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

Deliverable D9.4

Testing of Traffic Measurement Tools

Deliverable Type: PU-Public
Contractual Date: 30 June 2001
Actual Date: 30 September 2001
Work Package: 8
Nature of Deliverable: RE - Report

Authors:

Simon Leinen SWITCH
Michal Przybylski PSNC
Victor Reijs HEAnet
Szymon Trocha PSNC

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 a 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 provides 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 network 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 NetFlow-based on different hardware platforms and also non-flow-based mechanisms like MAC, TOS/DSCP and bucket-based accounting.

After a brief review of QoS needs for different applications, a set of SLS metrics is proposed. Those metrics are followed by a study of tools that can measure the user-visible SLS metrics. To do the above measurements in a structured way a dedicated measurement infrastructure is proposed. The infrastructure will have to support measurements over multiple domains consisting of measurement points and clients accessing the infrastructure via access servers.

Because most of the tools still need to be tested a template for evaluating them is provided. As an example a summary of such evaluation of four selected tools is presented.

2. FLOW-BASED AND OTHER ACCOUNTING MECHANISMS FOR HIGHLY AGGREGATED TRAFFIC

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 materialize in this fashion for both organizational and technical reasons. Instead, participants exchanged experiences from their respective work with flow-based accounting mechanisms at TF-TANT meetings and a mailing list. This section of the deliverable reports on recent results of the working group, starting from the scenarios that have been proposed as "test cases" for flow-based (or other) accounting mechanisms in early 1999. Other accounting mechanisms and how they can be used to solve the problems defined in the scenarios will be addressed.

2.1. Potential Applications of Flow-based 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 such as the IETF RTFM (Real-Time Flow Measurement) architecture [1], Cisco's NetFlow [2], or Cabletron's LFAP (Lightweight Flow Admission Protocol) [6].

Those scenarios included:

1) Traffic Statistics at Exchange Points

Where 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) Accounting for Volume-Based Charging

Where traffic should be accounted separately for pairs of (customer organization, 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.

3) Abuse/DoS Attack Detection

Where 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.

4) Long-Term Traffic Analysis

Where 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.

5) Detection of Routing Anomalies

Where local routing preferences are matched against observed ingress points of incoming traffic, in order to identify mismatches between one's own and other networks' routing policies.
Section 2.4 will provide a closer look at these problems and propose solutions based on flow-based and other accounting mechanisms.

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.

2.3.1. 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 specialized measurement probes, in many cases in addition to other accounting protocols. The following gives a list o products potentially able to generate NetFlow-compatible accounting records with a short description:

- **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, non-aggregated) 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.

- **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.

- **Cisco PXF-based routers (7200 NSE-1, 7600 OSR and others)**

The PXF (Parallel eXpress Forwarding) is a network processor which is the basis of several recent Cisco routers. It uses a small 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. Apparently the PXF implementation can maintain correct accounting at very high packet and flow rates. As of June 2001, sampled NetFlow has not been implemented for the PXF architecture, although this is planned.

- **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.

- **Juniper M-Series routers**

Juniper implemented NetFlow accounting in recent versions of its operating system - JUNOS. It requires the second generation of the centralized 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.

- **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.

- **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.

- **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.

- **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].

2.3.2. Flow-based Accounting Methods - IPFX Standarization Efforts

Starting in spring 2001, there has been some activity in the IETF community towards the standardization of an "IP Flow Export" protocol, to complement the existing RTFM (Real-Time Flow Measurement) standard. This is based on a "pull" model (usage data is collected from meters on demand of an analysis application), with a "push" mechanism (from routers or other packet-observing devices towards consumers of accounting information).

A BOF (Birds-of-a-Feather) meeting has been scheduled for the 51st IETF meeting in August 2001 in London to discuss the scope of this activity. A working group may be chartered after that meeting, but it is also possible that this work will be done in the framework of an existing working group, most probably RTFM.

So far, the group has produced two versions of a "requirements" Internet-Draft [5], as well as multiple drafts that document existing mechanisms in this space, including a new version of Cabletron's LFAP [6][7], InMon's sFlow [9] and CRANE [8].

2.3.3. 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.

2.3.3.1. 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 neighbors 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.

2.3.3.2. 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.

2.3.3.3. 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 labor 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.

2.4. Evaluation

This section attempts to present the evaluation of several application scenarios for flow-based traffic measurements. The scenarios presented below are in compliance with the ones listed in paragraph 2.1.

2.4.1. Traffic Statistics at Exchange Points (Scenario 1)

2.4.1.1. 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.

2.4.1.2. 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.

2.4.1.3. 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.

2.4.2. Accounting for Volume-Based Charging (Scenario 2)

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 links - *expensive links* - is counted separately for each customer organization, and the volume-based fee is computed based on these amounts of traffic. It is important to note that organizations are charged not only for the traffic they send over these links, but also for the traffic they receive [12][14].

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

2.4.2.1. 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.

2.4.2.2. 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.

2.4.2.3. 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 25, 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 realize 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.

2.4.3. Abuse/DoS Attack Detection (Scenario 3)

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 analyzed 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, router-aggregated 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.

2.4.4. Long-Term Traffic Analysis (Scenario 4)

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 optimizing 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, flow durations and flow sizes in terms of packets and bytes, presents a useful characterization of the network usage or the load characteristics at different points in the network.

3. QoS MONITORING AND SLS AUDITING

The work within the GÉANT working group TF-NGN on QoS monitoring and SLS auditing is based on its activity plan [20].
This section provides the results of a literature/web 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 and the following sections provide information on these.

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

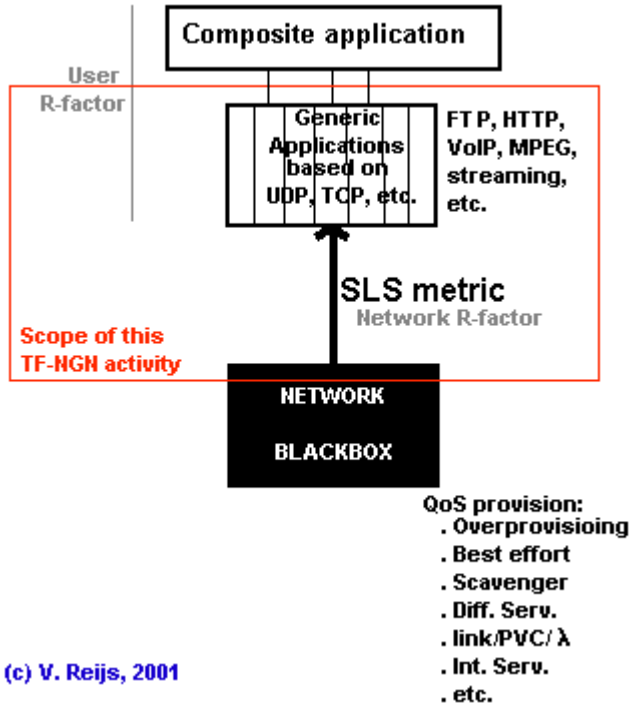

Fig. 1 TF-NGN activity scheme

The basic idea underlying this activity is that the network is considered as a blackbox. This blackbox delivers a certain level of SLS and determines the user-visible SLS. So this activity is not interested how QoS is provided by the network (i.e. by over-provisioning, differentiated/integrated services or dedicated circuits, see also [67]) and also not interested how these will be managed (that work will be left to the specific QoS feature implementations).

On the application side, this TF-NGN activity is dealing 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 of course important (like in a video conference session, lower video quality can be compensated with higher audio quality, etc.), but this area is outside the scope of the TF-NGN work (this is covered by Internet QoS workgroup [21]).

3.1. Perceived quantitative quality of generic applications and SLS metric

A lot of work is done on the subject of perceived quantitative quality of generic applications. Most applications, which seem to be built/designed around LAN, do not take into consideration the QoS

provided in wide area IP networks. This activity is gathering information which QoS metric is important for generic applications, such as:

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

It is important to distinguish between two issues when talking about 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 anymore. This may be caused by implementation issues, protocol time-outs, etc. Sometimes called: *Network R factor*.
- An application can become unusable for humans. 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 (in the below text, this type of issue will be marked with *****). Sometimes called: *User R factor*.

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 and/or effects caused by the application hardware/software implementations (such as en- or decoding delays).

3.1.1. 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] and [23]):

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

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]

3.1.2. Quality of UDP/RTP based applications

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

- Voice (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 jitter 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*
    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.

- VoIP quality
  Mean Opinion Score (MOS) ([27] and [28]) which is based on:
  - consecutive one-way packet loss,
  - one-way delay and
  - jitter.

- Synchronization between video and audio*****
  - 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]

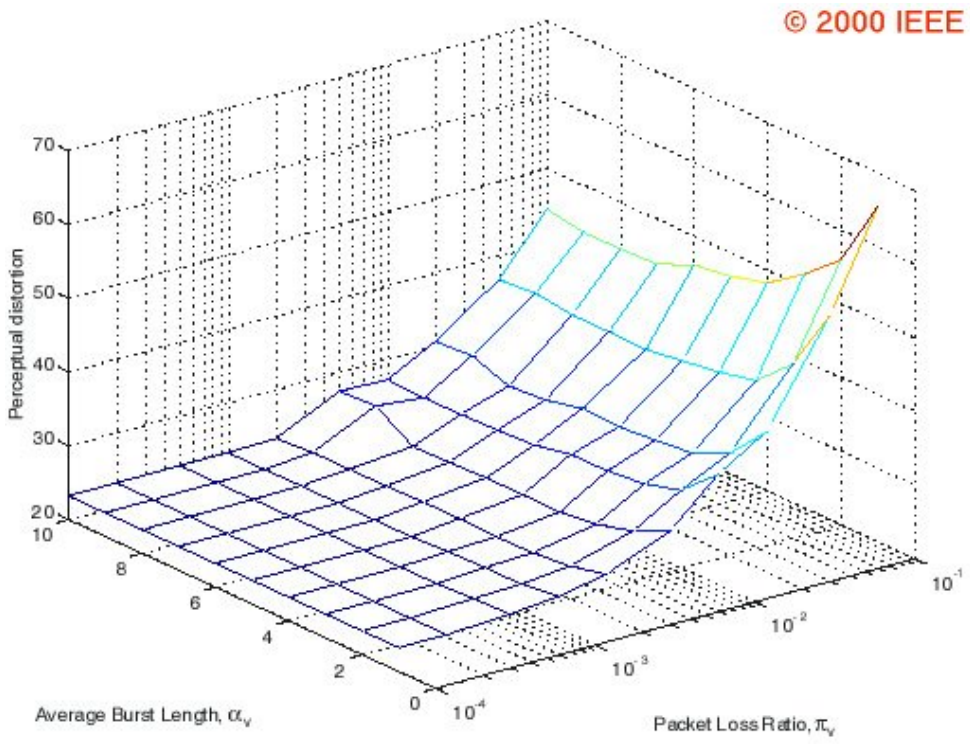

Fig. 2 The calculated perceived quality of MPEG-2 streams depending on packet loss (OWPL) and consecutive packet loss (COWPL)

3.1.3. Quality of other applications

Not much information is available on the other applications. One of the user groups trying to define SLS metrics comes from the game environment (which has similar issues as tele-emersion). 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]

3.1.4. 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 summarised as:

- Guaranteed bandwidth (comparable to available bandwidth – further referred to as the capacity)
- One way delay
- Jitter (ipdv)
- Packet loss

It is worth to note that packet loss is much lower rated than one would expect. This could very well mirror the present day low one-way packet loss in NREN's.

3.1.5. SLS metrics

The above considerations allowed creating a list of SLS metrics as shown below:

- Available bandwidth (AB): In general, each application requires a certain amount of bandwidth to transport its data. The detailed definition of the AB has not been finalized yet. A possible definition could be: "The Available bandwidth is the bandwidth usable by an application, that is equivalent to a leased line"
- One-way packet loss (OWPL) [35]: 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 application MPEG-2 unusable.
- Consecutive one-way packet loss (COWPL) [36] or One-way Loss Distance and One-way Loss Period [37]: One-way packet loss and consecutive one-way packet loss are primarily due to congestion (packets dropped in the bottleneck) or errors in circuits/interfaces. Minimizing the congestion and eliminating all circuit/hardware errors is an essential issue.
- IP packet delay variation (ipdv) [38]
  Ipdv is an another important QoS metric. It is mainly caused by buffering in IP routers and will be determined by congestion/prioritizing 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 or implementation) in routers.

A lot of work has to be done on this subject. Cooperation has been established with the Internet2 QoS Working Group [42]. More up to date information on the above topics can be found in [43]

3.1.6. Simulation of an impaired network

A tool that allows controlled, reproducible experiments with network performance sensitive/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/emulate:

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

TF-NGN members have gained no experience yet with this tool or others such as dummynet[71], because the TF-NGN priority is to make use of existing results instead of doing tests.

3.2. Tools for measuring the user-visible SLS metric

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

The section 3.1 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 behaviour of an application.

The following granularity is proposed for the user-visible SLS metric (possibly as low as possible systematic measurement error):

- Available bandwidth (AB): 300 kbit/s
- One-way packet loss (OWPL): half an order of magnitude
- Consecutive one-way packet loss (COWPL): one IP packet
- 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.

3.2.1. Overview of tools

This overview only provides an impression if necessary tools exist rather than 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.

3.2.1.1. Available bandwidth (AB)

There is no comprehensive definition of the tool that could determine this metric. A possible candidate is cprobe [46], but it has some errors due to its measurement method.

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

3.2.1.2. One-way packet loss (OWPL)

The systems than can measure this metrics include RIPE TTM[49 and annex 2], Surveyor [73] and Chariot [50 and annex 2].

3.2.1.3. Consecutive one-way packet loss (COWPL)

Chariot can provide insight in this parameter.

3.2.1.4. IP packet delay variation (ipdv)

This metrics can be measured by the tools that are based on RFC1889 [51], such as Chariot and MBone conferencing applications [52].

3.2.1.5. One-way delay (OWD)

To measure the OWD with a fine granularity, precise clock synchronization on both measuring sides is required. This can be done with GPS receiver such as with RIPE TTM, Surveyor[49] or by using PPS [53]. There is also an option of using NTP synchronization.

TF-NGN is still in search for tools, which could provide results with sufficient granularity (1 – 5 msec). The use of a nearby NTP-server provided on the same LAN is also considered. Chariot [54] may have an interesting clocking algorithm that can determine the OWD with a granularity of 1 msec without the use of NTP.

3.2.1.6. 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, COWPL and codec.
- VQmon (using passive measurements) also provides a modified MOS for VoIP ([70] and [56]), based on RTT, jitter buffer, OWPL, COWPL, packet loss distribution, recency (the way a listener would remember call quality) and codec.

Video MOS:

- No tools really exist that determines the Video MOS based on the SLS metric yet. ITS Video Quality Measurement (VQM) tool [57] is more concerned on the video picture quality side instead of the network issues that underlay this.

3.2.2. Evaluating tools

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 few (more general) tools have been tested (see Annex 2, and in more detail see [58] and [59]).

3.2.3. Conclusions

The above summary shows that for only one out of the five user-visible SLS metrics (available bandwidth) no tool exists. Besides existing tools are not yet in a fully integrated set. Members of TF-NGN will continue experimenting with some of the mentioned tools to better understand their operational environment and their pro's & con's.

Works need to be done in order to stimulate the development of methods, implementation and integration of tools (especially for AB 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 metric.

3.3. 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 an 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 will have to have the following characteristics:

- Support testing facilities for end-users and network managers. This means that the measurements need to be tailored to the correct metric and granularity for the specific user group.
- 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. A facility that is based on downloading scripts could support this flexibility (such as Chariot).
- Support of active and passive measurement. While both active and passive measurements are important in auditing the SLS, these measurement types need to be supported.
- 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, perhaps based on SSL [62] and/or AAA-servers.
- A comprehensive method for discovering the measurement points in a multi-management domain. Possible methodologies that could used are Jini or SOAP [B]
- 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].
- 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 (so defined methods for measuring [60] and exchangeable (which can be achieved by having a standard database, perhaps using XML and 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 and/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]).

3.3.1. A possible topology of the measurement infrastructure

A possible topology for SLS measurements is depicted on 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 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 one can use a system close to the communication path.

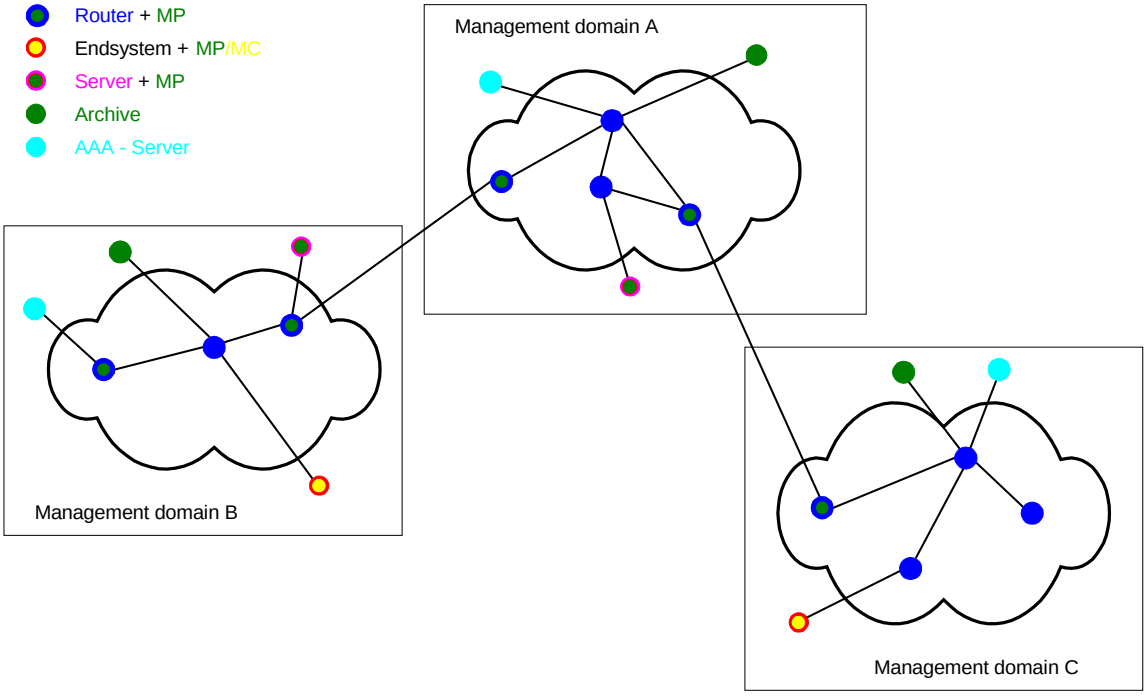

Fig. 3 Topology of the measurement infrastructure

- In principle everyone should be able make measurements between any two MP's, if one has proper access level. This is be done by using a Measurement Client (MC).
- The MC's must be able to downloading 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, etc.
- Archiving of measurements 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 building 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.

3.3.2. 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 will have to be studied in more detail in the coming period of TF-NGN activity. A cooperation with ViDeNet [66] has been established to study a possible implementation of such a measurement infrastructure.

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, http://members.home.net/garmitage/things/quake3-latency-051701.html, 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/internet-drafts/draft-ietf-ippm-loss-pattern-05.txt, July 20,2001

[38] IP Packet Delay Variation metric for IPPM, P. Chimento, http://www.ietf.org/internet-drafts/draft-ietf-ippm-ipdv-07.txt, Febr. 2001

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

[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://www.ietf.org/internet-drafts/draft-ietf-ippm-btc-cap-00.txt, February 2001

[49] Internet delay measurements using test traffic design note, H. Uijtendaal, http://www.ripe.net/ripencc/mem-services/ttm/Notes/RIPE_158/, 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, http://www.netiq.com/Downloads/Products/Chariot/Documentation/NetIQ_CHR_User_Guide.pdf, April 2001.

[55] Using Chariot to evaluate data networks for VoIP readiness, Walker, J.Q., Hicks, J., http://www.netiq.com/Downloads/Library/White_Papers/NetIQ_Net_Evaluating_Data_Networks.pdf, NetIQ, 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] A one-way active measurement protocol requirements, Shalunov S., Teitelbaum B., http://www.ietf.org/internet-drafts/draft-ietf-ippm-owdp-reqs-00.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

5. ACRONYMS

AAA	Authentication, Authorization, and Accounting
AB	Available Bandwidth
ACK	ACKnowledgement
AS	Autonomous System
ASIC	Application Specific Integrated Circuits
BGP	Border Gateway Protocol
BTC	Bulk Transfer Capacity
CLI	Command-Line Interface
COWPL	Consecutive One-Way Packet Loss
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
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
OWPL	One-Way Packet Loss
PFC	Policy Feature Card
PPS	Pulse Per Second
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
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, 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?
   - 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

This overview presents the test results for selected bandwidth/capacity measurement tools. It has been prepared as a summary of [58] and [59] 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 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 and Solaris.

Tool characteristics

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 approx. 72% of the real bandwidth for empty network and approx. 43% of the real bandwidth for the network with 80% load. Similar situations occurs 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, ipdv, packet loss, packet duplication, packer reordering and it can determine the MOS experienced in VoIP.
The tool can utilise multicast and TOS bits during these test.
The measurement points can be installed anywhere in the path as long as it is one of the OSs - approximately 15- supported by Chariot.

The tool simulates an application, so in principle it is possible to simulate streams of any speed desired. It is therefore also possible to test network or system behaviour under load. It uses user-level timestamping (no special user privileges are needed). It determines the one-way delay by means of a sophisticated synchronisation method. The granularity for one-way delay is in the order of 1 msec.

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 an active measurement 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 an access to both ends of the measured path. While client may be run by a non-privileged user, 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. This tool is not affected by Layer 2 devices, but inappropriate run-time parameters setting may affect the measurement quality.

This tool allows for an accurate measurement in 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.

Tool characteristics

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 are showing totally different numbers. For fast networks (with the speed > 100Mbit/s) reported bandwidth was much less than the real bandwidth – approx. 40% less. The measurements may be affected by Layer 2 devices or inappropriate setting of run time parameters.

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.

Tool characteristics

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.

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, ...). Results are available as both raw data and plots/numbers ready to be used by operators. The analysis tools run on Windows 95/NT and most Unices including Linux and Solaris.

The tool uses dedicated measurement hardware probes at each end. The tool runs at 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.