene, B. Rathke, and A. Wolisz. On the Importance of a VoIP Packet. In *Proceedings of ISCA Tutorial and Research Workshop on Auditory Quality of Systems*, Mont-Cenis, Germany, April 2003.

[24] T. Bonald, S. Oueslati-Boulahia, and J.W. Roberts. IP traffic and QoS control: the need for a flow-aware architecture. In *World Telecommunications Congress*, 2002.

[25] T. Bonald, A. Proutiere, and J. W. Roberts. Statistical Performance Guarantees for Streaming Flows using Expedited Forwarding. In *Proceedings of IEEE INFOCOM*, volume 2, pages 1104–1112, 2001.

[26] L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thomas. LDP Specification. RFC3036, Januar 2001.

[27] E. Rosen and Y. Rekhter. BGP/MPLS VPNs. RFC 2547, March 1999.

[28] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. RSVP-TE: Extensions to RSVP for LSP Tunnels. RFC3209, December 2001.

[29] S. Bhattacharyya, C. Diot, J. Jetcheva, and N. Taft. Pop-Level and Access-Link-Level Traffic Dynamics in a Tier-1 POP. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, San Francisco, USA, November 2001.

[30] Cisco IOS Netflow Website. http://www.cisco.com/warp/public/732/Tech/netflow.

[31] MRTG Website. http://people.ee.ethz.ch/oetiker/webtool/mrtg.

[32] I. F. Akyildiz, T. Anjali, L. Chen, J. C. de Oliveira, and C. Scoglio. A new traffic engineering manager for DiffServ/MPLS networks: design and implementation on an IP QoS Testbed. *Computer Communications*, 26:388–403, 2003.

[33] K. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot. Long-Term Forecasting of Internet Backbone Traffic: Observations and Initial Models. In *Proceedings of IEEE INFOCOM*, volume 2, pages 1178–1188, 2003.

[34] H. T. Tran. Adaptive Bandwidth Provisioning with Explicit Respect to QoS Requirements. Technical Report, http://cntic03.hit.bme.hu/˜hung/prov-report.pdf, May 2003.

[35] J. Choe and N. B. Shroff. A Central Limit Theorem based Approach for Analyzing Queue Behavior in High Speed Networks. *IEEE/ACM Transactions on Networking*, 6(5):659–671, October 1998.

[36] C. Fraleigh, F. Tobagi, and C. Diot. Provisioning IP Backbone Networks to Support Latency Sensitive Traffic. In *Proceedings of INFOCOM*, volume 1, pages 375–385, 2003.

[37] The Internet traffic archive. http://ita.ee.lbl.gov/index.html.

[38] A. Autenrieth and A. Kirstdter. Fault Tolerance and Resilience Issues in IP-Based Networks. In *Proceedings of Second International Workshop on the Design of Reliable Communication Networks*, Munich, Germany, April 2000.

[39] C. Alaettinoglu, V. Jacobson, and H. Yu. Towards Milli-Second IGP Convergence. Internet Draft, IETF, November 2000.

[40] J. Moy. OSPF Version 2. RFC2328, IETF, 1998.

[41] Intra-Domain IS-IS Routing Protocol. ISO/IEC JTCI/SC6 WG2 N323, International Standards Organization, September 1989.

[42] E. W. Dijkstra. A Note on TWo Problems in Connexion with Graphs. *Numerische Mathematik*, 1:269–271, 1959.

[43] G. Iannaccone, C. Chuah, R. Moriter, S. Bhattacharyya, and C. Diot. Analysis of link failures in an ip backbone. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, Marseilles, France, November 2002.

[44] S. Bhattacharyya G. Iannaccone, C-N. Chuah and C. Diot. Feasibility of IP Restoration in a Tier-1 Backbone. *IEEE Network Magazine (Special Issue on Protection, Restoration and Disaster Recovery)*, March 2004.

[45] S. Iyer, S. Bhattacharyya, N. Taft, N. McKeown, and C. Diot. An approach to alleviate link overload as observed on an ip backbone. In *Proceedings of IEEE Infocom 2003*, San Francisco, USA, March 2003.

[46] F. Ricciato, M. Listanti, A. Belmonte, and D. Perla. Performance Evaluation of a Distributed Scheme for Protection against Single and Double Faults for MPLS. In *LNCS 2601, Proceedings of the 2nd International Workshop on Quality of Service in Multiservice IP Networks (QoS-IP'03)*, pages 218–232, Februar 2003.

[47] F. Ricciato, M. Listanti, and S. Salsano. An Architecture for Differentiated Protection against Single and Double Faults in GMPLS. *Accepted for publication on Photonic Network Communications journal, to appear in July 2004*.

[48] A. Gupta, B. N. Jain, and S. Tripathi. QoS Aware Path Protection Schemes for MPLS Networks. In *Proceedings of ICCC'02*.

[49] D. Xu, Y. Xiong, and C. Qiao. Novel algorithms for shared segment protection. *IEEE Journal on Selected Areas in Communications*, 21(8):1320 – 1331, October 2003.

[50] C. Diot. A Tier-1 IP Backbone Network, Architecture, Performance . Tutorial at ICNP'02, 2002.

[51] A. Charny and J.-Y. Le Boudec. Delay bounds in a network with aggregate scheduling. In *Proceedings of the 1st COST 263 International Workshop on Quality of Future Internet Services (QoFIS)*, pages 1–13, 2000.

[52] K. Papagiannaki, R. Cruz, and C. Diot. Network Performance Monitoring at Small Time Scales. In *Proceedings of Internet Measurement Conference (IMC)*, pages 295–300, October 2003.

[53] H. T. Tran and T. Ziegler. An Admission Control Scheme for Voice Traffic over IP Networks. In *LNCS 2720, Proceedings of 6th IEEE International Conference on High Speed Networks and Multimedia Communications HSNMC'03*, pages 353–364, July 2003.

[54] B. Fortz and M. Thorup. Optimizing OSPF/IS-IS Weights in a Changing World. *IEEE Journal on Selected Areas in Communications*, 20(4), May 2002.

[55] A. Nucci, B. Schroeder, S. Bhattacharyya, N. Taft, and C. Diot. IGP Link Weight Assignment for Transient Link Failures. In *in Proceedings of 18th International Teletraffic Congress (ITC18)*, August 2003.

[56] I. Gojmerac, T. Ziegler, and P. Reichl. Adaptive multipath routing based on local distribution of link load information . In *Proceedings of 4th COST 263 International Workshop on Quality of Future Internet Services (QoFIS)*, pages 122–131, Stocholm, Sweden, October 2003.

[57] I. Gojmerac, T. Ziegler, F. Ricciato, and P. Reichl. Adaptive multipath routing for dynamic traffic engineering . In *Proceedings of IEEE Globecom*, pages 3058–3062, San Fransico, USA, December 2003.

[58] A. Sridharan, R. Guerin, and C. Diot. Achieving Near-Optimal Traffic Engineering Solutions for Current OSPF/IS-IS Networks. In *In poceedings of INFOCOM 2003*, volume 2, pages 1167–1177, 2003.

[59] M. Menth, A. Reifert, and J. Milbrandt. Backup Capacity Minimization for Simple Protection Switching Mechanisms. TD(03)046, COST 279, 2003.

[60] http://www.cisco.com/warp/public/732/Tech/nmp/netflow/. January, 2004.

[61] R. Sommer and A. Feldmann. NetFlow: Information Loss or Win? In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, November 2002.

[62] VPN Architectures - Comparing Multiprotocol Label Switching, IPsec, and a Combined Approach. White Paper, Cisco Systems, 2004. http://www.cisco.com/warp/public/cc/so/neso/vpn/vpnsp/solmk_wp.pdf.

[63] Deploying IPsec Virtual Private Networks. White Paper, Cisco Systems, 2002. http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/depip_wp.pdf.

[64] http://www.checkpoint.com/. January, 2004.

[65] L. Ong, F. Reichmayer, A. Terzis, L. Zhang, and R. Yavatkar. A Two-Tier Resource Management Model for Differentiated Services Networks. Internet Draft, November 1998.

[66] P. Sampatakos, L. Dimopoulou, E. Nikolouzou, I. S. Venieris, T. Engel, and M. Winter. BGRP: Quiet Grafting Mechanisms for Providing a Scalable End-to-End QoS Solution. *Computer Communication*, 2004.

[67] Z. Duan, Z-L Zhang, and Y. T. Hou. Service Overlay Networks: SLAs, QoS and Bandwidth Provisioning. In *Proceedings of IEEE 10th International Conference on Network Protocols (ICNP)*, pages 334–343, 2002.

[68] L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz. OverQoS: Offering QoS using Overlays. In *Proceedings of the 1st Workshop on Hot Topics in Networks HotNets-I*, October 2002.

[69] R. Zhang et al. MPLS Inter-AS Traffic Engineering requirements. Internet Draft, Januar 2004.

[70] N. Feamster, J. Borkenhagen, and J. Rexford. Guidelines for Interdomain Traffic Engineering. *ACM SIGCOMM Computer Communications Review*, 33(5):19–30, October 2003.