Project Number: **IST-2000-26417**
Project Title: **GN1 (GÉANT)**

Deliverable D9.7

Testing of Traffic Measurement Tools

Deliverable Type:	PU-Public
Contractual Date:	31 August 2002
Actual Date:	23 September 2002
Work Package:	8
Nature of Deliverable:	RE - Report

Editors:

Simon Leinen	SWITCH
Victor Reijs	HEAnet

Abstract:

This deliverable is focused on flow based accounting mechanisms for highly aggregated traffic and tools for measuring the perceived quantitative Quality of Service. It provides an update from D9.4 on the description of tools and generic measurement infrastructure to support the measurements of user-visible SLS metrics.

Keywords: QoS, GÉANT, SLS metric, Flow measurement, Tools

Table of Contents

1. EXECUTIVE SUMMARY

This deliverable identifies a set of potential tools for traffic flow measurement and reporting for highly aggregated traffic as well as providing a general framework for monitoring the perceived quality of generic applications. It also outlines a number of tools for measuring the user-visible SLS metrics.

Traffic measurement within IP networks is used by Network Operations personnel to aid in managing and developing a network. It is flow measurement that provides a way for measuring and understanding the network's traffic. Traffic accounting mechanisms based on flows should be considered as passive measurement mechanisms. Information gathered by flows are useful for many purposes: understanding the behaviour of existing networks, planning for network development and expansion, quantification of network performance, verifying the quality of network service and attribution of network usage to users.

This deliverable reports on the recent results of the GÉANT working group, especially the Flow-based Monitoring and Analysis (FloMa) activity. This activity has formulated several applications scenarios for traffic measurements based on flow-oriented accounting mechanisms (e.g. RTFM, NetFlow, LFAP). The working group has proposed those scenarios as test-cases for flow-based accounting mechanisms.

Evaluation of accounting mechanisms is also presented starting with mechanisms based on NetFlow on different hardware platforms and also non-flow-based mechanisms like MAC, TOS/DSCP and bucket-based accounting.

A choice of different technologies can be applied on a network for the purpose of equipping it with QoS capabilities. Many applications rely on the underlying network to provide them with QoS guarantees, while problems occurring at the network level can damage the perceived quality of the application making them unusable to its users. Important QoS metrics that affect the quality of various kind of applications have been identified and a list of SLS metrics (available bandwidth, one-way packet loss, one way loss pattern, IP packet delay variation, one way delay) is proposed in order of importance. Because there is no standard way to measure each of the mentioned SLS metric, individual tools are available that try to impose measurements at various points in the network in order to calculate the status of each metric. Most of the available tools have been evaluated and are listed with this document while are correlated to the SLS metric(s) they can measure.

To perform measurement in a structured way, a measurement infrastructure is proposed. It must provide a flexible platform, support both active and passive measurements, provide security, be capable of correlating results from different management domains as well as providing results to other concerned parties, and help the users involved to discover the available measurement resources. In addition a suggested topology of the measurement infrastructure is also supplied.

Many tools were evaluated in the context of this work. A substantial description for each tool is provided in the annexes.

This document is an update to D9.4.

2. ACCOUNTING MECHANISMS FOR HIGHLY AGGREGATED TRAFFIC

Work on flow-based and other accounting mechanisms has started in 1999 as part of the Quantum Test Programme (QTP). Its main results were presented in Quantum Deliverable D6.2 [3] and GEANT deliverable D9.4 [87]. This chapter has been revised to include new developments, in particular in the area of standardisation efforts within the IETF.

The work carried out within the Quantum Test Programme (QTP) emphasised a substantial interest in new accounting mechanisms for highly aggregated traffic. This interest was sparked mainly by Cisco Systems NetFlow system, which had been introduced in 1997 and which was already being used by several groups in the European NREN community for various accounting tasks. Initial plans were for collaborative experiments with various accounting software packages using a workstation in the TEN-155 PoP in Geneva. However, this work didn't materialise in this fashion for both organisational and technical reasons. Instead, participants exchanged experiences from their respective work with flow-based accounting mechanisms at TF-TANT meetings and a mailing list. The resulting information was distributed on a Web site [89] that is widely known and used by the network accounting community both in Europe and elsewhere.

2.1 IPFIX Standardisation Effort

Starting in spring 2001, there has been some activity in the IETF community towards the standardisation of an IP Flow Information Export protocol. The existing RTFM (Real-Time Flow Measurement) standard [1] is based on a "pull" model, where usage data is collected from meters on demand of an analysis application. IPFIX should complement this with a "push" mechanism, where data is continually exported from routers or other packet-observing devices towards consumers of accounting information.

A BOF (Birds-of-a-Feather) meeting has been organised at the 51st IETF meeting in August 2001 in London to discuss the scope of this activity. The IPFIX working group was chartered in September 2001. Dave Plonka, one of the co-chairs of the group, maintains a Web site [90] with pointers to all relevant documents as well as archives of the mailing lists.

The group has produced several revisions of a requirements Internet-Draft [91], and received multiple individual contributions.

Because of the multitude of flow information export mechanisms already in existence, the area directors at the Internet Engineering Steering Group (IESG) asked the working group to select one of these, rather than develop a new protocol. The requirements document should be used as the basis for evaluation of the submitted protocols.

The selection process was presented at the 54th IETF meeting in Yokohama in July 2002. Protocol submissions had to be submitted by September 2, and the set of candidate protocols includes:

- NetFlow Version 9 [92][93]
- LFAP Version 5.0 [94][95][96]
- CRANE [97][98]
- Streaming IPDR [99][100]
- DIAMETER [101]

An evaluation team (which includes the author) will evaluate the protocol submissions, and the selection will be presented at the 55th IETF in Atlanta in November 2002.

2.2 Experience with Flow-Based Accounting (NetFlow)

The participants in the FloMA activity have shared experience that they had gathered locally. This has been extensively documented in the Quantum deliverable D6.2 [3]. The paper introduces various flow-based accounting mechanisms, general issues in using these for various accounting tasks, as well as descriptions of several available tools for post-processing accounting data.

2.3 Evolution of Accounting Mechanisms

During the course of this project, new accounting mechanisms became available both within and outside the context of flow-based methods. These mechanisms are presented in the sections below.

2.4 Flow-based Accounting Methods - NetFlow

While NetFlow has been defined and trademarked by Cisco Systems, it has become an industry standard for flow-based accounting, which has been adopted by several vendors of routers and specialised measurement probes, in many cases in addition to other accounting protocols. The following gives a list of products with the capability to generate NetFlow-compatible accounting records.

2.4.1 Cisco CPU-based routers

Cisco has implemented NetFlow on all software-driven platforms that support recent IOS releases, from the small SOHO routers to high-end backbone routers. Classical (non-sampled, nonaggregated) NetFlow has a significant impact on CPU load, only realistically supports STM-1 speeds on the 7500 platform, and can make it difficult to use other features such as differentiated queuing and dropping at high traffic rates.

2.4.2 Cisco 12x00 Gigabit Switch Router (GSR)

On the GSR platform, the level of support for NetFlow depends on the generation of line card engines used. The current high-speed (STM-16 and above) line cards only support NetFlow in either sampled or router-based aggregation modes. Future line card engines may add ASIC support for NetFlow and will thus be able to generate accounting records at high packet and flow rates.

2.4.3 Cisco PXF-based routers

The PXF (Parallel eXpress Forwarding) is a network processor which is the basis of several recent Cisco routers such as the 10000 ESR or the 7200 NSE-1. It uses a 4-by-4 systolic array of simple processors to execute different functions in a pipeline for multiple parallel packet forwarding paths. NetFlow accounting was one of the first functions that was implemented on the PXF. The PXF implementation is said to be able to maintain correct accounting at high packet and flow rates. As of June 2001, sampled NetFlow has not been implemented for the PXF architecture, although this was planned.

It should be noted that different products use the PXF in widely varying ways. High-speed devices such as the Optical Service Modules (OSMs) for the 7600 OSR run the PXF at six million packets per second (1.5 Mpps per pipeline), so that fewer than a hundred processing cycles can be performed in each pipeline step. It is questionable whether NetFlow accounting can be implemented in such settings. Other products such as the 7200 NSE-1 use the PXF at much lower rates, on the order of 0.5 Mpps, or 125000 pps per pipeline. At these speeds, the cycle budget permits implementation of NetFlow accounting in a single pipeline step.

2.4.4 Cisco Catalyst 6x00 Layer 2/3 switches

The Layer-3 switching technology in Cisco's high-end Catalyst switches is based on a variant of NetFlow. The PFC (Policy Feature Card) uses a flow cache to accelerate the Layer 3 switching decision. The components of the cache key are selected based on required granularity of the forwarding process. These devices have hardware support for the maintenance and export of NetFlow accounting data. However, they use a restricted version of the accounting data format, NetFlow v7, which leaves several useful fields undefined, such as the ingress interface index, source and destination AS numbers, TCP flags and subnet masks. Additional functionality to have these fields filled in should be integrated in IOS release 12.1(13)E, which is planned for publication at the end of September 2002.

The PFC2 implementation was found to have other issues that can complicate the use of NetFlow accounting, especially in places of high aggregation and high flow rates. With the PFC2, flows are expunged from the cache by a periodic process that is run every couple of seconds. Each run can generate a large amount of NetFlow packets, which are exported in a single burst. A random packet trace from a busy border router shows that 39 1428-byte UDP packets were exported in 473 microseconds over a Gigabit Ethernet interface. This corresponds to a rate of 960 Mb/s, so all packets in the burst were practically sent "back-to-back". It can be quite challenging to reliably transport such bursts through a network and process them on a computer, even though the mean rate of NetFlow export may be relatively modest.

2.4.5 Juniper M-Series routers

Juniper implemented NetFlow accounting in recent versions of its operating system - JUNOS. It requires the second generation of the centralised forwarding engine (Internet Processor II). Sampled NetFlow v5 as well as NetFlow v8 are supported. In the current implementation, the rate of accounting records generated is limited to 7000 flows per system to limit the impact on other tasks of the router's management subsystem. A particular feature of Juniper's implementation is that it can be specified through filter lists which packets are eligible for NetFlow accounting. This allows performing in-depth flow analysis of specific traffic with limited impact on router resources.

2.4.6 Foundry BigIron/NetIron routers

As with the Cisco Catalyst 6x00, the Layer-3 switching functionality in Foundry's products is based on a flow cache. Newer Foundry products have support for NetFlow accounting. First-hand experience with this could not be gathered, but there are claims that newer network processor-based (NPA) line cards can do full NetFlow (v5) at STM-16 rates. Foundry also supports InMon's sFlow protocol.

2.4.7 InMon probes

These are dedicated probes with 10/100/1000 Mb/s Ethernet interfaces. They generate Cisco NetFlow or InMon's sFlow accounting format. These probes are also available as a software package under RedHat Linux.

2.4.8 Bina "FlowBox" probes

These are dedicated probes with OC-3c/OC-12c/OC-48c interfaces that generate NetFlow v5 and v8, the latter with AS and prefix aggregation.

2.4.9 Sun Bandwidth Manager

This is a software product from Sun Microsystems that adds traffic management functions for Sun machines acting as routers or servers. It has the capability of transmitting traffic accounting data in NetFlow format [4].

2.5 New Non-Flow-Based Accounting Mechanisms

While the FloMA activity was started due to widespread interest in Cisco's NetFlow technology, and flow-based accounting mechanisms in general, the developments outside this area have also been taken into account. The other accounting methods are presented in the following sections.

2.5.1 Packet Sampling - sFlow, XRMON, PSAMP

Packet sampling is a potentially very attractive alternative to flow-based accounting mechanisms. Rather than trying to classify packets into flows, and hoping that this aggregation will make accounting information manageable, packet sampling performs data reduction by simple sampling techniques. The data exported can consist of time-stamped packet headers or entire packets, possibly decorated with information such as input/output interfaces.

Compared with flow-based accounting techniques, packet sampling is more amenable to hardware-based implementation in fast routers. Statistical studies have shown that the measurement error introduced by sampling depends on the number of samples seen, so for large amounts of traffic even large sampling factors result in reliable results. Using statistical methods for applications such as volume-based charging, may be hard for customers to accept, but when the underlying method is simple, and the statistical properties of the sampling mechanism have been understood to be sound, the performance of such methods compare favourably with that of more complex mechanisms, which are designed to be accurate in the common case but often fail to operate reliably in practice, either because they cannot always handle the load, or because the implementation has other flaws.

Known documented packet sampling mechanisms and protocols include:

- InMon's *sFLOW* [102]
- Hewlett Packard's *XRMON*

In the IETF, the `psamp` (Packet SAMPling) working group has been formed in the Operations and Management Area to pursue standardisation of protocols in the area of packet sampling. So far the group have produced a framework document [103].

2.5.2 MAC Accounting

Supported by Cisco in IOS 11.1CC and later, this accounting mechanism maintains, at each interface, a table of input/output byte and packet counters indexed by link-layer (MAC) address. These counters can then be accessed either through the command-line interface (CLI) or through a specially defined MIB (CISCO-IP-STAT-MIB).

This mechanism is attractive in situations where the traffic between a router (or a specific interface on a router) and several of its neighbours on a LAN should be measured separately. A good example is an Internet exchange point (IXP) where a router connects to several other ISPs routers over a shared interface, but the system should be useful in other situations such as a set of servers sharing an Ethernet link to one or several routers.

An open issue is whether this can also be used to generate interesting per-host statistics in large bridged environments. The size of the tables may make this difficult, both for the router counting packets at high rates, and for the management station that has to extract information from these tables.

2.5.3 TOS/DSCP Accounting

Cisco's IOS 11.1CC and later also implement a mechanism that is similar to MAC Accounting, but that uses packets TOS value as a key. Where the TOS bits or the Differentiated Services Code Point (DSCP) bits are used to select different QoS treatments, TOS accounting provides an easy way to account for traffic per QoS class over a given interface.

As the number of DSCPs actually selecting defined services is usually rather low, access to these counters through SNMP is relatively convenient.

The functionality of TOS/DSCP accounting is also included in the Differentiated Services MIB [104].

2.5.4 Bucket-Based Accounting

Another very powerful accounting mechanism is based on a two-step approach:

When a route is learned from a routing protocol and installed into the router's forwarding table, the router can store an additional accounting tag - the bucket selector - in the forwarding table based on routing protocol information. Typically, this is done using route maps or similar policy definitions on incoming BGP announcements. When a packet is received at an interface, the destination address has to be looked up in the forwarding table so that the router can determine where to send the packet. If a forwarding entry is found, the router extracts not only the next-hop information that is necessary for forwarding the packet, but also the bucket selector tag. If such a tag is found, the packet and byte counters for the corresponding bucket are updated. The counters can be accessed through CLI or SNMP tables.

Through this division of labour between the control and forwarding planes, complex accounting rules can be specified through route maps, while the actual accounting overhead is very low, and independent of rule complexity.

This approach has been implemented by Cisco and Cabletron under the name BGP Policy Accounting, as well as by Juniper, where it is called Destination Class Usage. Current router implementations seem to share some common properties:

The number of buckets is limited to a small number:
- Cisco - 8 buckets,
- Juniper - 16 buckets,
- Cabletron - 24 buckets.

The bucket is selected by the destination address of incoming packets. This is natural, because the destination address is what has to be looked up in the forwarding table. This means that only "destination classes" are considered, not "source classes". When the source of a packet is of interest for accounting, one has to rely on other means.

In many cases, the bucket-based accounting mechanism can still be used, namely when source classes can be mapped to individual interfaces.

The rules that assign a bucket selector to a prefix in the forwarding table are only run when the route is installed by BGP. This means that routes that are learned by other means, such as static routes or internal routing protocols such as OSPF or IS-IS cannot be used by this system.

2.6 Various Measurement Tasks and Accounting Mechanisms

The Flow-based Monitoring and Analysis (FloMA) activity in QTP had formulated several application scenarios for large-scale traffic measurements based on flow-oriented accounting mechanisms. This section attempts to present the evaluation of several application scenarios for flow-based traffic measurements.

2.6.1 Traffic Statistics at Exchange Points (Scenario 1)

The goal was to gather per-peer traffic data at a public exchange point, i.e. a situation where an ISP exchanges traffic with several other ISPs over a single physical interface.

2.6.1.1 NetFlow Solution

Per-peer statistics at an exchange point interface can be produced from NetFlow v5 accounting data, by using the source and destination AS information in the accounting records. With NetFlow export, one can select either the origin-AS or the peer-AS to be exported for the source and destination address of each flow accounting record.

2.6.1.2 Alternative Solution: MAC Accounting

As an alternative, MAC accounting could be configured on the interface towards the exchange points. The MAC addresses of all peers can be found using the bgpPeerTable [16] and the ipNetToMediaTable [17]. Then, traffic to each peer can be polled from the cipMacTable under the corresponding MAC address.

2.6.1.3 Comparison of the Approaches

The MAC Accounting approach compares favourably to the NetFlow solution because:

- It is much simpler to implement on the accounting platform;
- Accounting overhead in the router is significantly lower;
- Counters are updated in real time;
- It accurately reflects the actual senders or receivers of traffic at the exchange point, while the NetFlow approach, in the general case, is unable to find the actual sending peer for incoming traffic. This is because NetFlow uses the local BGP table to determine the preferred egress peer for an address, but that does not always correspond to the ingress peer through which traffic from this address reaches the network.
- It is easy to notice traffic received from routers at the exchange point that one does not peer with, which could be a sign of a misconfiguration ("pointing default").

Disadvantages of the MAC Accounting solution include: If flow-based accounting is already active on the exchange point router for other applications, then it would not mean any additional overhead for the router to use this information for per-peer statistics. MAC Accounting, on the other hand, would mean additional work, although the performance impact should be negligible. Also, post-processing of MAC Accounting information must keep track of peers MAC addresses.

2.6.2 Accounting for Volume-Based Charging (Scenario 2)

Traffic should be accounted separately for pairs of (customer organisation, network region), such that on-net traffic could be charged at a different rate from off-net traffic, or traffic over a research backbone can be charged differently from traffic over commodity transit.

This scenario has been formulated in a deliberately vague way, because one can imagine many different volume-based charging schemes. In one scheme that is actually in use in some European NRENs, traffic over certain - expensive - links is counted separately for each customer organisation, and the volume-based fee is computed based on these amounts of traffic. It is important to note that organisations are charged not only for the traffic they send over these links, but also for the traffic they receive [11][14]. The per-volume fee can also vary over time-of-day or day-of-week (peak/off-peak).

Within the Volume-Based Charging context, three solutions may be considered:

2.6.2.1 NetFlow Solution (1): Accounting on External Border Routers

In this approach, NetFlow accounting is run on routers connected to the expensive links. Accounting data is then post-processed to extract those flows that traverse an expensive link and determine the customer that the traffic should be accounted to. If routes between the network and its customers are exchanged using BGP, the customer can be identified by its AS number. If other routing mechanisms (such as static routes or an IGP) are used towards customers, the customer can be determined from the source/destination addresses by lookup in a configured table of customer address ranges.

2.6.2.2 NetFlow Solution (2): Accounting on Customer Border Routers

In an alternative implementation, NetFlow accounting runs on the routers to which customer networks are connected. This inverts the post-processing problem: The customer can be identified simply by looking at the interfaces that the flow traverses; it then has to be determined whether the flow crosses an expensive link. This can usually be done using AS information from the flow accounting records, more straightforwardly when the router is configured to send the neighbor (peer) AS rather than the origin AS. However, this is fundamentally unreliable for traffic received by customers; outbound routing information may not accurately indicate the links over which inbound traffic will arrive.

2.6.2.3 Bucket-based Accounting Solution

Accounting mechanisms such as Cisco's BGP Policy Accounting or Juniper's Destination Class Usage seem ideally suited for this accounting task, but at least today, they have some limitations that may make their application less than straightforward.

If only sent (outbound) traffic is charged for, it is sufficient to measure at the customer ingress interface, and set up an accounting map such that traffic towards expensive links is counted in particular buckets. The only requirements are that each ingress interface is dedicated to a single customer (account) and that a bucket can be set up for each expensive link (each charging class of expensive link). In other words, multiple expensive links that are charged equally can be grouped in the same bucket.

The requirement to charge for received (inbound) traffic, where it exists, severely complicates matters however. It would work to run the accounting mechanism on the downstream (towards the customers) interface of every expensive link, and set up the destination address bucket map so that each customer maps into a separate bucket. The problem with this is that the number of buckets supported must be at least the number of customers with separate accounts. This is a severe limitation, given that the current numbers of buckets supported by the different implementations is between eight and twenty-five, and many networks have more customers than that. Another limitation is that for many NRENs, customer routes are not learned using BGP today, so setting up these bucket maps may be very difficult.

The following improvement to current bucket-based accounting mechanisms may be considered to make it easier to implement charging for traffic in both directions over expensive links:

If the number of buckets could be increased to match typical numbers of customers, then it would be easier to realise this charging scheme by counting traffic at an expensive link, and break it up for each customer.

- To make it possible to break up traffic for each customer, it would be helpful to allow the mapping of addresses to buckets using means other than BGP, because not all networks use BGP to exchange routes with their customers. For instance, static routes and routes learned over certain IGP adjacencies should be markable with a bucket's index.
- A more scalable solution for this accounting problem would involve measuring on customer routers (or customer aggregation routers) only. But determining whether traffic crosses an expensive link is only easy in the outbound direction. It might be useful to add a variant of the bucket-based accounting mechanism that would look at the source, rather than the destination address. As mentioned above, it is fundamentally problematic, and usually unreliable, to deduce the ingress path of a packet from one's own routing table entries for the source address.
- An alternative solution for the ingress problem would be to mark all packets on ingress so that traffic from expensive links can be distinguished from cheap traffic. One possibility would be to use distinguished DSCPs for this. Then, DSCP/TOS accounting can be used on customer interfaces to measure the amount of expensive traffic for each customer.

2.6.3 Abuse/DoS Attack Detection (Scenario 3)

Anomalies in traffic patterns should be used to detect abuse of the network, such as (distributed) denial of service (DoS) attacks, large-scale break-in attempts or network-wide scans for vulnerabilities.

If one wants to detect network abuse such as DoS attacks, or attempts to break into computers on customer networks, by looking at accounting data, flow-based schemes such as NetFlow provides a very good basis for that. One can use higher-layer information such as TCP port numbers, ICMP types and codes etc. to look for signatures of known attacks, or run sophisticated correlation algorithms to attempt to detect extraordinary traffic patterns.

Large-scale phenomena, like distributed denial-of-service (DDoS) can often be detected by looking at traffic aggregates.

DANTE uses one-minute probes of NetFlow data on TEN-155's routers to watch for extreme peaks of traffic between a pair of AS-es and generate alerts in case of DoS attacks [15]. Because transit networks carry a lot of traffic, which cannot be analysed at line rate, the advantage of sampling has been taken, when coding this tool. This approach does not necessitate full flow-based accounting, but can also be based on less expensive accounting mechanisms, such as sampled NetFlow, routeraggregated NetFlow, or bucket-based accounting schemes.

The use of the higher-layer information provided by classic NetFlow, would make it possible to provide much of the functionality known from traditional Intrusion Detection Systems (IDS), but at points of high traffic aggregation. On the one hand, this seems attractive because entire transit networks, including the external traffic of many customer networks, can be monitored at a single place. But on the other hand, a transit network often has few possibilities of reacting to security problems, other than by adding filters to prevent abusive packets from continuing to flow. For DoS attacks, such filters often achieve exactly what the attacker had intended, namely deny service to a given user of the network.

2.6.4 Long-Term Traffic Analysis (Scenario 4)

Useful information about the long-term changes of traffic patterns should be extracted, with the goal of being able to anticipate the take up of novel applications on a scale that significantly impacts traffic, and to make projections on the usefulness of potential interconnections with other networks.

An important contribution of flow-based accounting mechanisms to network engineering was that it made it possible to decompose the traffic on different parts of the network according to higher-layer information, and thus get at least a good estimate of the contribution of different application protocols to network load.

Knowing more about the protocol mix on the network can be very useful for the operation of the network. For instance, the presence of multiple high-volume NNTP flows over the same link can point to a possibility of optimising the USENET distribution mesh. If, for instance, a link carries lots of traffic in a few bulk flows, one could react by encouraging users - through tariffs for instance - to move their traffic to off-peak times, or to mark their traffic for lower-than best effort treatment. Detection of the emergence of new applications, such as peer-to-peer file sharing protocols in Windows 2000, can provide hints as to upcoming changes in traffic patterns that are useful for long-term capacity planning.

There are known limitations in flow-based application recognition. Some newer applications, such as H.323-based [19] multimedia communications, use negotiated, rather than well-known, port numbers, so to detect these applications reliably, one would have to listen in to the control stream where those connections are set up. Since flow-based accounting mechanisms generally do not permit this, one has to rely on heuristics such as those used in FlowScan for Napster recognition [12] and in Fluxoscope for passive-mode FTP [13].

Even without, or with incomplete, information on application protocols, the distribution of flow frequencies, flows duration and flow sizes in terms of packets and bytes, presents a useful characterisation of the network usage or the load characteristics at different points in the network.

2.7 Conclusions

Flow-based accounting mechanisms such as Cisco's NetFlow have proven extremely useful to research and other network operators over the past few years. The present deliverable tried to present some of the knowledge that has been collected through the exchange of experience between NRENs and DANTE in the TF-TANT and TF-NGN groups. The FloMA Web site [89] serves as a focal point for the dissemination of accounting and in particular NetFlow-related information.

Standardisation of IP Flow Information Export is well underway in the IETF, and input from the GEANT community has been and will be contributed to this process, both during specification of the requirements and during the evaluation phase, which is starting now.

3. QOS MONITORING AND SLS AUDITING

This chapter provides the results of a study on the perceived quantitative quality of generic applications. These demands result in the QoS needs of generic applications (user-visible SLS metric). To determine if the network can provide the QoS, the tools and a measurement infrastructure to measure this are needed.

3.1. The QoS model

Before going into depth a general description of the scope of the TF-NGN activity can be given by using the following picture:

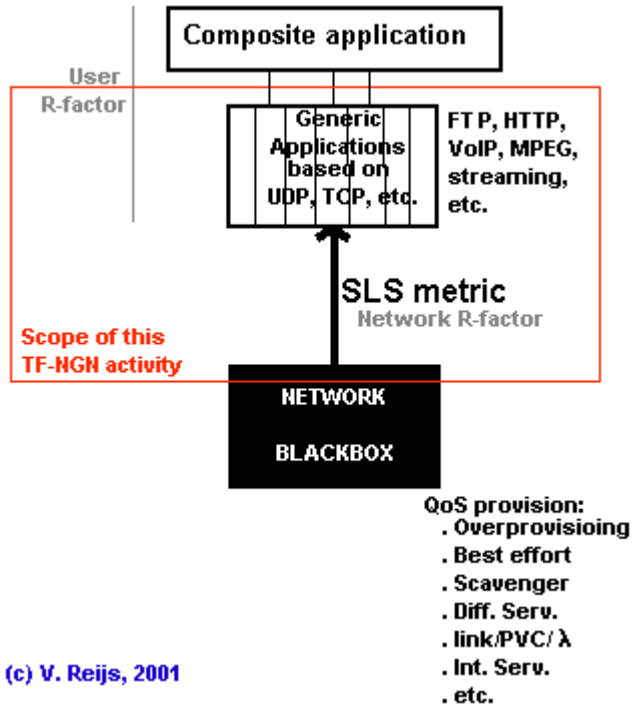

Fig. 1 The scope of TF-NGN QoS activity

The basic idea underlying this scope is that the network is considered as a blackbox. This blackbox delivers a certain level of SLS and determines the user-visible SLS. This activity has no interest in how QoS is provided by the network (i.e. by over-provisioning, differentiated/integrated services or dedicated circuits, see also [67]) and also has no interest in how these will be managed (that work is left to the specific QoS feature implementations).

On the application side, this TF-NGN activity deals with generic applications (such as FTP, VoIP, streaming) and not with composite applications (like video conferencing, database access), thus in principle voice and video are seen as different generic services. Composite applications are covered by the Internet QoS workgroup [21] and [81]).

3.1.1. Network technology based CoS metric

To provide a certain level of QoS towards the application layer, the network layer can use different technologies to provide a certain SLS. A number of technologies are available but the most widely used are over provisioning and Differentiated Services (e.g. Premium IP). Each of these technologies have their own control methods so that the SLS metric values can be achieved.

Overprovisioning will use for instance the control method 'current link load capacity' as CoS parameter. Differentiated Services will have control parameters like queue weight, queue priority, queuing method, queue length and drop probabilities, rate policing and link capacity. Tools also need to be developed that can check the threshold defined for these network technology based CoS metric values.

To be able to predict the SLS metric values, an understanding of how the mapping is done from these network technology based CoS metric values is needed. Although this is an important task, this work is not covered by this TF-NGN activity. This work is covered by the TF NGN activities that are specialized in certain network technologies (IP Premium, LBE and Overprovisioning activities).

3.1.2. Application based CoS metric

There is the issue that the SLS metric has to be translated into application based parameters. How to translate an SLS metric value into an application based CoS metric is the responsibility of the application. For instance, for TCP and voice a model is available to translate the SLS metric into application based CoS Metric: goodput (section 3.2.1) and Mean Opinion Score (MOS, section 3.3.1.6). For video streams such model is not yet available but it is important to have one (section 3.3.1.6).

The application layer can be divided into two parts; generic and composite. Each generic application depends on a multidimensional space of the SLS metric (like for voice: packet loss, packet loss pattern, available bandwidth, delay, delay variation, recency and codec). The perceived composite application CoS depends in some way on the perceived human interaction with the generic application. For instance in video conferencing, audio can be the most important, and in that case the video could be allowed to have a lower quality. Lip synchronization is also important in this specific composite application. Two way audio is another example of a composite application: in this case talker overlap becomes an issue. These interaction issues are called *User R-factor* (see also section 3.2)

3.2. Perceived quantitative quality of generic applications and SLS metric

Most applications, which seem to be designed for LAN environments, do not take into consideration the QoS provided in wide area IP networks. The scope of this activity is to gather information on which QoS metrics are important for generic applications, like:

- TCP based applications (FTP and HTTP),
- UDP/RTP based applications ($V^{(2)}$oIP, H.323 [19], MPEG, streaming, access grid, etc.)
- Other possible applications (computational grids, games and tele-immersion).

It is important to distinguish between two issues when referring to perceived (user-visible) quality of generic applications:

- A low QoS in an underlying network can hamper the application in such a way that it does not work properly. Implementation issues, protocol time-outs or other similar behaviour may cause this (*Network R factor).*
- An application can become unusable for human interaction. For example this may happen if due to long delays the effective conferencing between the parties is not possible anymore or if interaction between generic applications adds dependability **(*User R factor*).** In the text below, this type of issue is marked with *.

This document is primarily concerned with the first issue, because most of the points in the second issue are out of control of the network layer (except for the misconfiguration issues, i.e. when the delay is too long due to improper routing configuration). In the further consideration the assumption is made, that the network **cannot** overcome the light speed limitations or effects caused by the application hardware or software implementations (such as encoding or decoding delays).

3.2.1. Quality of TCP based applications

A theoretical model exists concerning the determination of the goodput of a TCP session (the Bulk Transfer Capacity) using normal IP networks [22]:

$$B(p) \approx \min\left(\frac{W_{\max}}{RTT}, \frac{1}{RTT\sqrt{\frac{2bp}{3}} + T_0 \min\left(1, 3\sqrt{\frac{3bp}{8}}\right) p\left(1 + 32\,p^2\right)}\right)$$

Where:

B(p): TCP goodput [packets/s]
W_{max}: maximum buffer size of receiver [packets]
RTT: Round Trip Time (comparable to 2*OWD) [sec]
b: number of packets that are acknowledged by a received ACK (b is typically 2)
p: probability that a packet is lost (comparable to OWPL)
T_0: time-out for retransmitting non-acknowledge packets [sec]

The 'probability that a packet is lost' (p) can be the result of packet loss due to congestion/queues or due to Bit Error Rates (BER) on links. The BER can be converted into packet loss, when knowing the packet size.
A JavaScript implementation has been made of the above model [23]. This implementation also holds the approximation formula of Mathis [74], however this approximation **only** works when no time-outs occur.

If a bad TCP connection is defined as being a connection that has half the optimal TCP goodput, then the OWPL is not allowed to be higher than 10^{-6} for 2 MByte buffers or 10^{-3} for 64 kByte buffers.

3.2.2. Quality of UDP/RTP based applications

Some information is available on UDP/RTP based applications. The information below is gathered from recommendations or documents. They concern voice, voice over IP and video.

- Voice without FEC/PLC (G.711, G.723.1, etc.)
  - The effect of one-way packet loss is over twice that of jitter. (G.711) [24]
  - One-way packet loss and delay variation affects subjective judgments more than one-way delay does. [25]
  - A change of 0.01 in one-way packet loss was worth a change of 220 msec in one-way delay. An one-way packet loss of 0.05 was unacceptable to the consumers, but an one-way delay of over 400 ms was acceptable, on average. [26]
  - Talker overlap problem (**User R factor*)
    The general idea assumes that a one-way delay of 0 to 150 ms means good interactivity, 150-400 ms means tolerable and 400 ms is bad interactivity.
- Voice with FEC/PLC (G.711)
  - A one-way packet loss of 0.1 is acceptable [75].
- VoIP quality
  MOS ([27] and [28]) which is a function of (see also section 3.3.1.6):
  - one-way loss pattern,
  - one-way delay and
  - delay variation.
- Synchronization between video and audio (**User R factor*)
  - Lip synchronization needs to be in the order of 1 to 2 video frames (around 50 msec).

- General video quality
  - A few recommendations concerning video quality (based on the assessment of spatial and temporal details) could be found in: ANSI T1.801.03 [29] and ITU-T P.910 [30] and [69].
- MPEG-2 without FEC
  - One-way packet loss $<10^{-5}$ for VHS quality video. [31]
- MPEG-2 with FEC [32]
  - One-way packet loss $<10^{-3}$ for VHS quality video:

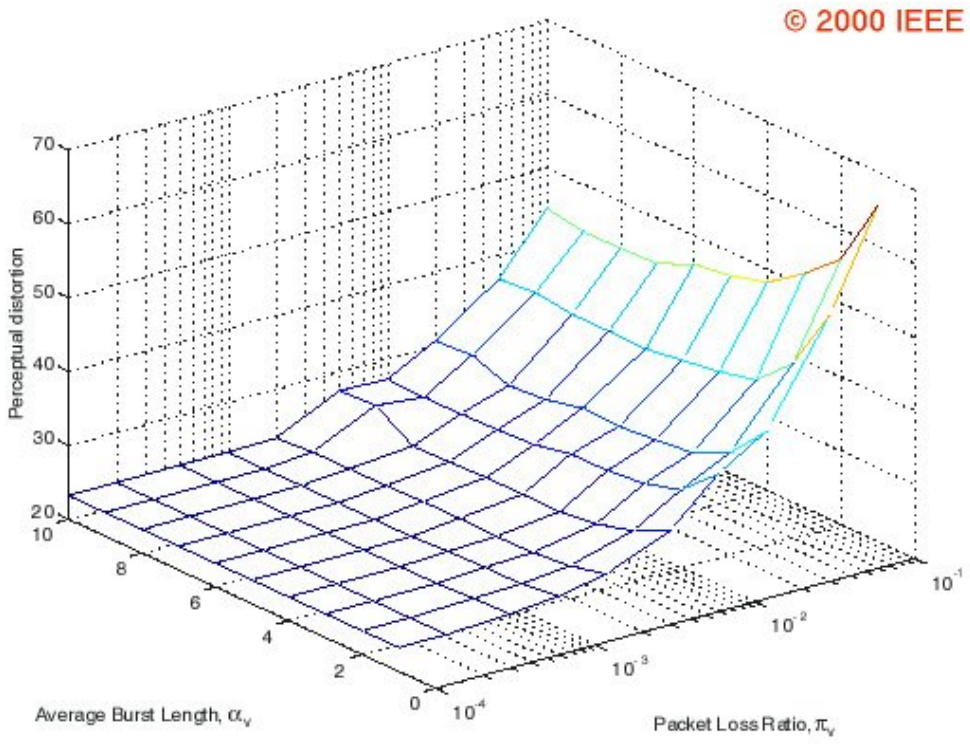

Fig. 2 The calculated perceived quality of MPEG-2 streams depending on packet loss (OWPL) and loss pattern (OWLP)

3.2.3. Quality of other possible applications

Not much information is available on the other possible applications. One of the user groups trying to define SLS metrics comes from the game environment. Their opinions are as follows:

- "When I work with game companies, I like to design for a 200 to 250 msec RTT and tolerate gracefully the rare instances when RTT might hit 1 second." [33]
- "An RTT over 150 msec is unacceptable." [34]

3.2.4. Quantitative Quality of Service as defined in SEQUIN project

The SEQUIN project [41] provides a QoS definition that reflects users' needs. The definition is the basis for the implementation of an end-to-end approach to QoS that is independent of the transport technology and will operate across multiple domains. Within the project work a review of research status on QoS international bodies (IETF, ITU-T) has been done to analyze their approach to the definition and measurements of IP performance parameters. Besides, a few interviews with various users have been carried out to understand their requirements for QoS.

Within the SEQUIN project, Pan-European groups of users were asked how they defined quantitative QoS. These can be summarized as (in order of priority):

- Guaranteed bandwidth (comparable to available)
- One way delay
- Jitter (comparable to ipdv)
- Packet loss

It is worth noting that packet loss is at the bottom of the priority list, much lower than would be expected. This could very well mirror current low one-way packet loss between NRENs.

3.2.5. SLS metrics

The above considerations creates the following list in order of importance of SLS metrics:

- Available bandwidth [76]: In general, each application requires a certain amount of bandwidth to transport its data. Issues that can influence this metric are link capacity, utilisation and rerouting due to link problems. The detailed definition of the available bandwidth has not been finalised yet. A possible definition could be "The available bandwidth is the bandwidth usable by an application, without incurring packet loss due to congestion and negligible IPDV"
- One-way packet loss (OWPL) [35]: One-way packet loss is primarily due to congestion (packets dropped in the bottleneck) or faults in circuits/interfaces. One way packet loss is a very problematic issue for almost all applications. Although TCP based applications can handle packet loss, the goodput (perceived quality) is largely determined by this metric. One of the examples may be a MPEG-2 stream without FEC (**not** designed for impaired networks) - a one-way packet loss of more than 10^{-5} makes that implementation of MPEG-2 unusable.
- One-way loss pattern (OWLP) [36] or [37]: One-way loss pattern is primarily due to congestion (packets dropped in the bottleneck, packet reordering, etc.) or faults in circuits. Minimising the congestion and eliminating all circuit or hardware errors is an essential issue.
- IP packet delay variation (ipdv) [37]
  Ipdv is another important QoS metric. It is mainly caused by buffering in IP routers and will be determined by congestion and prioritisation of IP traffic.
- One-way delay (OWD) [39]: Networked applications have to be able to handle at least a one-way delay caused by fiber circuits over half of the earth circumference: which is approximately 110 msec (to ~165 msec in case of bad coverage/routing between IP networks). The main requirement for applications is that implementations and protocols should, by definition, be able to handle this delay.
  Excessive delays need to be controlled in networks by using the shortest IP path possible and minimizing the delays (due to load, link speed, hardware implementation or access technologies).

A lot of work has to be done on this subject. Cooperation has been established with the Internet2 QoS Working Group [21]. More information on the above topics can be found in [43].

3.2.6. Simulation of an impaired network

A tool that allows controlled, reproducible experiments with network performance sensitive (or adaptive) applications and control protocols in a laboratory setting has been developed by NIST and is available at NIST Net [44]. This tool is a network emulation package that runs on Linux and allows a single PC to act as a router to emulate a wide variety of network conditions. By operating at the IP level, NIST Net can emulate the critical end-to-end performance characteristics imposed by various wide area network situations (e.g., congestion loss) or by various underlying subnetwork technologies.

This tool can simulate:

- bandwidth limitations (comparable to available bandwidth)
- packet loss (comparable to OWPL)
- network congestion
- packet duplication
- jitter (comparable to ipdv)
- delay (comparable to OWD)

No experience has been gained yet with this tool or similar other like dummynet [71], because of the TF-NGN's priority to make use of existing results instead of performing tests.

3.3. Tools for measuring the user-visible SLS metric

This section covers tools that are important for measuring the user-visible SLS metric.

Section 3.2 provided an overview of which user-visible SLS metrics are important to measure. Beside the metric itself, the tool granularity is also important. For example, when testing new IP features a much finer granularity is needed than for determining the user-visible behavior of an application.

An example of this difference in needed granularity is delay and delay variation. For testing new IP features a granularity of around 0.1 to 1 msec for these values is needed to fully debug and understand the feature, but applications are most of the time not interested in a granularity of lower than 1-5 msec.

The following granularity is proposed for the user-visible SLS metric:

- Available bandwidth: 300 kbit/s
- One-way packet loss (OWPL): an order of magnitude
- One-way packet loss pattern (OWLP): unknown (not a single numeric value)
- IP packet delay variation (ipdv): 1-5 msec
- One-way delay (OWD): 1-5 msec

There is an ongoing study in TF-NGN [45] with regard to tools that can measure the user-visible SLS metric, and the results are continuously updated.

3.3.1. Overview of tools

This overview only provides an impression whether necessary tools exist rather than an assessment of usability for measuring the user-visible SLS metric. This gives an overview of future work that needs to be done in stimulating the development of methods and/or the implementation of certain tools.

The following types of tools are important for this environment:

1. Tools that do the actual testing (can be active or passive measurement points); examples are RIPE TTM, ping and nprobe.
2. Tools that do composite tests (like MOS for audio, managing CoS service); examples are VQmon and Chariot.
3. Tools that are presenting raw data (coming from other tools) in graphs; an example is QoSplot
4. Tools that can be used as a measurement infrastructure, example is NIMI

The following sections will focus on the first two types of tools. The third type is covered in more detail in Annex 2, while the forth type is mentioned in section 3.4.1

3.3.1.1. Available bandwidth

The pathload tool can determine this metric. Some experience has been gained using such tools [76 and annex 2]. The end-to-end methodology it uses is called Self-Loading Periodic Streams (SLoPS) for measuring available bandwidth. The basic idea in SLoPS is that the one-way delays of a periodic packet stream show an increasing trend when the stream's rate is higher than the available bandwidth.

There are some known tools measuring other bandwidth metrics, such as Bulk Transfer Capacity ([47] and [48]), the capacity of each link in a path (pathchar, clink, pchar, pipechar and nettimer) or the bottleneck bandwidth (bprobe, cprobe [46] and pathrate).

3.3.1.2. One-way packet loss (OWPL)

The systems than can measure this metrics include RIPE TTM [49], Surveyor [73], RUDE/CRUDE/QoSplot and Chariot [50].

3.3.1.3. One-way loss pattern (OWLP)

Chariot [50] can provide insight in this parameter.

3.3.1.4. IP packet delay variation (ipdv)

This metric can be measured by tools that are based on RFC1889 [51], such as Chariot, RIPE TTM [49], RUDE/CRUDE/QoSplot [78] as well as MBone conferencing applications [52].

3.3.1.5. One-way delay (OWD)

One-way delay measurements can be performed in a several ways ([75] and [82]), though two main synchronisation issues must be handled carefully:

- Clock synchronization.
  Several methods can be used to achieve synchronization between distant locations:
  a) by 'internal' (to the measuring device) methods like an atomic clock.
  b) by 'external' methods like GPS, clocking derived from the telecommunication carrier (from SDH, as an example), frequency from radio or television stations, or IP networked solutions (such as NTP).
  For all methods it is very important to have the path between 'time source' and 'location where the time is used' as stable as possible (no delay variation, noise or other effects that can affect the signal). This is especially true for networked solutions.
  Some methods are preferred over other more complex methods; this is the reason why only two methods are pursued: GPS and networked solutions (such as NTP).
- Clock skew and clock adjustments
  Even though clocks are synchronised at regular time intervals, they will drift in time between synchronisations (clock skew). Another reason why clocks could be given the wrong time is due to the fact that it is not possible to make large step time changes in a computer. One has to do this in a slow way to keep the integrity of the file system; this is called clock adjustment. So even when using a GPS for synchronisation, these two issues can have an effect [84].

To measure the OWD with a fine granularity, precise clock synchronisation on both sides is required. This can be done with GPS receiver such as with RIPE TTM [49], Surveyor [73] or by using PPS [53].

There is also the choice of using NTP synchronization (see also [82]). The networked clock synchronisation method should both work in symmetric and asymmetric paths (which can be common in IP networks). It is important to note that Chariot is not able to determine the one-way delay in a setup with asymmetric path (their clock synchronisation method is not able to synchronise using such asymmetric path) [83].

TF-NGN is testing tools, which could provide results with sufficient granularity (1 to 5 msec). The use of a nearby NTP-server (verification with RUDE/CRUDE/QoSplot [78]), IPMP [64] or Ipanematech tool [79] is being tested. This issue of asymmetric path will be studied in more detail using these tools.

3.3.1.6. Composite tools

Some tools can provide a composition of a user-visible SLS metric.

Audio/VoIP MOS:

- Chariot (using active measurements) provides a modified MOS for VoIP [55], based on OWD, jitter buffer, OWPL, OWLP and codec.
- VQmon (using passive measurements) also provides a modified MOS for VoIP ([70] and [86]), based on RTT, jitter buffer, OWPL, OWLP, packet loss distribution, recency (the way a listener would remember call quality) and codec.

Video MOS:

- No tool yet exists that determines the Video MOS based on the SLS metric. The ITS Video Quality Measurement (VQM) tool [56] is more concerned on the video picture quality side instead of the network issues that underlay this. The Video Quality Experts Group [80] is also working in this area.

CoS management:

- A tool (Taksometro, see Annex 2) has been prototyped by DANTE for monitoring the behaviour of the different Classes of Service (CoS) implemented on a network. The tool is enabled to monitor the following list of network technology based CoS metric which can quantify any CoS type:
  - packet delay,
  - packet delay variation,
  - bandwidth utilisation and
  - packet drop

3.3.2. Concatenation of measurement results

In a multi-domain environment it is essential for the users to have an overall idea of the end-to-end performance; this is achieved by measuring individual domains and then concatenating the measurements results from the individual domains. This depends on the assumption of independent distribution of the measured value for the different domains. On one hand, since the next domain does not explicitly know about treatment in the previous one you could say that the independence assumption should be true. On the other hand, if there exists global congestion oscillation phenomena the assumption would be incorrect. The following text outlines a proposal for the concatenation of measurement results examining the user-visible SLS metrics (assuming independence):

- Available bandwidth.
  The minimum average value of the available-bandwidth$_i$ (measured per domain) will provided the overall average available-bandwidth$_t$.
- OWPL
  $OWPL_t = 1-(1- OWPL_i)*(1- OWPL_{i+1})*...$ For small values of OWPL, the first order approximation of the $OWPL_t$ is the sum of $OWPL_i$.
- OWLP
  It cannot be concatenated because it is not a single numeric value. This has to be measured end-to-end.
- ipdv
  If looking at the average $ipdv_t$, the average $ipdv_i$ can be summed per domain together.
- OWD
  If looking at the average OWD_t, one can safely sum the average OWD_i per domain together.

So again it is very important to stress that we are only talking about the average values (not about 95-percentile values, for example). Also no real concrete information is known about possible

distributions of these concatenated values. Furthermore the independence of the values over the domain can cause issues with the concatenation.
It looks that perhaps end-to-end measurements can take a way most of these statistical uncertainties with regard to concatenation away.

3.3.3. Tool evaluation

The above tools for the user-visible SLS metric need to be evaluated. The evaluation template for these tools can be seen in Annex 1. A number of tools (Clink, Chariot, Multicast Beacon, Netperf, Pathchar, Pathload, Pathrate, QoSmart, QoSplot, RIPE NCC TTM, RUDE/CRUDE, **Service Assurance Agent** and Taksometro) have been tested (see Annex 2, and in more detail see [58] and [59]).

3.3.4. Conclusions

The above summary shows that for only one out of the five user-visible SLS metrics (available bandwidth) no acceptable tool exists (though Pathload [76] is being worked on). In this context, members will continue experimenting with some of the tools mentioned to better understand their operational environment and their advantages and disadvantages.

Work needs to be done in order to stimulate the development of methods, implementation and integration of tools (especially for one-way delay measurement methodology, available bandwidth and SLS metric based video quality).

TF-NGN is taking part in the IPPM IETF Working Group ([68] and [60]) on determining methods to measure the needed metrics.

3.4. A specification of the measurement infrastructure

To do the above SLS measurements in a structured way and to be able to integrate other measurements (e.g. related to the operations of a network or testing new IP features), a measurement infrastructure is proposed. An overview of present-day measurement infrastructures is made available by Caida [61].

This measurement infrastructure needs to support the following characteristics:

- Support testing facilities for end-users and network managers; the measurements need to be tailored to the correct metric, accuracy and granularity for the specific user group. Precise clock synchronisation in scope of the whole measurement infrastructure is required for proper OWD measurements. Clock resolution in points of measurement should provide enough granularity to assign unique timestamp to each packet at high speed rates.
- Provide a flexible platform. As the network measurements will be under continuous development, a flexible platform has to be defined to support this *moving* and *evolving* environment (like network or service changes). A facility that is based on downloading scripts could support this flexibility (such as Chariot and Taksometro).
- Support of active and passive measurement. Both active and passive measurements are important in auditing the SLS.
- Access to the infrastructure in a secure way. As soon as a large group of different users is able to start and access measurements, a secure system must be implemented; as an example, the system could be based on SSL [62] or AAA-servers.
- A comprehensive method for discovering the measurement points in a multi-management domain. A possible methodology that could be used is Jini [72]. JINI is based on the Java Programming Language and provides a system where services (e.g. the measurement points) can be registered

at, while the users of the system (the clients that will eventually digest the data from the measuring points) can discover and use them.

- Measurements over multiple network management domains. The measurement infrastructure must support measurements between end-systems, between end-systems and edge systems and between edge systems. An edge system is in principle a system that is located *between* two different network management domains, but it can also be a system somewhere inside the network management domain, such a router or an Multi-point Conferencing Unit [MCU].
- The methodology must support the possibility to allocate possible faults or problems to a single domain in the path. It must also allow for threshold settings.
- Trustable and exchangeable measurement results. The infrastructure has to work in a multiple (network management) domain environment, so measurements done by others should be trustable ([60]) and exchangeable (which can be achieved by having a standard database, perhaps using XML or the RRDtool [63]).
- Independent of Operating System. Because the end-system or edge system can be any system, it is necessary that the tool is OS independent.
- Resource management. When running an active or passive measurement on a system, it is necessary to monitor the resources of that system, so that the actual performance is not affected by the running of the tests, gathering the information or processing the data.
- Provision of a measurement protocol. In order to do tests on many brands of equipment, a measurement protocol needs to be defined (and implemented). Some work in this direction has been done in the IPPM activity (see [64] or [65]).

3.4.1. A possible topology of the measurement infrastructure

A possible topology for SLS measurements is depicted in Fig. 3 and it consists of the following components:

- The Measurement Points (MP) could be implemented on the systems themselves (such as an end-system, a server [WWW server, file-server, MCU, etc.] or a router) or on a nearby system. Preferably they should be on the system that is part of the communication path of the application. If not possible, they can use a system close to the communication path.

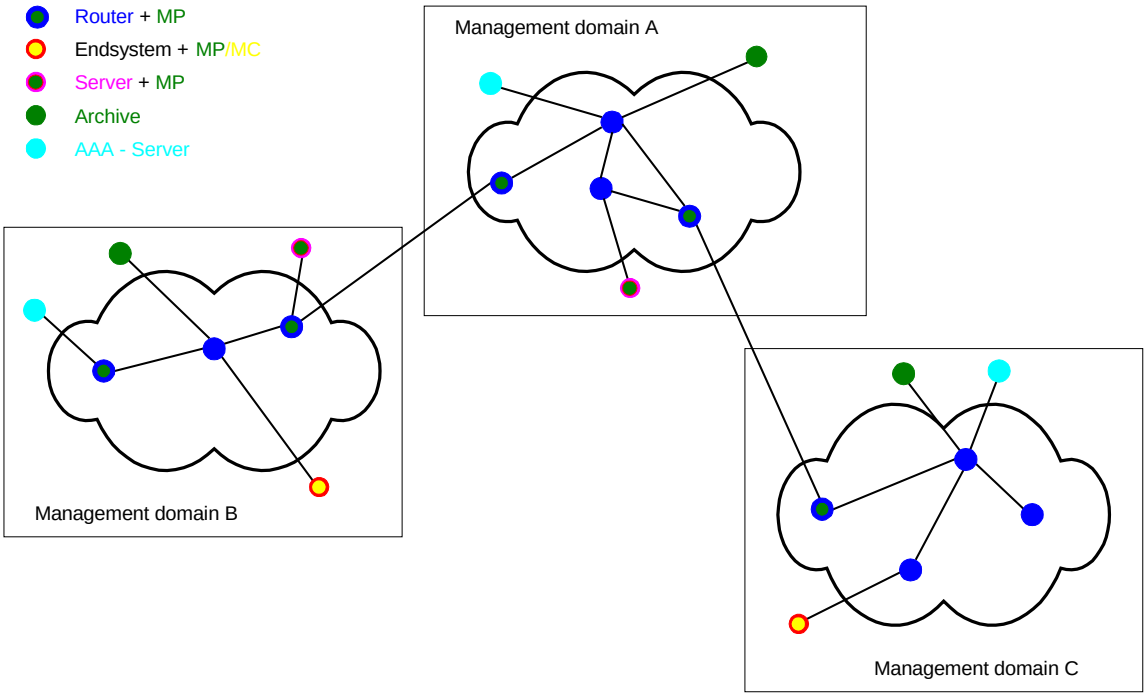

Fig. 3 Topology of the measurement infrastructure

- In principle everyone should be able make measurements between any two MP's, if there are proper access restrictions in place. This can be done by using a Measurement Client (MC).
- The MCs must be able to download scripts into the MP's for running the specific tests. The scripting should be able to perform all kinds of tests: active measurements, passive measurements, user-visible, network-manager-visible and so on.
- Archiving is needed for the results of passive and active measurements. If tests need real-time results, archiving is perhaps not very important, but anyway archiving has to be organized in a standard way (using standard protocols to input and output) so that history construction is guaranteed. Every network management domain is expected to have such an archive.
- AAA-servers should provide access to the measurement infrastructure. Every network management domain is expected to have such an AAA-server.

3.4.2. Recommendations on Measurement Points

Because a service provider does not have responsibility for the whole end-to-end path between MPs, the monitoring scope of that provider will be per domain (see also SEQUIN [82], figure 1b). The extent of measurements that can be done in a domain is thus between all peering points (in that domain) to other external domains (a customer connection is also an external domain). So in principle at every PoP where a customer connection or an external connection (to transit or peering providers) exists, an MP should be installed.

Due to the fact that the one-way delay within one domain could have a low value, tight time synchronisation between clocks may be the only way to guarantee one-way delays measurements in the order of msecs.Whether this tight synchronisation can be based on GPS or NTP is being investigated by TF-NGN.
When defining an end-system also as being a domain (of the end-user, and thus with its own MP), a full end-to-end performance measurement can be done between end-systems. Stepwise tests can be performed using the MP's add domain peering points. This provides a way to allocate possible problems (not only by end-user but also by network manager).

3.4.3. Performance Response Team

A future activity for TF-NGN is the Performance Response Team (PRT). The organisation is equivalent to the CERT (Computer Emergency Response Team [88]) set up, which is a well known structure in network land and maps very well to the European NREN structure and their connected institutes.

The PRT should help people to find the proper channels to solve performance problems. So a network of contacts is needed between applications people, network people, theoretical people and others (again this is the same environment as the CERT is working in). The PRT must have access to this expertise and must provide a structure how to access this expertise.
If needed the PRT could make tools, but normally it will leave this to specialists (again very comparable to CERT setup), but it will help to distribute the knowledge ortools. The PRT does not circumvent the responsibilities and opportunities of local computing centres or NREN centres! It is merely a network to find the correct people to solve performance problem, and may assist them in doing so.

3.4.4. Conclusions

As with all issues related to QoS and SLS a lot of work has to be done on the specification of a measurement infrastructure.

All the above characteristics have been crystallised in more detail so the time has come to try to implement some of these ideas in a concrete measurement infrastructure. Several NRENs in Europe (HEAnet, CESnet as well as GEANT) are thinking about these ideas. It is the aim of the TF-NGN to coordinate this effort so that a workable measurement infrastructure (according to as much characteristics as described in section 3.4) can be realised.

The PRT organisation could provide a proper structure to help people with performance issues.

4. REFERENCES

[1] Traffic Flow Measurement: Architecture (RFC 2722), N. Brownlee, C. Mills, G. Ruth, October 1999

[2] Cisco IOS NetFlow site, Cisco Systems, http://www.cisco.com/go/netflow/

[3] Report on Results of the Quantum Test Programme, T. Ferrari, S. Leinen, J. Novak, S. Nybroe. H. Prigent, V. Reijs, R. Sabatino, R. Stoy, QUANTUM deliverable D6.2, June 2000, available from http://www.dante.net/quantum/qtp/

[4] Processing Accounting Data into Workloads, A. Cockroft, October 1999, Sun BluePrints™ OnLine, http://www.sun.com/blueprints/1099/workload.pdf

[5] Requirements for IP Flow Export, J. Quittek, T. Zseby, G. Carle, S. Zander, July 2001 (work in progress), Internet-Draft http://www.ietf.org/internet-drafts/draft-quittek-ipfx-req-01.txt

[6] Light-weight Flow Accounting Protocol Specification Version 5.0, P. Calato, M. MacFaden, July 2001 (work in progress), Internet-Draft http://www.ietf.org/internet-drafts/draft-riverstone-lfap-00.txt

[7] Light-weight Flow Accounting Protocol Data Specification Version 5.0, P. Calato, M. MacFaden, July 2001 (work in progress), Internet-Draft http://www.ietf.org/internet-drafts/draft-riverstone-lfap-data-00.txt

[8] Common Reliable Accounting for Network Element (CRANE), K. Zhang, E. Elkin, June 2001 (work in progress), Internet-Draft http://www.ietf.org/internet-drafts/draft-kzhang-crane-protocol-00.txt

[9] sFlow: Method for Monitoring Traffic in Switched and Routed Networks, P. Phaal, S. Panchen, N. McKee, June 2001 (work in progress), Internet-Draft http://www.ietf.org/internet-drafts/draft-phaal-sflow-montraffic-01.txt

[10] Management Information Base for the Differentiated Services Architecture, F. Baker, K. Chan, A. Smith, August 2001 (work in progress), Internet-Draft http://www.ietf.org/internet-drafts/draft-ietf-diffserv-mib-11.txt

[11] JANET Traffic Accounting Site, K. Hoadley, 1998-2001 (work in progress), http://bill.ja.net/

[12] FlowScan: A Network Traffic Flow Reporting and Visualization Tool, D. Plonka, In: LISA 2000 Proceedings. Also available from http://net.doit.wisc.edu/~plonka/lisa/FlowScan/

[13] Fluxoscope – A System for Flow-based Accounting, S. Leinen, March 2000, http://www.tik.ee.ethz.ch/~cati/deliv/CATI-SWI-IM-P-000-0.4.pdf

[14] Flow-based Traffic Analysis at SWITCH, S. Leinen, April 2001, in: PAM2001 Proceedings, http://www.ripe.net/pam2001/Papers/poster_02.ps.gz. A poster that was presented at the workshop is available under ftp://ftp.switch.ch/software/sources/network/fluxoscope/papers/pam2001-poster.ps.gz

[15] Tackling Network DoS on Transit Networks, D. Harmelin, March 2001, Dante In Print (DiP) 42, available from http://www.dante.net/pubs/dip/index.html

[16] Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2 (RFC 1657), S. Willis, J. Burruss, J. Chu, July 1994

[17] Management Information Base for Network Management of TCP/IP/based internets: MIB-II (RFC 1213), K. McCloghrie, M. Rose (Eds.), March 1991

[18] Tackling Networks DoS on Transit Networks, David Harmelin, http://www.dante.net/pubs/dip/42/42.html

[19] http://www.itu.int/itudoc/itu-t/approved/h/h323.html

[20] QoS monitoring and SLS auditing, V. Reijs, http://www.heanet.ie/Heanet/projects/nat_infrastruct/qosmonitoringtf-ngn.html, 12 July 2001.

[21] Internet2 QoS Working Group charter, http://www.internet2.edu/qos/wg/apps/appsQoS-charter.html, 16 March 2001

[22] Modeling TCP throughput: A simple model and its empirical validation, J. Padhye, http://www.acm.org/sigcomm/sigcomm98/tp/abs_25.html, Sigcomm 98

[23] Calculating TCP goodput, Reijs V., http://www.heanet.ie/Heanet/projects/nat_infrastruct/tcpcalculations.htm, 5 Febr. 2001

[24] VoIP speech quality as a function of codec, manufacturer, jitter and packet loss, GTE Laboratories, May 1999

[25] VoIP speech quality as a function of delay, jitter and packet loss, GTE Laboratories, April 2000

[26] Trade-off value of VoIP speech quality vs. a messaging feature, GTE Laboratories, April 2000

[27] Methods for subjective determination of transmission quality, ITU-T recommendation P.800

[28] The E-model, a computational model for use in transmission planning, ITU-T recommendation G.107

[29] Digital transport of one-way video telephony signals - Parameters for objective performance assessment, http://www.its.bldrdoc.gov/n3/video/tutorial.htm, ANSI T1.801.03, 1996

[30] ITU-T P.910, Subjective video quality assessment methods for multimedia applications, http://www.itu.int/itudoc/itu-t/rec/p/p_910.html, Sept. 1999.

[31] Impact of IP performance on customer perceived quality, Telcordia, October 2000

[32] Joint source/FEC rate selection for optimal MPEG-2 video delivery, Frossard and Verscheure, IEEE International Conference on Multimedia & Expo 2000

[33] Designing fast-action games for the Internet, Y-Shen Ng, http://www.gamasutra.com/features/19970905/ng_01.htm, May 18, 2001

[34] Lag over 150 milliseconds is unacceptable, G. Armitage, May 25, 2001

[35] A One-way Packet Loss metric for IPPM, G. Almes, http://www.advanced.org/IPPM/docs/rfc2680.txt, Sept. 1999

[36] Methods for subjective determination of transmission quality, ITU-T recommendation P.800

[37] One-way loss pattern sample metrics, R. Koodli, R. Ravikanth, http://www.ietf.org/rfc/rfc3357.txt, August 2002.

[38] IP Packet Delay Variation metric for IPPM, P. Chimento, http://www.ietf.org/internet-drafts/draft-ietf-ippm-ipdv-09.txt, April 2002

[39] A One-way Delay metric for IPPM, G. Almes, http://www.ietf.org/rfc/rfc2679.txt?number=2679, Sept. 1999

[40] Quality of Service definition, Campanella, M., Chivalier, P., Sevasti, A., Simar, N., http://www.dante.net/tf-ngn/SEQ-D2.1.pdf, SEQUIN, IST-1999-20841, March 2001

[41] The SEQUIN project, http://www.dante.net/sequin/

[42] empty

[43] Perceived quantitative quality of applications, V. Reijs, http://www.heanet.ie/Heanet/projects/nat_infrastruct/perceived.html, July 12, 2001

[44] NIST Net home page, http://www.antd.nist.gov/itg/nistnet/, 14 March 2001

[45] Tools for measuring the SLS metric, V. Reijs, http://www.heanet.ie/Heanet/projects/nat_infrastruct/nettools.html, July 12, 2002.

[46] Measuring bottleneck link speed in packet-switched networks, R. L. Carter and M. E. Crovella, http://www.cs.bu.edu/techreports/1996-006-measuring-bottleneck-link.ps.Z, March 15, 1996

[47] A framework for defining empirical Bulk Transfer Capacity metrics, M. Mathis, M. Allman, ftp://ftp.rfc-editor.org/in-notes/rfc3148.txt, July 2001.

[48] A Bulk Transfer Capacity methodology for cooperating hosts, M. Allman, http://citeseer.nj.nec.com/allman01measuring.html , February 2001

[49] Internet delay measurements using test traffic design note, H. Uijtendaal, http://www.ripe.net/ripe/docs/designnote.html , May 30, 1997, RIPE-158

[50] Chariot, http://www.netiq.com/products/chr/default.asp

[51] RTP: A transport protocol for real-time applications, ftp://ftp.ripe.net/rfc/rfc1889.txt, January 1996

[52] Mbone conferencing applications, http://www-mice.cs.ucl.ac.uk/multimedia/software/, 28 June 2001

[53] Low-cost precise QoS measurement tool, Ubik, S., Smotlach, V., Saaristo S., Laine J., http://www.cesnet.cz/doc/techzpravy/2001/07/qosplot.pdf, CESNET technical report number 7/2001

[54] User guide Chariot, April 2001.

[55] What you need to know before you deploy VoIP, Hamilton S., Bruno C., http://download.netiq.com/CMS/NetIQ_What_You_Need_to_Know_for_VoIP.pdf, April 2001.

[56] Modeling the effects of burst packet loss and recency on subjective voice quality, Clark A.D., http://www.fokus.gmd.de/events/iptel2001/pg/final_program/21.pdf

[57] ITU-T P.910, Subjective video quality assessment methods for multimedia applications, http://www.itu.int/itudoc/itu-t/rec/p/p_910.html, Sept. 1999.

[58] Network measurement tools test, M. Przybylski, Sz. Trocha, Poznan, http://qos.man.poznan.pl/files/measurement_full.pdf, Supercomputing and Networking Center, 2001

[59] Network measurement tools test Part II, M. Przybylski, Sz. Trocha, http://qos.man.poznan.pl/files/measurement2.pdf, Poznan, Supercomputing and Networking Center, 2001

[60] IP Performance Metrics, http://www.ietf.org/html.charters/ippm-charter.html, 12 July 2001

[61] Internet measurement infrastructure, M. Murray, http://www.caida.org/analysis/performance/measinfra/, 24 May 2001

[62] Nettest: Secure Network Testing and Monitoring, http://www-itg.lbl.gov/nettest/

[63] RRDtool, T. Oetiker, http://www.lk.etc.tu-bs.de/lug/linuxtage/blt_2/cd_online/vortraege/rrdtool/website/index.html

[64] IPMP, http://amp.nlanr.net/AMP/IPMP/, 5 April 1999

[65] One-way active measurement protocol requirements, Shalunov S., Teitelbaum B., http://www.ietf.org/internet-drafts/draft-ietf-ippm-owdp-reqs-03.txt, July 2001.

[66] ViDeNet Scout, Ott D., http://scout.video.unc.edu/, 6 March 2001

[67] IP Quality of Service, M. Peuhkuri, http://www.tct.hut.fi/u/puhuri/htyo/Tik-110.551/iwork/iwork.html, 21 May 1999

[68] Framework for IP Performance Metrics, V. Paxson, http://www.ietf.org/rfc/rfc2330.txt, May 1998

[69] Video quality research, http://www.its.bldrdoc.gov/n3/video/Default.htm, 26 October 1999

[70] On the impart of policing and rate quarantees in Diff-Serv networks: A video streaming application perspective, Wolf S.,Guerin R., Pinson M., Ashmawi W., http://www.ee.upenn.edu/~guerin/publications/sigcomm2001_extended.pdf , SIGCOMM 2001

[71] dummynet, http://www.iet.unipi.it/~luigi/ip_dummynet/

[72] Jini, http://www.sun.com/jini/whitepapers

[73] Introduction to the Surveyor project, http://www.advanced.org/surveyor/, 16 July 1999

[74] Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi, Teunis Ott, The Macroscopic behavoir of the TCP congestion avoidance algorithm, http://www.psc.edu/networking/papers/model_ccr97.ps, ACM SIGCOMM, vol. 27, number 3, July 1997.

[75] Wenyu Jiang, Henning Schulzrinne, QoS measurement of Internet real-time multimedia services, http://www.cs.columbia.edu/~hgs/papers/Jian9912_QoS.pdf, Dec. 1999

[76] Manish Jain, Constantinos Dovrolis, End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput, http://www.acm.org/sigcomm/sigcomm2002/papers/e2ebw.html

[77] Sven Ubik, End-to-end performance cookbook, http://www.cesnet.cz/english/project/qosip, (to be finilized)

[78] Vladimir Smotlacha, One-way delay measurement using NTP synchronization, http://www.cesnet.cz/english/project/qosip/owd-meas/index.html, August 26th, 2002

[79] The QoSmart tool, http://www.ipanematech.com/products.html, Ipanematech Technologies.

[80] The Video Quality Expert Group, http://www.VQEG.org

[81] Dimitrios Miras, Network QoS of advanced Internet applications: A survey", http://www.internet2.edu/qos/wg/apps/fellowship/Docs/Internet2AppsQoSNeeds.html, July 2002

[82] Athanassios Liakopoulos, Monitoring and verifying Premium IP SLAs, SEQUIN, D2.1, IST-1999-20841, April 30, 2002.

[83] Victor Reijs, One-Way delay test results of NetIQ: Chariot, http://www.heanet.ie/Heanet/projects/nat_infrastruct/oneway.html, May 15th, 2002.

[84] Vern Paxson, On calibrating measurements of packet transit times, http://citeseer.nj.nec.com/paxson98calibrating.html, LBNL-41535, 1998

[85] Colin Whittacker, Testresults of QoSmart, http://www.redbrick.dcu.ie/~grimnar/HEAnet/owd.html, September 2002.

[86] VQmon Technology, Telchemy, http://www.telchemy.com/products.html, Nov. 2001

[87] Testing of traffic measurements tools, http://www.dante.net/tf-ngn/D9.4v2.pdf, Simon Leinen, Michal Prybylski, Victor Reijs, Szymon Trocha, IST-2000-26417, Sept. 2001

[88]CERT, http://www.cert.org/

[89] *FloMA Web Page*

S. Leinen, 1999-2002 (work in progress).

[90] D. Plonka, 2000-2002 (work in progress), http://ipfix.doit.wisc.edu/

[91] Requirements for IP Flow Information Export
J. Quittek, T. Zseby, B. Claise, S. Zander, G. Carle, K. C. Norseth, August 2002 (work in progress), Internet-Draft `draft-ietf-ipfix-reqs-05`

[92] Cisco Systems NetFlow Services Export Version 9
B. Claise, June 2002 (work in progress), Internet-Draft `draft-bclaise-netflow-9-00`

[93] Evaluation Of NetFlow Version 9 Against IPFIX Requirements
B. Claise, September 2002 (work in progress), Internet-Draft `draft-claise-ipfix-eval-netflow-00.txt`

[94] Light-weight Flow Accounting Protocol Specification Version 5.0
P. Calato, M. MacFaden, July 2002 (work in progress), Internet-Draft `draft-riverstonelfap-01.txt`
[95] Light-weight Flow Accounting Protocol Data Definition Specification Version 5.0
P. Calato, M. MacFaden, July 2002 (work in progress), Internet-Draft `draft-riverstone-lfap-data-01`
[96] Evaluation Of Protocol LFAP Against IPFIX Requirements
P. Calato, September 2002 (work in progress), Internet-Draft `draft-calato-ipfix-lfap-eval-00.txt`
[97] XACCT's Common Reliable Accounting for Network Element (CRANE) Protocol Specification Version 1.0
K. Zhang, E. Elkin, August 2002 (work in progress), Internet-Draft `draft-kzhang-crane-protocol-05.txt`
[98] Evaluation of the CRANE Protocol Against IPFIX Requirements
K. Zhang et al., September 2002 (work in progress), Internet-Draft `draft-kzhang-ipfix-eval-crane-00.txt`
[99]Reliable Streaming Internet Protocol Detail Records
J. Meyer, August 2002 (work in progress), Internet-Draft `draft-meyer-ipdr-streaming-00.txt`
[100]Evaluation Of Streaming IPDR Against IPFIX Requirements
J. Meyer, September 2002 (work in progress), Internet-Draft `draft-meyer-ipfix-ipdr-eval-00.txt`
[101] Evaluation of Diameter Protocol against IPFIX Requirements
S. Zander, September 2002 (work in progress), `draft-zander-ipfix-diameter-eval-00.txt`

[102] InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks
P. Phaal, S. Panchen, N. McKee, September 2001 (RFC 3176)

[103] A Framework for Passive Packet Measurement
N. Duffield (Ed.), September 2002 (work in progress), Internet-Draft `draft-ietf-psamp-framework-00.txt`

[104] Management Information Base for the Differentiated Services Architecture (RFC 3289)
F. Baker, K. Chan, A. Smith, May 2002

5. ACRONYMS

AAA	Authentication, Authorization, and Accounting
avail-bw	Available Bandwidth
ACK	ACKnowledgement
AS	Autonomous System
ASIC	Application Specific Integrated Circuits
BER	Bit Error Rate
BGP	Border Gateway Protocol
BTC	Bulk Transfer Capacity
CERT	Computer Emergency Response Team
CLI	Command-Line Interface
DSCP	Differentiated Services Code Point
FEC	Forward Error Correction
GPS	Global Positioning System
HTTP	Hyper Text Transfer Protocol
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IGP	Internet Gateway Protocol
IOS	Internetwork Operating System
IP	Internet Protocol
Ipdv	Ip Packet Delay Variation
IS-IS	Intermediate System-to-Intermediate System
ISP	Internet Service Provider
IXP	Internet Exchange Point
LAN	Local Area Network
LBE	Less than Best Effort
LFAP	Lightweight Flow Admission Protocol
MAC	Media Access Control
MC	Measurement Client
MCU	Multi-point Conferencing Unit
MIB	Management Information Base
MOS	Mean Opinion Score
MP	Measurement Point
MRTG	Multi Router Traffic Grapher
NNTP	Network News Transfer Protocol
NREN	National Research and Educational Network
NTP	Network Time Protocol
OS	Operating System
OSPF	Open Shortest Path First
OWD	One-Way Delay
OWLP	One-Way Loss Pattern
OWPL	One-Way Packet Loss
PFC	Policy Feature Card
PLC	Packet Loss Concealment
PPS	Pulse Per Second
PRT	Performance Response Team
PXF	Parallel eXpress Forwarding
QoS	Quality Of Service
RRD	Round Robin Database
RTFM	Real Time Flow Measurement
RTP	Real-Time Protocol
RTT	Round Trip Time

SLA	Service Level Agreement
SLS	Service Level Specification
SLoPS	Self-Loading Periodic Streams
SNMP	Simple Network Management Protocol
SOHO	Small Office Home Office
TCP	Transport Control Protocol
TF-NGN	Task Force – Next Generation Networking
TF-TANT	Task Force – Testing of Advanced Networking Technologies
TOS	Type Of Service
TTM	Test Traffic Measurements
UDP	User Datagram Protocol
V^2oIP	Voice & Video Over IP
VoIP	Voice Over IP
VQM	Video Quality Measurement

6. ANNEX I

This overview presents issues important for evaluating measurements tools, based on the initial list of Constantinos Dovrolis, from the Internet2 IPPM working group.

1. What is the name and version of the tool?
2. What does the tool attempt to measure (capacity, available bandwidth, BTC, RTT, OWD, is raw data available for manipulation, etc.)?
3. What protocols can be supported (such IPv4, IPv6, etc.)?
4. On what measurement principle is the measurement based? E.g. with bandwidth tools: Is it based on the dispersion of packet pairs/trains, on the one-way delay variations of variable-sized probing packets, on actual TCP transfers (e.g., BTC measurements) or on emulated TCP flows?
   - Is it using active or passive measurements?
   - Does it use information from the routers (i.e., SNMP-based tools like MRTG), or is it based on end-point measurements?
   - In case of active measurements: Does the tool require access (and software) at both ends of the path?
   - Does it measure estimates for each circuit in the path, or at an end-to-end basis?
   - What minimum and maximum (logical/physical) transmission speeds can be handled?
   - How does the tool handle parallel-circuits in the path (such as channelized ethernet, OC-3**c**)?
   - Can the reverse path affect the measurements?
   - Does it require ICMP replies from systems in the path?
   - For some tools: Is it robust to cross traffic? In other words, can it make accurate measurements even when the path is heavily loaded?
   - Can the tool utilize QoS bits (like TOS/DSCP)?
   - Is the tool loading the path with the traffic over the duration of the measurements? And at what level? What is the duration of the measurement?
   - What is the influence of the run-time parameters on the overall measurement output (configuration parameters such as: packet size, number of probes, etc.)?
   - Does it take user-level or kernel-level timestamps?
     Time stamping accuracy and resolution are major issues in measuring high-speed paths. Resolution is not a problem these days (1 μsec is typical). Accuracy, though, can be a major problem in both kernel and user-level timestamps (more in the later), and especially when the measuring host is not totally idle.
   - Does it require access at the raw-IP socket (i.e., need special user privileges)?
5. What is the granularity of the tool?
   Network managers need very granular measurement (like μsec for OWD), but users need only msec granularity.
6. What is its accuracy?
   Obviously, the background can influence the results in a statistical way. This means that there is no guarantee that they will give you the right value every time you run them. The possibility of error is always there in every area of measurements and experimentation. However, do they give the right estimate "most of the time"?
7. Does the tool calculate (correct) experimental errors?
8. Do hidden Layer-2 switches and/or other store-and-forward devices affect the measurements?
9. On which operating systems does the tool run?
10. Are there known bugs and limitations?
11. Is the tool actively maintained/developed?

7. ANNEX 2

We present an overview of a number of measurement tools. It has been based on the issues described in Annex 1.

Clink

Tool: http://rocky.wellesley.edu/downey/clink/

Doc: http://rocky.wellesley.edu/downey/clink/clink.doc

Version tested: 1.0

Clink is designed for measuring capacity for each link along the given path. It does active measurement tests, based on RTT time variations for variable-sized probing packets. This is a standalone application, using raw sockets – thus requiring root privileges. It works on many platforms, including FreeBSD, Linux and Solaris.

Clink ensures an accurate measurement for slow and over-provisioned networks (speed <10Mbit/s, load <10-20%). For heavy loaded links, this tool does not perform well – the reported bandwidth is approximately 72% of the real bandwidth for empty network and approximately 43% of the real bandwidth for the network with 80% load. Similar situations occur on fast links (speed >100Mbit/s) where the tool showed less than 50% of the bandwidth, even for empty networks.

Clink does not produce significant network load. The presence of the Layer 2 devices and run-time parameters setting may strongly affect measurements.

Chariot

Tool: http://www.netiq.com/products/chr/deafult.asp
Documents:
http://www.netiq.com/Downloads/Products/Chariot/Documentation/NetIQ_CHR_User_Guide.pdf
Version tested: 4.01

Chariot is designed to simulate applications over the IPv4 Internet. It uses scripts that are downloaded to the measurement point, between which the tests will be performed. The script simulates an application, which can be for instance an ftp session, a client server communication, a VoIP call, etc. It uses active measurements and can measure several parameters; such as RTT, one-way delay (which does not work as expected [83]), ipdv, packet loss, packet duplication, packet reordering and it can determine the MOS experienced in VoIP. The tool can utilise multicast and TOS bits during testing. The measurement points can be installed anywhere in the path as long as it is one of the operating systems, approximately fifteen, supported by Chariot.

The tool simulates an application, so in principle it is possible to simulate streams of any desired speed. It is therefore also possible to test network or system behaviour under load. It uses user-level timestamping (no special user privileges are needed).

Multicast Beacon

Tool & docs: http://dast.nlanr.net/projects/beacon
http://noc.man.poznan.pl (applications)
Version tested: 0.63

Multicast Beacon is an application for monitoring multicast traffic characteristics. These characteristics are:

- Packet loss
- One way delay (to have accurate information the clock on the sender **and** receiver must be closely synchronised)
- delay variation

- duplicate
- order

The application consists of two parts: server and clients. The role of the server is to collect information received from clients and to present them by means of a stand-alone GUI tool or web interface. The second approach is very helpful for a large number of users interested in the results. The clients exchange packets using the multicast technology and this way they can easily determine values of traffic. The clients send such information to the server.
To have a matrix of states of multicast traffic between some locations, the beacon administrator can locate the clients there and in the central place deploy the server, which would be the interface for a user.
The original version of the Multicast Beacon was developed in the National Laboratory for Applied Network Research (NLANR) in USA. Poznan Supercomputing & Networking Center (PSNC) in Poland introduced some new additional features (like statistics storing, message module, mtrace module).
The application is put on the open source initiative, is developed in Java language, is operating system independent and ipv6 enabled.

Netperf

Tool & doc: http://www.netperf.org/netperf/NetperfPage.html

Version tested: 2.1

Netperf is designed to measure the network performance between endpoints. It does active measurement tests by trying to send packets at maximum possible network speed. The tool allows for measuring the BTC (TCP goodput) and capacity (using UDP).

This is a client-server application and requires access to both ends of the measured path. While a non-privileged user may run the client, the server must be run by root. Netperf is distributed in source version and should work on Linux, FreeBSD and other platforms.

Tool characteristics

The Netperf (when using UDP test) is a very stable tool – the measurement range variation was less than 1% of the link capacity (for 10Mbit/s link). The capacity estimate usually showed about 85%-90% of the original link speed, regardless of the network load. The main problem with the Netperf measurement is the produced network load. Netperf sends packets at the maximum possible speed, thus saturating the link and possibly, starving other flows. Layer 2 devices do not affect this tool, but inappropriate run-time parameters setting may affect the measurement quality.

This tool allows for an accurate measurement in a very short time (less than 3 min).

Pathchar

Tool: ftp://ftp.ee.lbl.gov/pathchar/

Doc: www.cs.colby.edu/~downey/pathchar

Version tested: 2.0.30

Pathchar is designed for measuring capacity (and other parameters, which were not tested) for each link along the given path. It does an active measurement, based on RTT time variations for variable-sized probing packets. This is a standalone application, using raw sockets – thus requiring root privileges. It works on many platforms, including FreeBSD, Linux, OSF and Solaris.

Pathchar performs very well for slow, over-provisioned (speed < 10Mbit/s, network load < 10-20%) networks. With higher network load the tool repetitiveness becomes unacceptable – the subsequent measurements show totally different numbers. For fast networks (with the speed > 100Mbit/s) reported bandwidth was much less than the real bandwidth – approx. 40% less. Layer 2 devices or inappropriate setting of run time parameters may affect the measurements.

Pathload

Tool: http://www.cis.udel.edu/~dovrolis/bwmeter.html
Version tested: 1.0.2

Pathload is designed for measuring end-to-end available bandwidth; that is the actual IP layer throughput between the two ends achievable at the time of measurement. The measurement is based on the idea that delay variation of a periodic packet stream show increasing trend, when the stream rate is larger than the available bandwidth. Pathload is a client-server software and requires the tool to be installed at both ends.

Currently, internal constants hardcoded in pathload limit the measurements to speeds up to 120 Mb/s. Testing the tool on higher speeds, by tweaking the internal constants, reported a lot of test packet losses. The test was performed on a high speed, long distance path consisting of 12 hops of either OC-48 or Gigabit Ethernet lines with round trip delay of approximately 40ms and available capacity of approximately 600 Mbs (tested by brute force streams). These losses were probably caused by lots of processing in pathload.
The authors assume that measurements up to the OC-12 range should be possible on appropriately powerful hardware.

Pathrate

Tool & doc: *www.cis.udel.edu/~dovrolis/bwmeter.html*

Version tested: 2.1.1

Pathrate is designed for measuring path capacity – understood as a maximum IP layer throughput that a flow can get. The measurement is based on a dispersion of packet trains/pairs. This is a client-server application, so access to both endpoints is required. The tool works on FreeBSD, Linux, DEC-Unix, Solaris and HP-Unix platforms.

Pathrate is probably one of the best tools for capacity measurement. It performs well for slow and fast networks, giving an accurate bandwidth estimate, which usually differs no more than 5% from the real path capacity. The network load has an influence on the measurement time only – for over-provisioned networks it takes less than a minute to get an output, while heavy loaded path requires even half an hour.

Layer 2 devices have little or no influence on the Pathrate measurements. The tool itself does not require any run-time parameters.

QoSmart

Documents: http://www.ipanematech.com/products.html

Version tested: 2.5 release 16 (ip|engine)

QoSmart form ip|anema consists of three components.

- ip|boss which is the management software suite.
- ip|agent which is the measurement software running on the endpoints.
- ip|engines which are the actual hardware devices at the endpoints.

QoSmart is a passive measurement system, which can measure OWD, throughput, delay variation and packet loss between any two endpoints.

An ip|engine is installed at the egress point of the end point network, either in line in the ethernet uplink or on a mirror port on the switch (available speeds 10/100/1000 Mbit/s). The ip|agent software then monitors packets leaving the network and reports details of these flows to the management

software. By having an ip|engine at both ends of a flow it is possible for the QoSmart system to report OWD, throughput for the flow, jitter and packet loss, but only for the portion of the link between the two ip|engines.

ip|engines have fixed capacity in terms of traffic volumes that can be monitored. Exceeding this will overload the unit and prevent measurements from taking place.

Time synchronisation is via local GPS or ITP between ip|engines. ITP is an NTP clone.

If the ip|engine is installed inline then it is possible to take advantage of QoS management features of the platform for manipulating TOS bits and restricting flows and ensuring that QoS levels are maintained. None of the QoS features were tested in the trial.

QoSplot

Tool: S. Ubik (QoSplot), http://www.cesnet.cz/english/project/qosip
Version tested: 0.45

QoSplot takes as input the log files created by CRUDE and outputs the data and command files for gnuplot, which can then be used to plot graphs depicting all primary network QoS characteristics - loss rate, throughput, delay, delay variation and distribution of delay and delay variation. Characteristics are computed and plotted with specified time granularity. Precision of one-way delay measurement of course depends on time synchronisation on sending and receiving PCs. The PCs run Linux with nanokernel. We analysed and measured sources of instability in such environment and concluded that one-time delay measurement with the precision of less than 10 microseconds is easily achievable.

There are plans to add central configuration management for multiple-point measurements, support for IPv6, multicast and data aggregation for long-term high-speed measurements.

RIPE NCC TTM
Tool and docs: http://www.ripe.net/test-traffic

It actively measures in an IPv4 environment the end-to-end values of OWD, OWPL (using RFC 2679 and RFC 2680) and all quantities that can be derived from that (ipdv, RTT and so on). Results are available as both raw data and plots ready to be used by operators. The analysis tools run on Windows 95/NT and on most Unices including Linux and Solaris.

The tool uses dedicated measurement hardware probes at each end. It runs at the kernel level, is GPS driven, and has an overall accuracy of the order of 10 us. Traceroute is used to determine the path that the packet will take. RIPE NCC TTM will survive ICMP non-replies though. The amount of traffic (measurements run continuously) is small compared to the capacity of a typical link.
The tool is actively supported.

RUDE/CRUDE

Tool: J.Laine, S. Saaristo, R. Prior (RUDE/CRUDE), http://rude.sourceforge.net/
Version tested: 0.61

RUDE, a real-time UDP data emitter, produces a set of streams of packets of specified size and rate which are sent to specified destinations. Packet size and rate can be changed at any time. Streams can also be generated using a trace file specifying the size of each packet and the following inter-packet gap. A tool is available that creates the trace file from tcpdump output.

CRUDE, a collector for RUDE, stores a short snapshot about each packet received from RUDE. Both RUDE and CRUDE have a low and stable overhead, which allows generation and reception of high-speed streams. RUDE can dispatch packets with a steady precision of 1 microsecond (provided that the clock on the sending PC is stable enough).

Service Assurance Agent (SAA) and Internetwork Performance Monitor

Documentation**:** http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/ipmcw2k/ipm20/ipmug20
Version: v2.3 (IPM)

SAA is the new name for Cisco's Response Time Responder (RTR). SAA runs on Cisco router platforms as part of the IOS and allows the measurement of network metrics between RTR probes and RTR responders. An RTR probe initiates tests and RTR Responders reply to test packets in an appropriate way. This allows measurement of packet loss, RTT and delay variation on network links.

SAA supports IPv4 and SNA and also allows application-level testing of DNS, HTTP and DHCP response times.

IPM is a tool, which enables Cisco Service Assurance Agent to be configured from a Microsoft Windows or Solaris host without resorting to the IOS command line interface. IPM also provides text and graphical reporting of SAA generated data. IPM uses SNMP to read and write data to RTR probe routers and requires SNMP read access to RTR Responder routers. Data is collected every 60 minutes from the probe router – also called the 'source'.

On network links, IPM and SAA define a number of operation that can be performed, including Echo, Path Echo, UDP Echo, TCP connect, 160 byte Voice, 60 byte Voice and 1024 byte Video. These default operations may be re-configured and saved as new operations allowing flexibility in the testing required. Also, where appropriate IP precedence values can be set. Unfortunately DSCP values are not currently configurable – though it may be possible to re-write them on the router itself.

Operations can take place at intervals from 10 seconds to 1 hour. Data recorded includes, as appropriate, the maximum, average maximum, average, average minimum and minimum for each value and these can be graphed, reported on a table or exported to an external database.

Thresholds can also be set and SNMP traps generated should a threshold be exceeded – allowing automated notification of NOC staff.

RTT and delay variation measurements report with a granularity of 1ms. Packet size for each operation is fixed within that operation but may be varied between operations.

As yet IPM/SAA has not been tested back-to-back against another tool to check for errors in measuring, this will hopefully be done in the next couple of months. In general the tool has given a useful 'weather map' type view of the network and its operation and has shown up link problems that were not obvious before.

The tool is actively maintained by Cisco and version 2.4 is currently available.

SmartBit

SmartBits is a traffic generator device from Spirent that can perform active measurements. It timestamps packets in hardware allowing a measurement accuracy of up to 10 nanoseconds. Clock synchronisation between the participating SmartBits can be achieved by connecting each individual device to a GPS antenna.
A large number of driving applications is supported by Spirent, thanks to which a significant amount of different network performance parameters is possible to be measured.

As an example of an application, SmartFlows, which was used in the context of evaluating the performance of a Less than Best Effort (LBE) service, can monitor the following metrics for each generated test stream:

- number and percentage of packets lost
- number and percentage of packets received out of sequence
- per-flow throughput
- one-way delay distribution
- one-way delay over time
- one-way delay per packet
- frame loss, throughput and latency of jumbo frames

A variable number of concurrent source and destination addresses can be configured for each test flow, allowing the simulation of large networks; tests can run on top of the IP, TCP, UDP and ICMP transport protocols. In more detail, the TOS byte of each test flow can be set, thus making the SmartBits suitable for Differentiated Services measurements. Moreover, flows with static or dynamic MPLS labels can be generated.

At the management side, multiple users can concurrently operate one or more SmartBits devices. The test results are exported in either chart or table format, while for each test, the duration, packet sizes and type of data flows can have variable values.

Taksometro

Tool: http://www.dante.net/nep/
Version: initial

The Taksometro tool is designed for monitoring the behaviour of the different Classes of Service (CoS) implemented on a network. The tool is enabled to monitor the following list of network technology based CoS metric which can quantify any CoS type:

- packet delay,
- packet delay variation,
- bandwidth utilisation and
- packet drop

Taksometro has a modular architecture that provides the base on top of which different kind of network devices is possible to be monitored; its modules provide the logic to the tool how a specific technology based CoS parameter can be retrieved from a specific network device.
Currently the tool is bundled with five modules to automate the monitoring of the passively measured metric: bandwidth utilisation and packet drop on Juniper and Cisco router equipment:
A module for monitoring DCU and Firewall byte counters on Juniper routers for measuring CoS capacity
A module for monitoring Modular QoS byte counters on Cisco routers for measuring CoS bandwidth utilisation
A module for monitoring byte counters by the MAC Address Accounting and Precedence Accounting feature [http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/macacct.html] on Cisco routers for measuring CoS bandwidth utilisation.
A module for monitoring Firewall packet counters on Juniper routers for counting CoS packet drops
A module for monitoring Modular QoS packet counters on Cisco routers for counting CoS packet drops
Another two modules (one for each of the mentioned network technology based CoS metric) act as command placeholders where the tool administrator can fill in with commands for instructing external programs to collect the necessary data. This gives the tool administrator the possibility of calling external programs that actually connect to the devices, retrieve the data and then hand the data over to this tool to archive and project them. This enables the tool to interface with external programs that

know how to interact with unknown (to the tool) devices. Since at the moment there is no standardised way to query delay and delay variation statistics, the tool currently only offers the placeholder modules for the user to specify the external programs which actually do the collection. The statistics are projected as graphs in the time domain, and are accessible through a web interface. For each network technology based CoS metric collected from a device there are hourly, daily, weekly and monthly graphs. The tool circumvents the problem of navigating the graphs by letting the tool administrator define views of graphs available to the web interface, instead of having a default view that might be cumbersome to navigate in some circumstances.
The tool is portable and can be installed on any system that can accommodate a Python interpreter while it needs a running instance of Cricket (http://cricket.sf.net) as a dependency.