Internet NG

Deliverable D2.7

Service Level Agreements

Colophon

Date:	April 18, 2001
Project reference:	Internet NG, WU2, D2.7
URL:	http://ing.ctit.utwente.nl/WU2/
Editor:	Ron Sprenkels
Authors:	Ron Sprenkels, Aiko Pras

Table of Contents

Service Level Agreements

1 Introduction

Internet is currently evolving from a best effort only service towards a service that supports different levels of Quality of Service (QoS). The service provider makes a (legally binding) commitment to deliver those specified levels of QoS. The next step is to enable customers to influence the behavior and configuration of their own instance of the service. This is called Customer Service Management. A key concept to enable customer service management is the concept of a Service Level Agreement. In this deliverable Service Level Agreements (SLAs) are defined and examined in detail, in particular for IP based networks like the Internet.

The deliverable is structured as follows. First the concept of a SLA and the rationale for having SLAs is discussed (section 2). Then the SLA life cycle is discussed (section 3).

Up to this point the discussion of SLAs was service independent; the remainder of the deliverable focuses on the SLA of a specific service. For this purpose an IP transport service is used. The SLA of this service is introduced (section 4). Then the performance parameters (section 5) and the accounting aspects (section 6) are covered. Finally the verification of this type of SLA (section 7) and some specific tools for that purpose (section 8) are discussed and some concluding remarks are presented (section 9).

2 The Service Level Agreement Concept

A general definition of a Service Level Agreement is given in [Verma99] to be:

> *"an explicit statement of the expectations and obligations that exist in a business relationship between two organizations: the service provider and the customer"*

Bilateral SLAs can also be defined among pairs of organizations that have a symbiotic relationship. In such case each organization has both roles at the same time: it is the provider of its own service and the customer of the service of the other organization.

The SLA constitutes the legal foundation for the delivery of the service. SLAs are used by both parties involved; the service provider uses it to have a definite, binding record of what is to be provided. The provider can use this record in case of disputes with the service customer. This also works the other way around: the customer also uses the SLA as a legally binding description of what the provider has to deliver.

A SLA typically has the following components [Verma99]:

- A description of the service that is to be provided.
- The expected performance of the service.
- A detailed procedure for handling problems with the service.
- A procedure for monitoring and reporting the service level to the customer.
- The consequences of the service provider not meeting the agreed service level.

- A description of under which circumstances the SLA does not apply.

The service customer in turn uses the SLA to verify if he is actually getting the agreed upon service levels. This is possible since the SLA also contains feedback parameters to the customer about the actually achieved service levels.

In traditional SLAs the customer can perform only a limited number of actions. In most cases the service that is to be delivered is static.

The customer can complain if he feels that the service level is not in accordance with what was agreed in the SLA. A reporting procedure or trouble ticket system is described in the SLA. Another important customer action is paying for the received service.

With the onset of more configureable services a larger set of service parameters comes into play that needs to be managed. Values for these service parameters must be determined based on customer preferences, and these parameters need to be actually given these values. In a Customer Service Management approach the customer gets direct control over this process. A category of SLA parameters arises that the customer can change autonomously, within the pre-determined boundaries specified in the SLA.

3 SLA Life Cycle

We identify three phases in the existence of a SLA: the creation phase, the operational phase and the removal phase. The division into three phases will help in structuring the various SLA management activities later on.

3.1 Creation Phase

A SLA is first created when a customer subscribes to a service that is offered by a service provider organization. A (possibly complex) chain of events leads to the point where the customer wants to subscribe to the service. The customer first has found out about the existence of the service offering, and has gathered enough detailed information about this offering to judge if it is a service that he wants. The customer might have been actively searching for a service offering to fit a given customer need that exists. Another possibility is that the customer was not actually searching for a service offering, but got to know about it through unsolicited advertisements, word of mouth, or via some other means.

SLA creation involves a number of activities:

- Officially (this is, legally binding) establishment of the agreement. This reflects that the customer has actually subscribed to the service, is aware of the detailed, legally binding extent of what is comprised in the service delivery, has copies of all relevant information about the service, etc. In this step the customer signs a service delivery contract.
- All required Service subsystems need to be configured to accommodate this new service subscription. So this includes access authorization systems for the service, entries into billing systems, entries into the service logic of the service, reservations of required, per-customer service resources, etc.

SLA creations are probably also input to longer term resource planning activities for the service provider as a whole.

3.2 Operational Phase

In this phase, the service customer has online access to the static, read-only terms of the SLA, and also to the dynamic read only parameters of the service. This includes the performance related data that the service customer is interested in.

The customer can also make changes to his SLS parameters. This possibility is also completely described in the SLA). In order to make SLA changes, the customer needs some form of access to a Customer Service Management interface to the service. This is the topic of chapter X. Any changes the customer makes need to be mapped onto the actual service configuration. This is covered in chapters X and X.

Note that this type of service configuration changes in most cases will have an effect on the bill for this service.

3.3 Removal Phase

When a customer subscription to a service is ended the SLA and all associated configuration information in the service systems needs to be removed. The same service sub-systems that were addressed for SLA creation are again involved for this step.

The event that triggers SLA removal is the fact that the customer service subscription is not renewed, or actively ended.

The service subsystems contain configuration information about this subscription, that information needs to be removed. Resources that were claimed to support this subscription need to be freed.

Note that the configuration information for a given subscription can be aggregated with configuration information for other subscriptions. It is therefore possible that information needs to be *changed*, not just only deleted.

4 Case of a SLA: The IP service

To better understand what SLAs are in more detail, this section examines a concrete instance of a service, and its associated SLA. For this purpose a well known service is used: a Internet connectivity service.

To be able to describe various SLA aspects of the IP service a suitable service model is needed.

4.1 IP Service Model

The basic functionality of the service is the service customer’s ability to send IP packets to other customers, and to receive IP packets from other customers. It is a bare IP Internet connectivity service. The service is delivered *to* the customers *by* the Internet Service Providers.

The basic property of the IP service model is that a Internet Service Provider (ISP) delivers the IP service to a set of customers. The most basic configuration of a ISP and two customers is shown below:

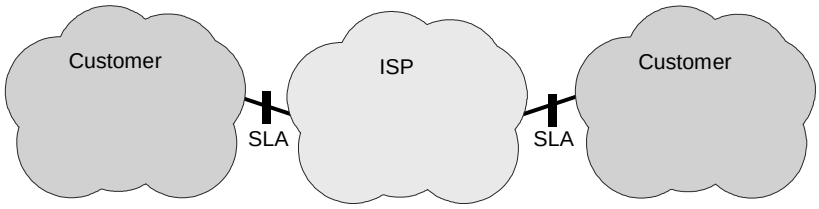

Figure 1. IP Service Model

Each customer has its own Service Level Agreement with the ISP about the delivery of the IP service.

The performance of the IP service is important for the customer, so the SLA contains a specification of what the minimum acceptable performance is. For this purpose, the SLA contains some parameters that are in effect boundary values for some particular IP performance measures. To be able to define these performance measures, a model of the IP service is needed that allows these measures to be defined.

The most obvious place to look for such a model is the Internet Engineering Task Force [IETF], since this is the place where standardization activities regarding the Internet Protocol take place. The core protocol definitions were developed there [IPv4, IPv6], as are other main IP standards. Also for performance related issues there is work being done in the IETF, amongst others in the IP Performance Metrics working group [IPPM-wg].

However, the IETF has not defined a network model for IP on which performance measures are based. The ITU-T does have such a model, as specified in recommendation I.380 [I.380]. The model for an IP service and its associated performance parameters is constructed from the ground up there, and this model will be used as a basis for the work in this thesis. The terms used for the various components of the IP service are the same as are used there.

An overview of the IP service model defined by ITU-T in recommendation I.380 is shown in Figure 2 below.

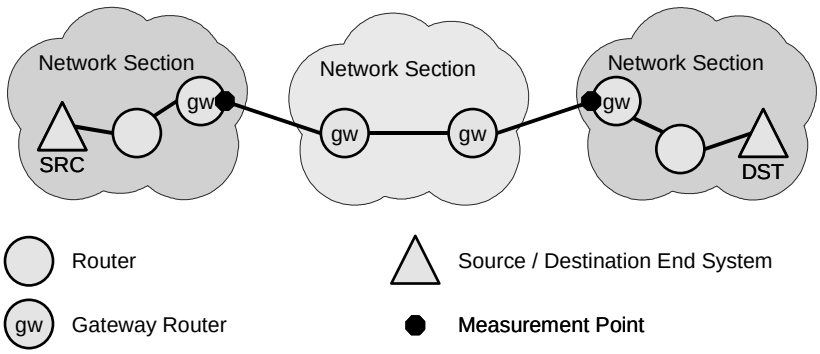

Figure 2. ITU-T IP Service Model

The figure shows an IP infrastructure, and how it is built up from two basic component types: hosts and links.

A *host* is a system that communicates using the IP protocol. If a hosts forwards incoming packets at the IP layer, and as such enables the communication between other hosts, it is called a *router*. A host where IP packets originate is called a *source host*, (marked SRC in the figure) and a host where IP packets finally arrive is called a *destination host* (marked DST in the figure).

A *link* is a point-to-point connection for transporting IP packets between a pair of hosts. It is below the IP layer, so it does not contain any intermediate hosts.

To allow for additional structuring of the IP infrastructure, network sections and circuit sections are defined. A *network section* is a set of hosts together with their interconnecting links that all fall under a single responsibility. A *circuit section* is a link that either connects a source or destination host to an adjacent host, or it connects a router in one network section to a router in another network section.

Finally, the term *gateway router* is used to denote a router at the border of a network section that sends and receives packets across a circuit section to a gateway router in another network section.

To facilitate the definition of performance metrics for the IP service, measurement points are defined. A measurement point is located at the boundary between a host and an adjacent link. Figure 2 shows two measurement points that allow performance metrics about the middle network section to be defined.

Peering and Backbone Service Providers

In general, a complete configuration of a IP service will have multiple customers, and also multiple cooperating providers. In Figure 3 a IP service configuration is shown that has multiple customers, multiple ISPs and a backbone service provider.

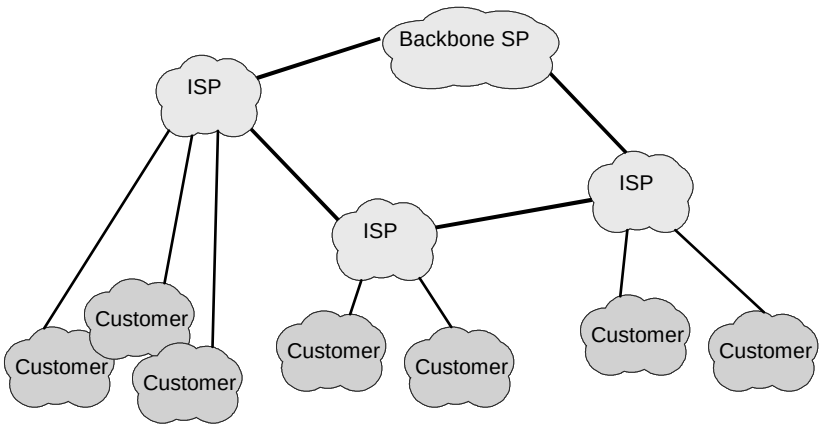

Figure 3. General configuration example of IP service

In the figure, SLAs exists at each point where a provider is connected to a customer or to another provider.

For the IP service, not all customers are in the same category: one can think of different types of customers. Examples include private persons that want to connect to the Internet, and businesses or other institutions that want to have a Internet connection.

ISPs will cooperate with other ISPs and with backbone connectivity providers to enlarge the coverage of their service for their customers. This cooperation takes the form of interconnecting the network infrastructure of a ISP with the network infrastructure of other ISPs, and with backbone providers. In case of differences in the exact IP service offering between the two interconnecting parties, the ser-

vice offerings of both sides have to mapped onto each other. For such a interconnection also a SLA is created.

If two ISPs that are directly connected to each other are equal, such an agreement is also called a *peering agreement.* Equal in this case means that they are comparable in size, and that their level of interconnections to other ISPs is comparable. As a result, the flow of traffic across their interconnection will also be comparable. Both ISPs will benefit equally from the peering agreement, and for this reason there is usually no financial agreement needed.

If one of the ISPs has a far bigger advantage from a ISP interconnection than the other, there is no longer a peering situation between two equal parties. This is for example the case when a small ISP connects to a big, well interconnected ISP. In such a case, the interconnection gets the characteristics of a customer to provider connection, and the customer (the smaller ISP) will have to pay for the connectivity to its provider (the bigger ISP).

A *Backbone Service Provider* is a special type of service provider. It does not have direct customers of the IP service in the way ISPs do. Instead, it has only connections with other ISPs (see Figure 3).

Types of Customers

Earlier two types of customers were identified: private persons and larger organizations (businesses, institutions). For each type, the SLA properties will be discussed separately.

Larger organization. For a corporation a high service availability is usually of prime importance. The financial losses that occur when the IP service not available can be enormous. This is in particular the case when a corporation uses the IP service for its primary business, like online bookstores and online travel agencies. Therefore, the SLA can contain a detailed, fast procedure to recover from service outages. Also the SLA can contain

Private customer. For a private person the service availability ratio is important, but not that important that the customer is willing to pay a lot for a very highly available service. As a result, a detailed procedure to report and fix service outages will in general not be a part of this type of SLA.

Categories of SLA information

An SLA between ISPs is also called a peering agreement. It specifies the terms under which both ISPs exchange local and transit traffic with each other, as described earlier. Next some examples of such terms are discussed.

Traffic levels: The SLA can describe how much traffic can be exchanged. For each of the two ‘directions’ of the SLA this can be done, and also for a given direction the total amount of traffic can be grouped in categories. Criteria for traffic to be in a given category can for example be the IP destination of the traffic, or if the service supports different QoS classes, the QoS class associated with the traffic.

Transit traffic: when one ISP sends traffic into the other ISP, and this traffic has a destination that is outside that other ISP, this is called *transit traffic*. As such, transit traffic is of interest for ISP to ISP SLAs. The SLA specifies if transit traffic is allowed, and if it is, what the conditions for that traffic are. Conditions include the amount of such traffic, the quality levels of the traffic etc. These conditions can also be different for different groups of destination addresses.

In most cases IP traffic between communicating customers is exchanged in both directions. This is for example true for all traffic that is carried over TCP connections, since control information has to be exchanged. A peering agreement for transit traffic is only useful if traffic can in one way or another

flow in both directions. This is an example of the fact that there will be dependencies between the various SLAs that a given ISP (or backbone provider) has with other ISPs.

Roaming access: ISPs might mutually agree to accept each others dial-in customers into their modem-pools. For this to work, provisions must be made that allow roaming users to be authenticated in order to gain local access.

5 IP SLA performance Parameters

The SLA contains boundary values for specific IP performance measures. This section identifies a set of relevant IP Performance measures that will be used in IP SLAs. For each of these measures a short discussion is included that motivates why each parameter is useful and meaningful for an IP SLA. Examples of applications of the service that show the significance of the SLA parameter will be given.

A systematic approach to identifying the required IP QoS parameters for use in the SLA is needed. ITU-T recommendation I.380 [I.380] proposes such a systematic approach, that will be taken here as well. The approach concentrates on the 'User Information Transfer' function from the 3 x 3 matrix approach from [I.350], since of the total of the three basic functions (Access, User Information Transfer, and Disengagement) this is the only relevant one for IP networks.

In section 4.1 the basic model for an IP infrastructure was presented. When a IP packet is transmitted from a source host to a destination host, four basic IP packet transfer outcomes are possible, these are the Successful, Errorred, Spurious and Lost outcomes.

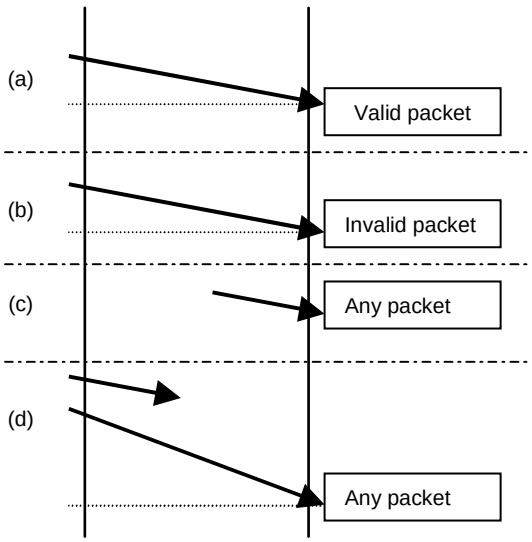

Figure 4. Successful (a), Errorred (b), Spurious (c) and Lost (d) IP Packet outcomes

Section 5.5 of [I.380] gives the detailed definitions of the four basic IP packet transfer outcomes. A successful packet outcome has occurred when a IP packet is transmitted, and is received in time (so without too much delay), with a valid header and error-free payload. A Errorred IP packet outcome has occurred when the packet is received in time, but either the header or the payload have errors in them. A spurious IP packet outcome has occurred when a packet with a valid header arrives, but it was not sent into the network anywhere. If a packet is dropped somewhere in the network, or it is delivered intact but too late, this is a lost IP packet outcome.

When applied to the specific case of IP networks, the 3 x 3 matrix approach results in the following IP performance parameters:

- IP Packet Transfer Delay (IPTD) and its mean and variation,
- IP Packet Error Ratio (IPER),
- IP Packet Loss Ratio (IPLR).
- Spurious IP Packet Rate
- Throughput (IP packet throughput IPPT and Octet based IP packet throughput IPOT)
- IP Service Availability

Most of these parameters are defined over a set of packets of interest called the ‘population of interest’. This usually denotes a total set of packets transmitted from a source to a destination. The IP performance parameters will be discussed in separate sections next.

5.1 Packet Transfer Delay

The IP Packet Transfer Delay (IPTD) of the IP service is defined as time difference between the time the first bit of a packet passing the ingress measurement point, and the time the last bit passes the egress measurement point at the other end. It is determined by the IP packet size, queueing delays inside the networks, the raw medium transmission speeds at the sending and receiving end, and of course the (geographical) distance. Note that although the shortest path between any two points is a straight line, network cables tend to be put in different places from that line, and are hence longer. Also the route that packets take through networks are determined by factors like routing tables, the topology of the various subnetworks, peering agreements between various networks and so on.

Because routing in IP networks is dynamic down to the individual packet level latency can vary considerably from one packet to the next. If fragmentation of packets occurs, the individual packet fragments can travel along different routes before getting reassembled, and cause considerable differences in delay. Also, when sending a larger number of packets, the nature of the routing and forwarding functions in IP networks is such that these packets can travel different routes, and thus experience different packet delays.

Packet delay is of interest for anything interactive, like video conferencing, audio/telephony over IP, real time transaction systems (banking), web-browsing, etc. Applications can be sensitive to delay even when one might not expect this. Transport layer protocols like TCP for example have difficulty achieving high throughput values if the transfer delay is too high.

The sensitivity of an application to packet delay can be asymmetrical. For example, an application that uses the TCP transport protocol to receive bulk data is sensitive to a high delay in the direction of the actual data flow, but less sensitive in the direction of the flow of TCP acknowledgements. For this reason it is useful to not only specify in the SLA values for the round trip delay, but also for the one-way delay.

For a Voice over IP connection to be usable, the one-way delay should be below 150 ms [G.114]. [ITU-R M.1079] probably has some info on acceptable delay and delay variation for voice traffic, but I haven’t been able to get hold of this document yet.

5.2 Variation in Packet Transfer Delay

The variation in packet transfer delay (or IPTD, also know as ‘jitter’) is defined to be the variation in latency experienced by individual packets in a sequence of transmitted packets.

Packet delay variation is of interest for streaming applications that want to use small play-out buffers, and/or for applications that are intolerant to large values for the packet delay variation, but want to take limited or no measures to compensate for this effect.

5.3 IP Packet Loss Ratio

The IP Packet Loss Ratio (IPLR) the proportion of all sent packets of a population of interest that is received too late, or not received at all at the destination.

Various applications have different sensitivities to errors in packet transmissions. The following two examples show this effect.

An example of error ratio insensitive traffic: non-compressed audio is not that sensitive. In the end, the audio information is intended for the human ear, and it happens that the human hearing is quite forgiving if the quality of the audio gets worse. If the audio information is sent in sufficiently small packets, the degradation of the audio signal will be noticeable, but it will still be usable. If the audio is encoded a bit smarter for transportation, it can even get better still, up to a point where an occasional packet loss is hardly if at all noticeable. This can be achieved by spreading around consecutive audio samples across multiple transport packets. Then, if a packet is lost, this means that a number of audio samples are missing, but due to this smart transport encoding, there will not be longer periods of time that have no samples. Instead, a single packet loss will result in a small number of missing samples at multiple different places in the total set of samples. Due to the nature of the human ear, and perhaps with some interpolation of neighboring samples, the overall effect will be minimal.

Example of error ratio sensitive traffic: encoded/compressed audio or video can be very error sensitive. If a video stream is encoded as the initial image, and after that only as the delta image to get to each next image, then a single loss of such a delta will affect the quality of the video stream, for a longer period of time, right up to the point where a new complete image is sent.

Error ratios of an IP service are particularly important in cases where you cannot spare additional round trips to correct errors.

Banking applications are extremely transmission error sensitive. Therefore, this type of application will use application layer error detection and correction, at the expense of additional round trip times anyway.

Gaming-style applications (which are a form of a distributed processing system) are another example of a error rate sensitive application. Other examples of latency and error intolerant applications are distributed computing systems where the separate computational tasks are closely bound to results from other tasks. The results of each small simulation step are needed for the next step elsewhere. These applications are characterized by the fact that the latency of the transmission should be low, so low that an additional round trip to get a retransmission is unacceptable. The retransmitted packet would arrive so late that, even though it now arrived error free, because it is too late the retransmission has become useless. The application needed the information contained in the packet earlier, and when it arrives late, the information can no longer be used. If the computation cannot continue without the information, the overall computational process is considerably slowed down if latency is too high.

5.4 Spurious Packet Rate

Spurious packets are packets that are received at a destination, that have a valid IP packet header, but were not transmitted at the sender that is indicated in the packet header. It is a rate instead of a

ratio, because it is not expected that the occurrence of spurious IP packet outcomes has little to do with the particular piece of the infrastructure that is being tested.

5.5 Throughput

The throughput of an IP service is a measure that applies to a particular pair of source and destination IP address. It is defined in two variants: the IP Packet Throughput (IPPT) and the Octet based IP Packet Throughput (IPOT). The IPPT indicates the number of packets per second the service can handle, and the IPOT indicates the accumulated number of octets in those packets that the service can handle.

About the SLA parameter throughput the following observations apply:

- Throughput is important. Example: a video stream is defined at 1 MBit, and only useful at 1 MBit. At 0.9 MBit, it cannot be displayed. Another: company wants to synchronize its various data bases outside business hours, not during peak sales hours.
- In case of different service classes, throughput applies for each individual service class, and can be different per class.
- The parameters also depend on source address and/or destination address (differences between local traffic, long-distance traffic etc.). In this aspect the model of the network infrastructure as a set of ‘pipes’ with QoS attributes becomes evident.
- Service Throughput can depend on time of day/day of week etc. Either the value for this parameter applies always (also in peak hour), or there might be different values for different times (inside or outside business hours).

5.6 Availability

The IP service is defined to be available at a given point in time if the IP packet loss ratio (section 5.3) is below a defined treshold t. The availability function periodically verifies if the service is available between a pair of measurement points or not (this is a boolean function) and in this way divides time into periods of service availability and service un-availability. From this division, the availability ratio can easily be computed. The proposed measurement interval is 5 minutes.

The ITU-T uses an end-to-end notion of availability; availability is defined between two endpoints in the network, and an active measurement method for checking the availability is used.

Sidestepping for a moment to modern telephone networks, we see availability numbers in the range of 99.999 percent which amounts to an acceptable service down time of a few minutes per year! Depending on how they are designed, provisioned, managed and used, IP networks currently cannot yet reach those high availability figures.

Businesses will in most cases have stricter availability requirements when compared to private persons requirements. The reason for this is that a business loses more money per unit time than a private person, so a business can and wants to afford a more expensive, higher availability service.

Availability guarantees might differ depending on time of day; off-peak and on-peak hours, inside or outside business hours. Again, for business type customers this can be useful.

6 Accounting aspects

In most cases the customer of a service has to pay for using the IP service. In case of a customer subscribing to a public commercial ISP, the payment is directly from customer to ISP. In other cases of SLAs the payment can be more implicit, for example a corporation that internally differentiates the

total external internet connectivity costs over its individual departments. Also in that case some form of accounting has to be done. Therefore, an important component of a SLA is the description of how the accounting for the service is arranged.

Before we go into the service oriented aspects of accounting we will first take a look at different approaches to accounting from the customer perspective. Extensive research to accounting and charging of services has been done. In an overview research work of Andrew Odlyzko [Odlyzko00] Internet pricing is examined in the light of the longer history of communications in general. This leads to some interesting points of view on how customers of services experience accounting. A few of these points that are particularly relevant for the accounting of the IP service are the following.

A flat rate charging scheme tends to increases overall usage of the service, whereas a usage based charging scheme acts as an incentive to make little use of the service.

Users have a strong preference for uncomplicated, risk free accounting schemes. Even in cases where a usage based charging scheme would result in less total costs, still there is a strong preference for simple, flat-rate schemes. One explanation for this effect is believed to be the amount of mental resources it takes from the human user to deal with lots of separate decisions wether or not to use the service in case of a usage based charging scheme. Another explanation is the natural risk avoiding behavior of people. If the total charge for the service is a predetermined fixed amount, the risk of having to pay a large amount is gone. Users are willing to pay for the fact that they are not running such a risk.

Accounting schemes can vary in many different ways. A way to structure accounting schemes has been proposed in [Hartanto99], where the total charges have been structured into a subscription charge and a session charge, that each consist of a setup component and a recurring or usage component.

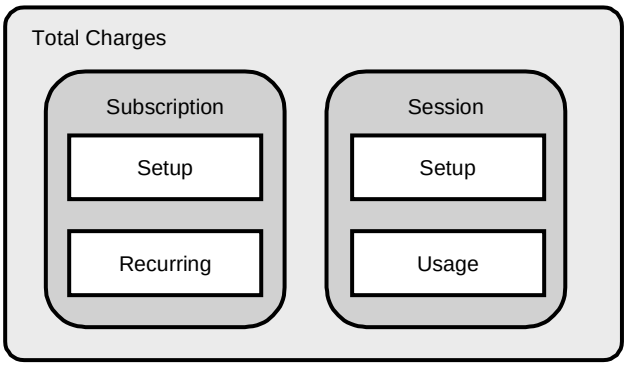

Figure 5. Structure of charges for services

For the case of the IP transport service, we can use this structure to recognize three categories of accounting schemes: flat rate accounting, reservation based accounting and actual usage based accounting.

6.1 Flat rate accounting

Flat rate accounting is the simplest of the accounting schemes. Both the setup and usage components of the per session charges are zero. According to the research mentioned earlier, this type of accounting scheme is most appealing to the majority of service users.

One benefit of adopting this charging scheme is that it is not needed to gather any data on sessions of service customers for charging purposes. Since gathering this kind of data with the required accuracy that is needed for accounting is an expensive activity in itself, this is an advantage. Note that a trade-off has to be made between this fact and the bigger incentive for customers to make use of the service under a flat rate accounting scheme.

6.2 Reservation Based accounting

This accounting scheme is targeted at an IP service that supports different Quality of Service levels. It is a fundamental property of having different QoS levels that there must be some incentive for customers to not just always use the highest possible quality level. The application of a suitable accounting scheme can be such an incentive.

In the QoS aware IP service, the customer can select different QoS levels. In view of the charging structure of Figure 5, the reservation based charging scheme regards the fact that a customer selects a higher QoS level as the starting point of a session, and as soon as the customer switches back to the default (lowest) quality level the session has ended. For these sessions a separate charge applies, with a usage component (for example time duration) and possibly a setup component.

This charging scheme is more complex than the flat rate scheme, in the sense that data has to be gathered about customers changing the QoS level of their service. These changes however are expected to occur at relatively large time scales.

6.3 Actual usage based accounting

The actual usage based accounting scheme measures actual use of the IP service, at a very fine grained level. Of the per session charges both the setup and usage components are used. For the usage components the following types of information can be gathered:

- The number of sent and received bytes

These byte counts can in turn be counted separately based on the QoS class that was used, or based on the origin or destination category of the packet.

- Duration based charging can be used

This type of charging occurs when a per unit time charge applies for using the service. This model was in widespread use at ISPs that supply dial-in type connections to the Internet.

7 Verification of TCP/IP SLAs

SLAs contain different categories of provisions and parameters that all potentially need to be verified. Both the provider of the IP service and the customer will want to verify if the SLA is being met.

The customer is paying to receive the service levels specified in the SLA, and wants to verify if he is actually getting those service levels. Likewise, the provider has committed to delivering the service levels specified in the SLA, and wants to monitor if this is actually the case.

The provider can also use the verification results as a record and as a form of proof of the actually achieved service levels. This information can be useful in case of any disputes with customers.

One more important section of SLA parameters that is subject to a need to be verified is the section on the charging and billing for the service. Under any other charging scheme than a strict flat rate charging scheme, the total amount of the bill will depend on some form of actual usage of the service.

The service provider gets this usage data from the service infrastructure, and uses that data to make a bill. The customer wants to be able to verify the accuracy of the bill.

Not all SLA provisions are suited for automated verification. The SLA can for example indicate a maximum time that customers are allowed to be put on hold when reporting a problem with the service by telephone. A customer can verify this parameter, but has to do this by keeping track of time while being on hold on the phone.

This section focusses on the verification of the IP performance parameters that were discussed in section 5. Each of the parameters that were discussed there will be examined too see how that parameter can be verified.

Selection of measurement points

The values for the IP performance parameters are defined *at* the interface between provider and customer. It is to be expected that the service provider can only take actual measurements at points that are within the responsibility domain of the provider, and likewise, that the customer can only take measurements at points that are within the responsibility domain of the customer. Care should therefore be taken that the measurement points are placed as close as possible to the ‘SLA border’, to reflect as accurately as possible the real SLA performance parameters.

7.1 Verification of Delay and Delay Variation

The delay and variation in delay are measures that are defined between two measurement points in the IP infrastructure. Values for both of these measures can be obtained from the same set of delay measurements of individual IP packets being sent between the two measurement points. Packet delay can be defined and therefore measured in two ways: one way delay and round trip delay.

The IP Performance Metrics working group of the IETF has defined a framework for the specification of IP Performance metrics [RFC2330]. The working group has defined two metrics dealing with the delay in IP networks: a one-way delay metric [RFC2679] and a round-trip delay metric [RFC2681]. These documents also contain a description of the required procedure to obtain a value for the metrics.

To obtain a value for the one-way delay between a pair of IP addresses, the procedure is the following.

- Measuring the delay of a single packet involves the recording of the current time at both the source and the destination of the packet. For this to result in accurate values for the delay of the packet, the two systems should both have a accurate notion of the current time.
- A stream of measurement packets is sent from the source to the destination. The distribution in time of the packets can be a Poisson distribution. Upon sending a packet the sender stores the current time in the packet. The receiver records the time of arrival of each packet. By substracting these times an estimate of the one-way delay of the packet is obtained.
- From the resulting series of delay values for individual packets a mean value for the packet delay as well as the variation in packet delay can be computed.

7.2 Verification of Error Ratio

The error ratio in an IP network is defined between two IP addresses, and over a set of IP packets. In contrast to some of the other performance measures that are discussed here, a procedure to measure the error ratio has not been defined in the IETF IP performance working group so far.

To obtain a value for the error ratio between a pair of IP addresses, the procedure can be the following.

- The receiver is prepared to receive IP packets.
- A stream of measurement packets is sent from the source IP address to the destination IP address. These packets have a known content, different for each packet. This can be achieved by means of a sequence number for example.
- At the receiving end all received packets are examined. Of each packet is determined if it belongs to the stream of measurement packets, and if so, if it was received without errors.

Errors can occur in the payload of the packet, but also in the packet header. Errors in the header need to be treated in a special way at the receiver, since errors can affect the apparent source or destination address, port numbers and so on of the packet. As a result, the receiving measurement system has to determine for every packet that arrives at the system if it belongs to the measurement stream, not only of the packets that have the expected header values. Due to errors in the header the packet can appear to come from any IP address and port number, be destined for any IP address and port number, but still belong to the measurement stream. Implementing this type of measurements might require implementations of the IP protocol that are especially tailored for this type of measurement.

7.3 Verification of Loss Ratio

The error ratio in an IP network is defined between two IP addresses, and over a set of IP packets. A procedure to measure the loss ratio is defined by the IETF IP Performance Metrics working group in [RFC2680].

The measurement procedure involves sending a stream of IP measurement packets from the source IP address to the destination IP address, and determining of each packet if it is lost or not. The steps involved are the following.

- To determine if a packet is lost involves measuring the time it has been in transit. If this time is over a treshold upon arrival, the packet is still considered to be lost. Therefore, the sender and receiver systems need to have synchronized clocks.
- A poisson distributed stream of packets is sent from source to destination. Of each packets it is determined if it was lost or not. This involves inserting sequence numbers into each of the packets.
- After the packet stream has terminated, the ratio between the lost packets and the complete set of packets is computed.

7.4 Verification of Spurious Packet Rate

In contrast with the other metrics defined discussed in this section on verification, the spurious packet rate is defined at a single measurement point in the IP network. There is no measurement procedure defined for this metric in the IETF IP Performance Metrics group.

Measuring the spurious packet rate at a measurement point is quite a complicated task, because for every packet that is received at the measurement point, it has to be determined if the packet was actually sent into the network. From a functional point of view, the procedure to measure the spurious packet rate at a measurement point is the following.

- For a period of time (the measurement period) collect the header information about all arriving packets at the measurement point.

- During that same period, also gather the header information of all packets entering the network at all possible network ingress points.
- For each of the packets that arrived at the measurement point, determine if the packet was actually sent into the network at one of the ingress points. If such a corresponding ingress event cannot be found, the packet was a spurious packet.

From the last step a total number of spurious packets in the measurement period is found, and when divided by the measurement period this results in the spurious packet rate.

The procedure to measure the spurious packet rate for a given measurement point on a IP network is very extensive. It involves recording the IP header information of *all* packets that flow into the network. As a result, the measurement procedure quickly becomes very expensive, and for that reason impractical. It is to be expected that in most environments there is no justification to spend the effort for doing this type of measurements.

7.5 Verification of Throughput

When measuring the achievable throughput between a pair of IP addresses it is important to make sure that the performance of the IP network is being measured, and that not some other components of the measurement system are the real bottlenecks for performance.

The IETF IPPM group is currently in the process of defining a metric called the Bulk Transfer Capacity or BTC [Mathis00]. This metric reflects the throughput that a single transport layer connection can achieve between the two measurement points. The metric therefore closely matches the most common ways in which the service will be used. But there is a potential problem with this definition as well. The Bulk Transfer Capacity between a pair of IP addresses can be less than the actual capacity that could be achieved using multiple transport connections. This is true in particular for high capacity internet connectivity, where the access capacity is orders of magnitude higher than the access speeds of individual end-systems.

Another issue that needs to be taken into account is that this type of active measurements run the risk of using up a lot of resources of the service by generating a large amount of measurement traffic. The risk is that the measurement traffic severely degrades the service levels for end-users. This has to be avoided.

7.6 Verification of Availability

The availability between a pair of IP addresses can be measured in line with the ITU-T based definition of availability that was given in section 5.6. The availability is a derived measure from the error ratio, and can be found by periodically comparing the error ratio to a treshold value. If the error ratio is above the treshold the service is considered to be unavailable for the duration of the sample interval, otherwise it is considered to be available. This procedure divides time into period of availability and periods of unavailability of the service, and from that the availability ratio over a given period can be computed.

The IETF IPPM working group also has a definition of connectivity that can be used as a availability measure, this is the definition given in [RFC2678]. In this document first a notion of instantaneous connectivity between two points is defined. In the IPPM terminology set this is a singleton measure. From the instantaneous connectivity a connectivity measure over a time interval is defined.

The IETF approach differentiates between three forms of connectivity, and hence three forms of availability. These are the uni-directional availability, the bidirectional availability and the two way

temporal availability. The latter two measures are both bidirectional. The rationale for still having two different kinds of bidirectional measures is that the first one (although it is bidirectional) does not denote a general useful notion of connectivity. Under that definition of bidirectional connectivity, it can happen that one side can reach the other and vice versa during some interval, while at the same time a response packet to a request packet can not reach the requester anymore. This can be the case due to the temporal ordering of events. The second kind of bidirectional connectivity does have this temporal ordening constraint, and for this reason it is considered a generally useful notion of connectivity.

8 SLA Verification Tools and Techniques

In the previous sections we have seen definitions of SLA parameters and general procedures for the verification of TCP/IP SLAs. There are many such tools available. Also there are more general implementations of techniques that can be used. This section examines some of the available measurement tools and techniques that can be used to actually carry out the measurements that are needed for the verification.

8.1 Netramet

Netramet is an implementation of the Meter MIB [RFC2720] defined by the IETF Real-time Traffic Flow Measurement working group. This working group has completed this work, and is therefore no longer an active working group. A record of the output of this group is still available elsewhere [RTFM].

The basic architecture of a RTFM traffic flow measurement system consists of three basic building blocks:

- *Meters* gather data about packets and condense this data into ‘flow data’.
- *Meter Readers* retrieve flow data from meters using the SNMP protocol.
- *Managers* co-ordinate the activities of the meters and the meter readers.

The basic operation of the system involves the creation of rulesets. Each rule in a ruleset defines the criteria that determine which packets belongs to which flow. Any information that is available in the IP packet header can be used for this, so rules can look at source IP address, destination IP address, source and destination port numbers, protocol identifiers (TCP, UDP, ICMP) and so on. A flow is defined in the NeTraMet manual as

> *“a stream of packets exchanged between two network hosts, which we refer to as the flow's source and destination. Flows are bi-directional in that packets and bytes can be counted in the 'to' (source to destination) and 'from' (destination to source) directions”*

The rulesets can be descibed in the native format that NeTraMet uses, or they can be written in a higher level language called the Simple Rule Language [SRL] and then compiled into the native format.

The manager downloads the ruleset into the meter, and the process of matching packets against the ruleset and counting the packets in each individual flow begins. The network card is put into promiscuous mode, so that every packet in the complete broadcast domain is examined by the NeTraMet software. Each arriving packet is matched against the rules in a rule set, and if it matches, it belongs to a flow. If a flow did not exist, a ‘record’ for it is created. As long as new packets arrive within the flow time-out value, the flow stays in existence, otherwise, it is closed.

The meter readers gather the data from the meters at regular intervals for further processing. A separate program called nifty is available that provides a near-Real-time graphical view on the measured flows in a ruleset.

It is in the processing step in the meter reader that the actual SLA parameter verification has to take place.

8.2 Ntop

Ntop [Ntop] is an integrated tool that can do various kinds of measurements and types of analysis on IP traffic. The four main functions of Ntop are traffic measurement, traffic monitoring, network optimization and planning, and detection of network security violations. Ntop is available as an open source package [OpenSource].

The basic operation of Ntop is the following. The tool in installed on a machine in a subnetwork, and starts capturing and analysing all packets that can be observed by that machine. This is done by putting the network card of the machine into promiscuous mode. This technique can therefore only be applied within a broadcast domain, for instance a shared ethernet segment. The information about the captured packets is then used to perform the four main Ntop functions.

Measurement: traffic from and to each host in the subnetwork is measured and categorized, based on IP layer information but also higher layers like TCP and UDP, and based on well-known protocols on top of that like HTTP, NFS, X11, SMPT etc.

Monitoring: the information from the captured packets can be used to detect some common mis-configurations of parts of the network. This includes hosts that act as routers, mis-configured address masks, host with network cards that are in promiscuous mode while they should not be and so on. From the same data bandwidth utilization information can be derived.

The two other main functions of Ntop are *Network optimization and planning,* and *Detection of network security violations*. These functions are not directly applicable for the purpose of SLA verification, and are therefore not discussed here.

Ntop has a integrated graphical user interface, in the form of a integrated web server. NTop users can make use of standard web browsers to access the information that Ntop makes available.

In terms of the IP SLA performance parameters described in section 5, Ntop can be used to verify only a limited subset of those parameters, and for some parameters that can be measured special measurement setups are needed. Parameters that can be measured between a pair of IP addresses are the throughput, the error ratio and the loss ratio. These measurements can either be based on user data that is sent anyway, or they can be based on measurement packet streams that are injected into the service especially for the purpose of measuring those parameters. Also the availability parameter can be derived from these measurements. Spurious packet rates at specific hosts in the network can be measured using Ntop. A measurement setup for doing spurious packet rate measurements required running Ntop at every host in the network. Spurious packets can then be detected by comparing the measurement results gathered at all of these hosts. NTop is in its current form not suited for doing delay and delay variation measurements, since it does not keep track of the precise timing relations of packet events in the network.

8.3 MIB-II

The IETF defined MIB-II [RFC1213] as a MIB that all IP compliant devices should implement. The document has the status of ‘full standard’, which is the highest level in the IETF process of defining standards. MIB-II has experienced wide deployment in IP capable equipment. Furthermore, all the groups of management objects in MIB-II are mandatory; there are no optional objects defined. It is the combination of these two facts that make MIB-II a good candidate for retrieving information about SLA conformance of IP systems. Since the publication of this standard the management information it contains has been updated once more, and the single document was split into a set of separate documents [refs, to, them]. Since this happened only fairly recently, at this date the new set of documents is not experiencing the wide deployment in a similar way as MIB-II.

MIB-II organizes its management information into ten groups of objects, these are the system, interfaces, at (address translation), icmp, tcp, udp, egp (exterior gateway protocol), dot3 (transmission) and snmp (simple network management protocol) groups.

The objects in the various groups of MIB-II can be used to verify some of the SLA parameters defined in section 5. How this can be done and what particular MIB objects relate to what particular SLA is discussed next.

The IP packet transfer delay nor the variation in packet delay can be verified using MIB-II objects. The detailed timing requirements that would be needed for this type of verification are not present anywhere in the MIB.

The packet error ratio and the packet loss ratio can to some extent be verified by examining the relevant MIB objects in the IP group and the interfaces group. An estimated value for the error ratio at a given interface can be found by dividing the ifInErrors value for that interface by the sum of ifInUcastPkts and ifInNUcastPkts. Although some information can of course be attributed to this value, it still is just a poor estimate, for the following two reasons. First, the counter values are determined only at a single point in the network. The definition of the error ratio SLA performance parameter is in fact a end-to-end parameter, not a parameter at a single point. Problems elsewhere in the network cannot be detected at the measurement point, while at the same time those problems will affect the real value of the end-to-end performance parameter. Second, the counters count packets, not octets. Differences in size between packets exists, and will introduce an error even in the measured value at a single point.

An estimated value for the loss ratio at a given interface can be found by dividing the ifInDiscards for that interface by the sum of ifInUcastPkts and ifInNUcastPkts. This computation suffers from the same problems as the previous one for the error ratio.

The spurious packet rate cannot be measured using MIB-II.

An indication of achieved throughput can be obtained using the objects provided by MIB-II. In the ifTable the number of sent and received octets are counted, and from two of these values and the time interval between the two measurements the octet rates in both directions can be computed. Also here this is a single point measurement, whereas the definition of the throughput performance parameter is a two point, end-to-end measure.

The availability of the IP service is another SLA performance parameter that can be approximated by looking at specific objects in MIB-II. Also in this case it is a single point approximation of an end-to-end performance parameter. Availability of a IP system (a router for example) can be verified by periodically checking the value of the sysUpTime MIB object. If the object cannot be read, or if a discontinuity is found in the object value, a service unavailability is detected. Also the approximated

duration of the service can be determined, and from that the overall service availability ratio can be computed.

A more detailed analysis of MIB-II and the objects it contains can amongst other places be found in [Stallings99].

8.4 Remote Monitoring (RMON)

The RMON standard is defined by the IETF and it enables the monitoring of complete network segments consisting of multiple hosts. A large variety of parameters and events can be monitored, and the monitoring process can be controlled from a distance (remotely), and also the results can be retrieved from a distance. RMON currently exists in two versions (RMON-1 and RMON-2) which are described in a set of four documents [RFC1513, RFC1757, RFC2021, RFC2074]. RMON-1 can only be used to monitor networks at the ethernet layer, whereas RMON-2 adds capabilities to RMON to also monitor layers 3 up to 7 of the OSI model, so, the network layer all the way up to the application layer. This makes RMON a potentially powerful monitoring technique.

Because of the multitude of different monitoring possibilities that RMON-1 and RMON-2 have, only those monitoring features that are of direct relevance to SLA parameter verification will be described here.

RMON cannot be used to measure transfer delay or variations in transfer delay, because RMON does not keep high resolution time stamps on any packet events that it can observe.

RMON can be used to determine error ratios and loss ratios within a single broadcast domain. This involves capturing a known stream of test packets close to the two measurement points of interest. Analysis of the captured packets can take place off line, and by comparing the captured packets at both points the error ratio and loss ratio of the test packets can be computed.

The spurious packet rate at a given measurement point can be determined with RMON, but only if a dedicated measurement setup is used for this purpose. This involves having RMON probes at every possible ingress and egress point of the portion of the network that is of interest. Of every packet leaving the network it has to be verified if it actually entered the network at one of the ingress points. This measurement setup is most likely a highly costly one.

RMON can be used for throughput verification purposes, because on a measured segment the achieved throughput from and to each system directly connected to the segment can be determined. Note that for a good estimate of the upper bound of the achievable throughput a test traffic source is needed. This type of measurement is then a active measurement.

8.5 Multi Router Traffic Grapher (MRTG)

The Multi Router Traffic Grapher (MRTG) [MRTG] is a tool that can monitor sent and received traffic at router interfaces, and present the results in a graphical form on a web page. MRTG can also monitor and graphically present the amount of errors that occurred on those router interfaces.

It is a stand-alone tool that collects and stores measurement information about routers. The tool also has a internal web server that serves the pages with the measurement results.

MRTG can be used for the verification of only a limited number of the IP SLA performance parameters that were identified earlier. In particular, the information that MRTG provides can be used as an indication for the values of the Throughput and Error Ratio. These will just be indications, since MRTG takes measurements only at a single point, whereas for a accurate measurement that is in

line with de definition of those parameters, a measurement method involving two measurement points is required.

9 Concluding Remarks

This section summarizes the key observations made in this deliverable.

The prime motivation for introducing SLAs into the Internet environment is the increasing complexity (both in number of configureable parameters as in total number of individual customers) of the Internet. In order to facilitate a far stretching automation of service management activities the SLA concept is needed. It is the central concept that enables Customer Service Management.

The SLA life-cycle has three phases: SLA creation, the operational phase and the removal phase. From a customer service management point of view, the operational phase is the most important.

The SLA concept has been applied to a particular instance of a service: the IP transport service. A service model for this service has been constructed, and the IP performance parameters that form the basis of the SLA have been identified, based on earlier external work. The accounting aspects of SLAs have been discussed, resulting in three accounting scheme categories: flat rate, reservation based and usage based accounting.

SLAs have to be verified, to see if both SLA parties keep within specified limits of behavior. How this can be done has been discussed for each of the SLA parameters identified earlier, and some general tools and techniques (including various public domain software packages) for this purpose were presented.

In summary, SLAs are an important concept for the Internet as it is evolving today, and in particular to be used in combination with Customer Service Management.

References

[Hartanto99] Felix Hartanto & Georg Carle. *Policy-Based Billing Architecture for Internet Differentiated Services*. In Proceedings of IFIP Fifth International Conference on Broadband Communications (BC'99), 1999.

[I.350] ITU-T. *General Aspects of Quality of Service and Network Performance in Digital Networks, Including ISDNs*. Recommendation I.350, March 1993.

[I.380] ITU-T. *Internet Protocol Data Communication Service - IP Packet Transfer and Availability Performance Parameters*. Recommendation I.380, February 1999.

[Mathis00] Matt Mathis and Mark Allman, *A framework for defining empirical bulk transfer capacity metrics.* internet draft, draft-ietf-ippm-btc-framework (work in progress, expires May 2001), December 2000.

[Odlyzko00] Andrew Odlyzko. *Internet pricing and the history of communications*, December 2000.

[Ntop] Luca Deri, Ntop measurement tool, web-site, http://www.ntop.org/

[RFC2330] Vern Paxson, Guy Almes, Jamshid Mahdavi and Matt Mathis. *Framework for IP Performance Metrics*. IETF RFC 2330, May 1998.

[RFC2678] Jamshid Mahdavi & Vern Paxson, *IPPM Metrics for Measuring Connectivity.* IETF RFC 2678, September 1999.

[RFC2679] Guy Almes, Sunil Kalidindi & Matthew J. Zekauskas. *A One-way Delay Metric for IPPM*. IETF RFC 2679, September 1999.

[RFC2680] Guy Almes, Sunil Kalidindi & Matthew J. Zekauskas. *A One-way Packet Loss Metric for IPPM*. IETF RFC 2680, September 1999.

[RFC2681] Guy Almes, Sunil Kalidindi & Matthew J. Zekauskas. *A Round-trip Delay Metric for IPPM*. IETF RFC 2681, September 1999.

[RFC2720] Nevil Brownlee. *Traffic Flow Measurement: Meter MIB.* IETF RFC 2720, October 1999.

[Verma99] Dinesh Verma. *Supporting Service Level Agreements on IP Networks*. Macmillan Technical Publishing, 201 West 103rd Street, Indianapolis, IN 46290 USA, 1999.

[Stallings99] William Stallings, *SNMP. SNMPv2, SNMPv3 and RMON 1 and 2*, third edition, Addison-Wesley, One Jacob Way, Reading, Massachusetts 01867, USA, 1999.

[IPPM-wg] IP Performance Metrics working group, Internet Engineering Task Force, 2001.

[IETF] Internet Engineering Task Force, *www.ietf.org.*

[OpenSource Open Source reference, e.g. www.opensource.org or 'The cathedral and the bazaar', ISBN 1-56592-724-9, 1999.

[MRTG] Multi Router Traffic Grapher, web pages, *www.mrtg.org*