Project Number: **EP 29212**
Project Title: **QUANTUM**

Deliverable D6.2

Report on Results of the Quantum Test Programme

Deliverable Type: PU-Public
Contractual Date: 31 May 2000
Actual Date: 23 June 2000
Work Package: 6 – Test Programme
Nature of Deliverable: RE - Report

Authors:

Tiziana Ferrari CNAF/INFN
Simon Leinen SWITCH
Jan Novak DANTE
Simon Nybroe Ericsson/Telebit AS
Herve Prigent CRIHAN
Victor Reijs Heanet/SURFnet
Roberto Sabatino DANTE
Robert Stoy University of Stuttgart

Abstract:

This deliverable reports of the results on the Quantum Test Programme which evaluates emerging technologies with the ultimate goal to understand how to implement operational services useful to the NRNs. It follows on from D6.1 in which the interim results of the QTP were summarised. Some of the activities outlined in D6.1 have not been developed further and are not therefore repeated herein, whilst other activities will continue to develop until the end of the Quantum project (October 2000) after which more results will be made available. This report focuses on the results obtained since publication of D6.1and outlines which activities will continue until October 2000.

Keywords:

Differentiated Services, QoS, MPLS, IP Multicast, ATM, ATM signalling, IPv6

Table of Contents

1 Executive Summary 5

2 Summary of Results per Experiment 6

3 Multicast (IP and ATM) 8

3.1 Problem Statement 8
3.1.1 Inter-domain Multicast Routing 8
3.1.2 Intra-domain Multicast Routing, point to multi-point ATM-SVC service 9
3.1.3 User Site multicast performance monitoring and routing monitoring 9
3.2 Objectives of the Experiment 9
3.2.1 Inter-domain multicast routing 9
3.2.2 User site multicast performance and routing monitoring 10
3.3 Outline Solution 10
3.3.1 Inter-domain Multicast Routing 10
3.3.2 Multicast Performance and Route Monitoring 11
3.4 Current Results 11
3.4.1 TEN-155 multicast topology deploying MBGP/MSDP 11
3.4.2 Multicast Performance and Route monitoring 12
3.5 Future Activities 15

4 Differentiated Services and QoS Measurement 16

4.1 Problem statement 16
4.2 Objectives of the experiment 17
4.3 Outline solution 17
4.4 Resources 18
4.4.1 Loan 18
4.4.2 Hardware available on site 18
4.4.3 Test partners 18
4.5 Description of the experiments 18
4.5.1 Technical set-up 18
4.5.2 QoS metrics 19
4.5.3 Methodology 20
4.5.4 Planned timetable and work items 21
4.6 Results of the experiments 21
4.6.1 Transmission queue 21
4.6.2 Weighed Fair Queuing service rate 22
4.6.3 Priority Queuing 24
4.6.4 Comparison of WFQ and PQ 26
4.6.5 PQ under traffic congestion and aggregation (multi-hop scenario) 28
4.6.6 Comparison of WFQ and PQ in the WAN 32
4.6.7 WRED performance 33
4.7 Summary of test results 35
4.8 Implications for future services 36
4.9 References 36
4.10 Appendix 38
A Priority queuing 38
B WFQ 38
C WRED configuration on router C7500 40

5 MPLS 42

5.1 Introduction: an overview of MPLS 42
5.1.1 Implications 42
5.2 Experiments 43
5.2.1 Goal of the experiments 43
5.3 Schedule 43
5.3.1 February 2000 : Preparation of the tests. 43
5.3.2 March 2000 43
5.3.3 April-June 2000 44
5.4 Documentation / day-to-day operations 44
5.5 Description of the tests done in Cisco Laboratories 44
5.5.1 Fast reroute on the MPLS backbone 45
5.5.2 VPNs 45
5.5.3 Traffic engineering 45
5.6 TF-TANT European testbed 45
5.7 Experiments to date 46
5.7.1 MPLS fast re-route feature 46
5.7.2 MPLS VPNs 46
5.7.3 Traffic Engineering 47
5.7.4 QoS Mapping on an MPLS network 49
5.7.5 Interoperability 49
5.8 References 49

6 Flow-based Monitoring and Analysis (FloMA) 50

6.1 Introduction 50
6.2 Motivation 50
6.3 Flow-based Accounting Mechanisms 50
6.3.1 NetFlow 51
6.3.2 LFAP 51
6.3.3 RTFM 51
6.4 NetFlow Accounting 52
6.4.1 Flow Definition in NetFlow 52
6.4.2 NetFlow Export 53
6.4.3 Distributed NetFlow 54
6.4.4 Router-based Aggregation 54
6.4.5 Handling (Or Ignoring) Flow Start/End Times 55
6.4.6 Heuristics to Determine Application Protocols 56
6.4.7 Overview of Existing NetFlow Post-Processors 56
6.5 Conclusions 58

7 IP over ATM 59

7.1 Introduction 59
7.2 Objectives 59
7.3 Test Plan 59
7.3.1 60
7.3.2 Use of SBR3 with SCR >> 0 60
7.4 Fine tuning of SBR parameters 61
7.5 Future work and implications for future services 61

8 IPv6 62

8.1 Objectives of the Experiment 62
8.2 Description of the Experiments 62
8.2.1 Interoperability tests (JOIN) 62
8.2.2 Multihoming issues for QTPv6 64
8.2.3 DNS 67
8.3 References 70

9 SDH issues 71

9.1 Description 71
9.2 Goals 71
9.3 Effects of SDH implementations on aggregated IP traffic 71
9.3.1 European SDH/STM implementations 71
9.3.2 American SDH/OC implementations 72
9.4 Raising awareness of implications of SDH concatenations issues 72

10 Virtual Private Network service 73

10.1 Description 73
10.1.1 What is a VPN 73
10.2 Goals 74
10.3 Standardisation and implementation status of VPNs in IP networks 74
10.4 Effects of VPNs in WAN environments 75
10.5 Security issues 75
10.6 VPNs in National Research Networks 75
10.7 Testing VPNs in the TF-TANT network environment 75
10.8 References 76

11 Wave Division Multiplexing (WDM) 77

11.1 Description 77
11.2 Goals 77
11.3 Standardisation process and use of WDM in WANs 77
11.3.1 Documentation 77
11.3.2 Activities within the NRNs 77
11.4 Test environment 78
11.5 Management and resilience issues 78
11.6 References 78

12 Glossary of Terms 80

1 EXECUTIVE SUMMARY

The goal of the Quantum Test Programme (QTP) is to evaluate emerging technologies with the aim of understanding how to implement operational services with them. Particular attention is paid to the provisioning of Quality of Service (QoS), but also to multicast (IP and ATM), IPv6 and ATM signalling. Other efforts are devoted to understanding and developing techniques in support of the above such as QoS monitoring, flow-based monitoring, route monitoring and policy control.
A joint DANTE/TERENA task force, TF-TANT, carries out the work.
Activity on the QTP started in November 1998 with a meeting in Cambridge, in which the areas of activity and the people responsible for them were identified. Subsequently, work was carried out to define in more detail the test proposal for each experiment, together with the finalisation of the participants in each activity and the definition of a test plan.
Each activity has been developed to a different level. The reasons for this are mainly due to prioritisation within the task-force: most of the members of TF-TANT take part in more than one activity, therefore there are simple time constraints on the possibility of performing all tasks at the same time. Other factors such as availability of test equipment influenced the development of the activities.
An interim report on the results of the testing activity was produced in October 1999 (D6.1). Since then there has been significant progress in some activities, whilst other activities have either not been developed further or have had limited development.
The task force plans to continue its work until October 2000, and more results on activities such as MPLS and IPv6 will be available. A report on these further results will be produced and made publicly available.
Various equipment vendors have contributed to some of the work items. Cisco and Netcom Systems have contributed with the loan of equipment for the testing of MPLS. Cisco, IBM, Netcom Systems and Cabletron have contributed with the loan of equipment for the testing on diff-serv and QoS monitoring. Telebit Communications is an Associated Partner of the Quantum project whose involvement is mainly for the QTP, with particular attention to IPv6.
Work on IP multicast has been very successful in that in October 1999 a pilot IP multicast service based on PIM-SM, MBGP and MSDP started on TEN-155 and this has been transformed into an operational service in April 2000.
The developments of the Flow Measurement and Analysis activity of the task force have also been partly deployed on the TEN-155 operational network and NRN operational networks, especially in the area of traffic analysis and DoS detection.
The work on IP over ATM has outlined that the current use of ATM on TEN-155 is the most effective, whilst other mechanisms that in theory may be more efficient are either not developed yet or not working properly.
All the other activities have produced many experimental results, but have not resulted in service specifications nor pilot services. It is expected that the Diff-Serv, IPv6, Policy Control and MPLS activities will produce such results in the period May-October 2000.

2 SUMMARY OF RESULTS PER EXPERIMENT

- **IP and ATM multicasting**: This experiment focused on IP multicasting, and resulted in the deployment of native IP multicast, using PIM-SM, MBGP and MSDP, on TEN-155. The work was done mostly on Cisco routers, but the technologies have also been evaluated on Juniper routers by DANTE. In July multicast will be supported on TEN-155 using both Juniper and Cisco routers. Additional work has focused on mechanisms for monitoring the stability of multicast (MRM). Future work should address BGMP/MASC and the mapping of ATM point-to-mutipoint functionality to IP multicast.

- **IP over ATM**: As TEN-155 is based on ATM, this work item was introduced in order to ensure that the most effective way of providing an IP over ATM service in combination with an ATM based managed Bandwidth Service (MBS) on TEN-155 was adopted. The work was carried out by DANTE, KPNQ and Lucent, whilst the TF-TANT group acted as a forum for discussion and consultation. The work outlined that the current use of ATM on TEN-155 is the most effective, although a detailed level of tuning of the ATM parameters was required to reach this objective. No further work is planned for this activity.

- **Differentiated Services and QoS Monitoring**: Since D6.1, the testing activity has focused on techniques to implement AF and EF PHBs. To characterise the PHBs, measurements focused on one-way delay, as defined in RFC 2697 and packet delay variation. The techniques tested were Cisco's Priority Queuing, WFQ and WRED. These were tested in situations of congestion and with variable background traffic. The results showed that the actual behaviour of the PHBs depends on the characterisation of the background traffic and the EF or AF traffic. It also depends on the level of aggregation performed and on internal buffering parameters within the routers. In general the results show that the techniques have an acceptable behaviour when applied to aggregated traffic. More testing is planned until October 2000, in particular in respect to integration with MPLS and Policy Control implementations.

- **MPLS**: The results published in D6.1 outlined general stability of the technology which was tested only on Cisco equipment, but also various performance problems related to re-routing and stability of the code for VPN functionality. Due to a change in the experiment leader, activity has resumed in March 2000, so few new results are available in this report. However, a new test plan has been detailed and some new results are available. Some of the tests done are a repetition of the tests described in D6.1 either because poor results (re-route) were obtained or because new and more stable software is now available. The new results show that the performance problems related to re-route have now been solved and the new software for VPNs is more stable. More results on Traffic Engineering and integration of MPLS and diff-serv are expected in October 2000.

- **Flow-based Monitoring and Analysis**: In this work package, we look at recent developments in traffic accounting, in particular router-based accounting methods, and some possible applications of these mechanisms in the context of backbone IP networks such as National Research Networks (NRNs) or a Trans-European backbone Network (TEN). The activities in the period covered by this report were mostly related to the exchange of experience between the participating networks, and to an analysis of accounting mechanisms, in particular Cisco's NetFlow accounting, and the tools available to operate in conjunction with them. In addition, an experimental set-up has been configured on one of the workstations at TEN-155's Point of Presence in Geneva. The samplicator tool is used to send it a copy of the NetFlow accounting data from the Geneva router (ch.ten-155.net), aggregating all traffic for CERN and SWITCH, as well as one of TEN-155's interconnections to AUCS for the commercial part of the European Internet and some transit traffic for other NRNs.

- **Policy Control**: No experimentation has taken place yet, and the plans remain the same as described in D6.1. There have been a number of presentations of techniques to the TF-TANT group on commercial implementations of policy control mechanisms and experimentation with these is expected to start in June 2000. Therefore detailed results will be available in October and no further reporting is contained in this deliverable

- **IPv6**: An international testbed (QTPv6) has been set-up, and work is underway to evaluate issues such as DNS, multi-homing, applications and interoperability. Several interoperability problems have been outlined as well as the instability of DNS for native IPv6. All the testing with Cisco equipment was with pre-production software, and some of the problems encountered are documented in they Release Notes. More results will be available in October 2000.

- **ATM Signalling**: The goal of this experiment was to develop an ATM signalling service for TEN-155 and the connected NRNs. DANTE and KPNQ tested and developed ATM signalling on TEN-155, but the tests were not extended into the NRNs, as none planned to use it on an international scale. The final results are as published in D6.1, and no further work is planned nor has taken place on this activity. Therefore ATM signalling is not included in this report.

- **RSVP to ATM signalling mapping**: Although the activity is described in sufficient detail in D6.1, no work was carried out on this activity, purely for human resource reasons. It is not planned to develop further this activity as the longer term developments of pan-European Research Networking will rely less on ATM and more on IP technologies.

- **SDH concatenation issues**: This activity was mostly concerned with raising awareness within the research and telecommunications service provider communities of the implications of concatenated vs. non-concatenated SDH services. It also aimed at understanding what the effects of non concatenated SDH services are on aggregated traffic streams.

- **WDM**: Some NRNs have deployed or are in the process of deploying national backbone services based on WDM technology. This activity aimed at providing a forum for exchanging experiences and make use of available fibre between countries for experimentation. No fibre was however made available for use for the task force.

- **VPN**: This activity was carried out mostly within the MPLS activity

3 MULTICAST (IP AND ATM)

Experiment Leaders: Robert Stoy – University of Stuttgart, Jan Novak - DANTE

Participants: all TEN-155 NRNs

Keywords: DVMRP, PIM-SM, MBGP, MSDP, MRM

3.1 Problem Statement

For several years multicast on the internet has been supported by the Mbone, which is in effect an overlay network interconnecting several routers on the internet. The technology used is based on DVMRP, which is a multicast forwarding and routing protocol. DVMRP is a distance vector routing protocol and therefore does not have the ability to apply routing policies.

The current DVMRP based world-wide Mbone is reaching its limits in providing an IP multicast service in a production environment because of the following reasons:
DVMRP as a distance vector routing protocol does not scale with the growing numbers of multicast sessions and networks providing a multicast service;
Multicast forwarding with DVMRP is based on a flood and prune mechanism which in too many cases is not implemented correctly thus leading to a considerable amount of unwanted traffic on the internet and causing congestion on the Mbone itself,
The multicast infrastructure is based mainly on rate-limited tunnels, which are affected by the congestion due to the flood and prune mechanism. Tunnelling in itself is a cause of performance limitations;
The multicast topology is different form the unicast Internet, thus leading to sub-optimal utilisation of expensive international circuits.

As a result, multicast sessions are often suffering from high packet losses, and inefficient multicast routes leading overall to poor performance and quality.

The solution to these problems is the introduction of hierarchical multicast routing topologies by using new inter-domain and intra-domain routing protocols.
Accompanied to these efforts a permanent and effective multicast performance and routing monitoring environment between user sites and between the multicast domain borders is required to detect and isolate faults in the hierarchical multicast infrastructure to ensure good quality of the mutlicast service across the domain borders up to the user sites.

3.1.1 Inter-domain Multicast Routing

The IETF Mbone-WG and IDMR-WG have been working for several years on a new multicast forwarding and routing protocols. Both inter-domain and intra-domain protocols are being developed and standardised. For intra-domain forwarding and routing the development of PIM-SM has reached stability that allows its deployment.

Inter-domain routing and forwarding, which are needed to provide a multicast service across multicast domain borders is, on the other hand, still being developed. The IETF has engineered an interim solution to inter-domain routing, which is the combination of the MSDP protocol (Multicast Source

Discovery Protocol) together with the MBGP protocol (Multi-protocol BGP), Implementations of these protocols are available on routers.

On the longer term the IETF works on a "final" multicast distribution model, the BGMP/MASC architecture within the BGMP and MALLOC working groups. First implementations of components of this architecture have been expected to appear in fall 1999; they have been delayed and first implementations are going to be available during spring 2000. These components should be tested as soon as thy are available

3.1.2 Intra-domain Multicast Routing, point to multi-point ATM-SVC service

TEN-155 provides an IP over ATM service according to RFC 1483, and the same applies for the IP multicast service. With the availability of ATM SVCs the mapping of PIM-SM to point-to-multipoint (p2mp) ATM SVCs is potentially an effective way of distributing IP multicast within TEN-155 backbone if it would be based in future purely on ATM.
RFC 2337 *Intra-LIS IP multicast among routers over ATM using Sparse Mode PIM* describes this mapping in detail.

Some initial basic tests of this functionality have been successfully carried out within the TF-TEN project in 1998. Before this functionality could be introduced on TEN-155, point to multi-point ATM SVC should have been tested as well as the QoS associated to these.

However in July 2000 the TEN-155 backbone topology will move from a pure ATM based infrastructure to a mixed ATM and Packet-over-Sonet infrastructure which leads to a unfeasibility of the original idea.

3.1.3 User Site multicast performance monitoring and routing monitoring

Multicast network performance monitoring between user sites is currently done only from time to time and often only during troubleshooting procedures, after problems occurred and when users send trouble tickets. Multicast network statistics are required to analyse problems and can help to find and locate problems *before* they become urgent.

Currently the mtrace program, which provides multicast routing path and per hop packet loss information, is used together with RTP/RTCP based monitoring tools between user workstations for failure isolation *after* problems have occurred.
The disadvantages of using this procedure are its inflexibility during fault isolation, because workstations are required that can normally not be accessed by network administrators and the lack of the involvement of routers as active components for originating multicast traffic.

As a component of the test set-up in the inter-domain activities a multicast service routing and performance monitoring system at the user sites and, with involvement of multicast routers within NRNs and TEN-155 will be established. This can be used for monitoring a production multicast backbone service. Besides the common tools, such as MRTG and HP Open-View the emphasis of the tests will be on the Multicast Reachability Monitor (MRM) which addresses the disadvantages of the current mtrace/RTP/RTCP usage described above.
This mechanism is currently worked out within IETF's mbone-WG and first implementations of its components are available on workstations and Cisco routers since winter 1999.

3.2 Objectives of the Experiment

3.2.1 Inter-domain multicast routing

The migration from DVMRP to PIM-SM/MSDP required the following steps to be performed:

- Evaluation of Cisco's implementation of MBGP/MSDP on non production equipment. Cisco was chosen as test equipment because these are the routers currently deployed on TEN-155;
- Implementation of MBGP/MSDP on TEN-155 routers;
- Connection of NRNs to MBGP/MSDP in the core and tests of data transfers between different domains.

As soon as test implementations of the MASC/BGMP components are available, these should be tested in a non-production environment. Depending on the test results (and also on the status of deployment of these protocols in the general Internet) the migration from MBGP/MSDP to BGMP/MASC will be planned.

3.2.2 User site multicast performance and routing monitoring

The goal of the monitoring activity is to provide an environment that allows end-to-end monitoring between dedicated user sites and between selected, ideally domain border routers, on one permanent dedicated multicast test session, which is used to periodically generate a small load on the network for troubleshooting purposes. The system should have a WWW interface.

3.3 Outline Solution

3.3.1 Inter-domain Multicast Routing

The initial implementation of IP multicast on TEN-155 with deployment of MBGP/MSDP is described in detail in the following documents:

http://www.dante.net/mbone/mcast99/migration.html and
http://www.dante.net/mbone/mcast99/mphase2.html

Basically, test equipment in the TEN-155 PoP in Frankfurt was installed and the TEN-155 DVMRP cloud was connected to it. DFN and CESNET were also connected to it using the new protocol stack MBGP/MSDP. The stability of the test router with new software and data transfer between DVMRP and MBGP/MSDP domains were successfully tested. After this MBGP/MSDP was enabled on TEN-155 routers in Sweden, Netherlands, United Kingdom and France.
SURFnet, NORDUnet, Grnet, RCCN and RedIRIS have a native connection to TEN-155's multicast infrastructure, whilst Belnet, CESNET, DFN, Machba/ILAN, SWITCH and Renater use dedicated equipment. This dedicated equipment was connected either via separate ATM PVC or an IP-in-IP tunnel to the TEN-155 equipment.

TEN-155's multicast was successfully stress-tested during the IETF meeting in Oslo in July 1999 (when it delivered data to both the DVMRP and MBGP clouds) during which two parallel sessions of 2 Mbps each were transmitted via TEN-155.

From these initial steps the topology of TEN-155 multicast has evolved as shown in Figure 1
A description of the further steps to the current situation is described in detail in the following document:

http://www.dante.net/mbone/mcast99/art

In parallel to the migration of TEN-155 to MBGP/MSDP, the BGMP/MASC development at IETF was followed. A first implementation of BGMP is available in GATED and has been retrieved for initial testing. A first MASC implementation is also available. The specification of a BGMP/MASC test description is ongoing work.

3.3.2 Multicast Performance and Route Monitoring

The tests and measurements are done on a dedicated multicast session called 'Places all over TEN-155'. MRM is used to provide:
end to end loss statistics between workstations;
end to network border statistics between workstation and network border routers.
Route monitoring using mtrace provides end to end multicast route information together with per hop loss statistics reported by the routers on the multicast route.
The results are displayed on a WWW page at RUS, whilst real-time display of graphical results is ongoing work.

Phase 0
In phase 0 evaluation and initial tests with the MRM workstation implementation from UCSB (http://imj.ucsb.edu/mrm) in a local testbed at RUS have been done.

Phase 1
On the following phase, started in Feb. 2000, MRM agents have been installed on workstations at RedIRIS, CESNET, DANTE (UK) and RUS/DFN. A central MRM manager at RUS controls these agents and initial measurements have been made on the TEN-155 multicast infrastructure.
Permanent MRM measurements have been started between the workstations on the dedicated multicast test session 'Places all over TEN-155'.
The MRM measurement results are currently stored on the MRM manager workstation at RUS. A real-time graphical display of the results on a WWW server at RUS is ongoing work.

Phase 2
In the future phase 2 additional MRM agents on workstations at other NRNs will be installed and the MRM functionality on some selected NRN's Cisco routers will be deployed. This will enable the testing of the MRM functionality on Cisco routers and the interoperability with the workstation implementations, and second to thus increase the statistics coverage.

Since phase 1 The MRM measurements are accompanied by a permanent route monitoring using mtrace, which results and are displayed in a textual form in real-time on a WWW page at RUS (http://www-ks.rus.uni-stuttgart.de/TF-TANT/mtrace-out).

3.4 Current Results

3.4.1 TEN-155 multicast topology deploying MBGP/MSDP

The following figure illustrates the TEN-155 multicast topology as of March 2000:

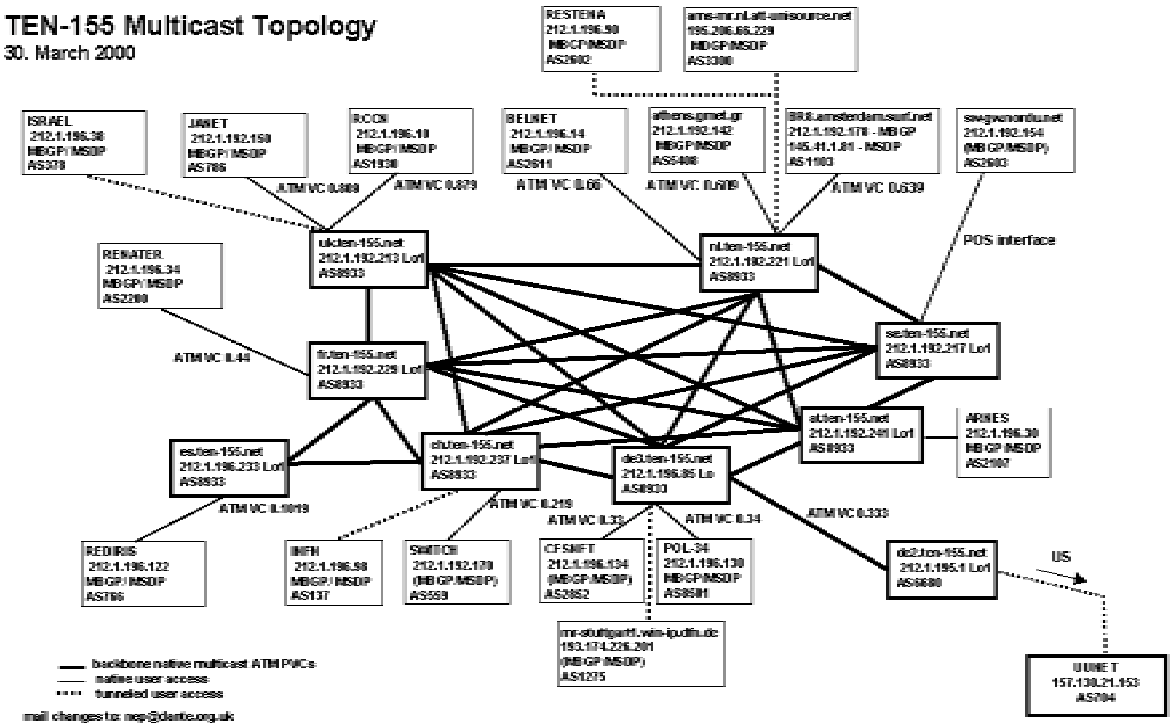


Figure 1 TEN-155 Multicast Topology

The TEN-155 native multicast service is being offered on a best efforts basis, whilst arrangements are being made to transform this into a fully operational service. This requires IP NOC training for monitoring and troubleshooting.

3.4.2 Multicast Performance and Route monitoring

During phase 2 MRM tests have been done using UCSB's beta implementation on workstations over some NRN's and TEN-155's multicast 'service' as shown in the following figure.

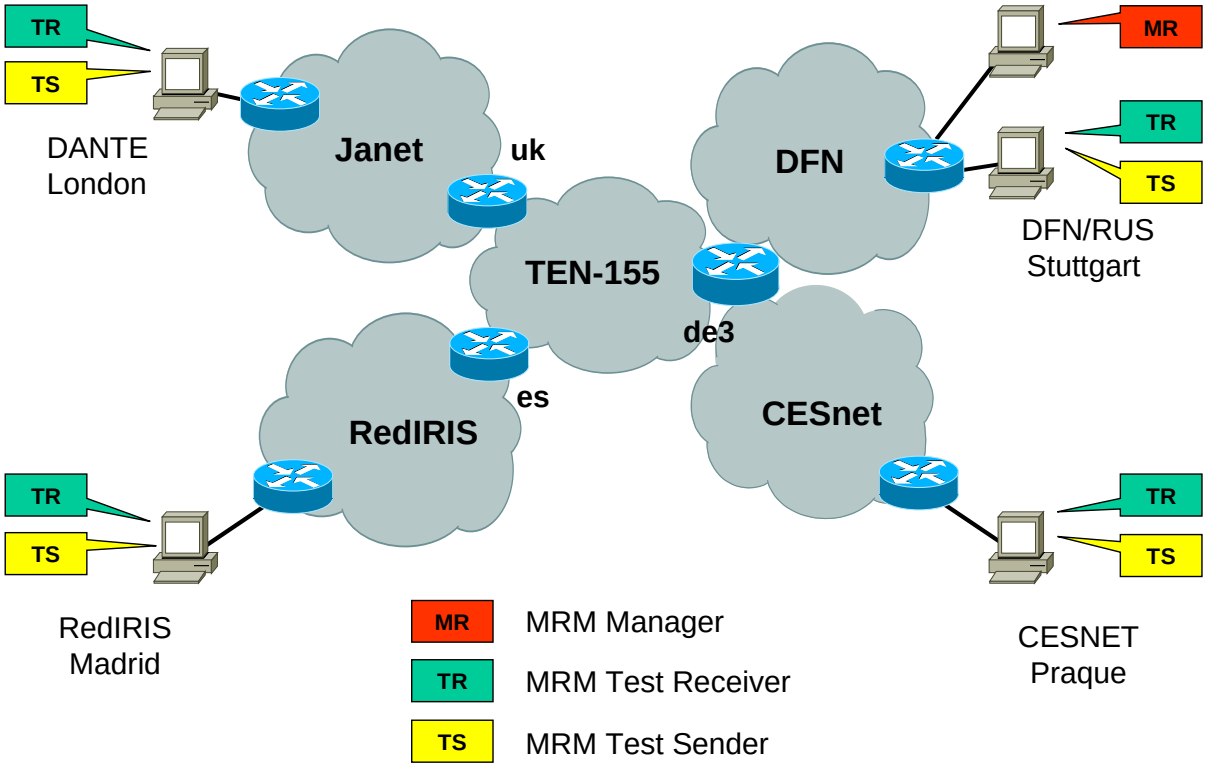


Figure 2 Test-Setup Multicast Monitoring Phase 1

The measurements have been controlled by the MRM manager at RUS which:

- started Test Senders (TS) at DANTE, RedIRIS, and CESNET. The parameters given to the Test Senders were to send a constant rate of about 100 kbps into the mulitcast group 'Places all over TEN-155'.;
- started a Test Receiver (TR) at RUS, which was told to join the same multicast group and to calculate packet duplicate or loss ratio based on an average sliding window of 1 second.

The reports sent from the Test Receiver to the Manager (MR) are displayed on the workstations terminal window as shown in the following example in which only one test sender on the RedIRIS workstation sent traffic into the multicast group to which only one test receiver at RUS was joined.

```
Received ACK from 130.206.1.32
Received ACK from 129.69.30.20
Timestamp       Manager         Test Receiver   Test Sender     L/D (%)    Ehsr
Apr 14 22:29:45 129.69.30.15    129.69.30.20    130.206.1.32    100 (100%) 0
Apr 14 22:29:46 129.69.30.15    129.69.30.20    130.206.1.32    100 (100%) 0
[....]
Apr 14 22:30:21 129.69.30.15    129.69.30.20    130.206.1.32    100 (100%) 0
Apr 14 22:30:22 129.69.30.15    129.69.30.20    130.206.1.32    39  (39%)  4776
Apr 14 22:30:29 129.69.30.15    129.69.30.20    130.206.1.32    11  (11%)  5472
Apr 14 22:30:30 129.69.30.15    129.69.30.20    130.206.1.32    11  (11%)  5579
```

After receiving confirmations (ACKs) from the test sender and test receiver, the manager displays for each report received from the test receiver:

- the timestamp when the report was received,
- the MR's address to which the report was sent,
- the TR's address which sent the report,
- the source (i.e. the TS) of the test traffic received by the TR,
- the number of lost or duplicated packets during the last sliding average window (1s was configured). Duplicated packet are shown as negative numbers. If there is no lost or duplicated packet the TR will not send a report.

- the calculated packet loss or duplicate ratio (in the example the packet number and packet ratio are equal because the senders are configured to send 100 packets per second)
- and the sequence number of the last test packet received from the TS.

The calculation of the lost or duplicated packets is based on the sequences numbers of the test packets which the TR receives from the TS.
The reports with 100 % packet loss ratio at the beginning of the MRM test session indicate that the multicast distribution tree was not established immediately after the session start. The time difference between the last 100% loss report and the session start time can be therefore interpreted as the multicast distribution path set-up time.

The following figure shows a graphical representation of TR loss ratio reports displayed on the manager during first long term measurements (12 hours) between a TS on the Rediris-WS and a TR on the RUS-WS. The result was validated with mtrace runs in 5 min intervals, which showed during the given test timeframe packet losses between 10 % and 20 % on each of two routers along the multicast path, whereas on the other routers no losses have been shown.

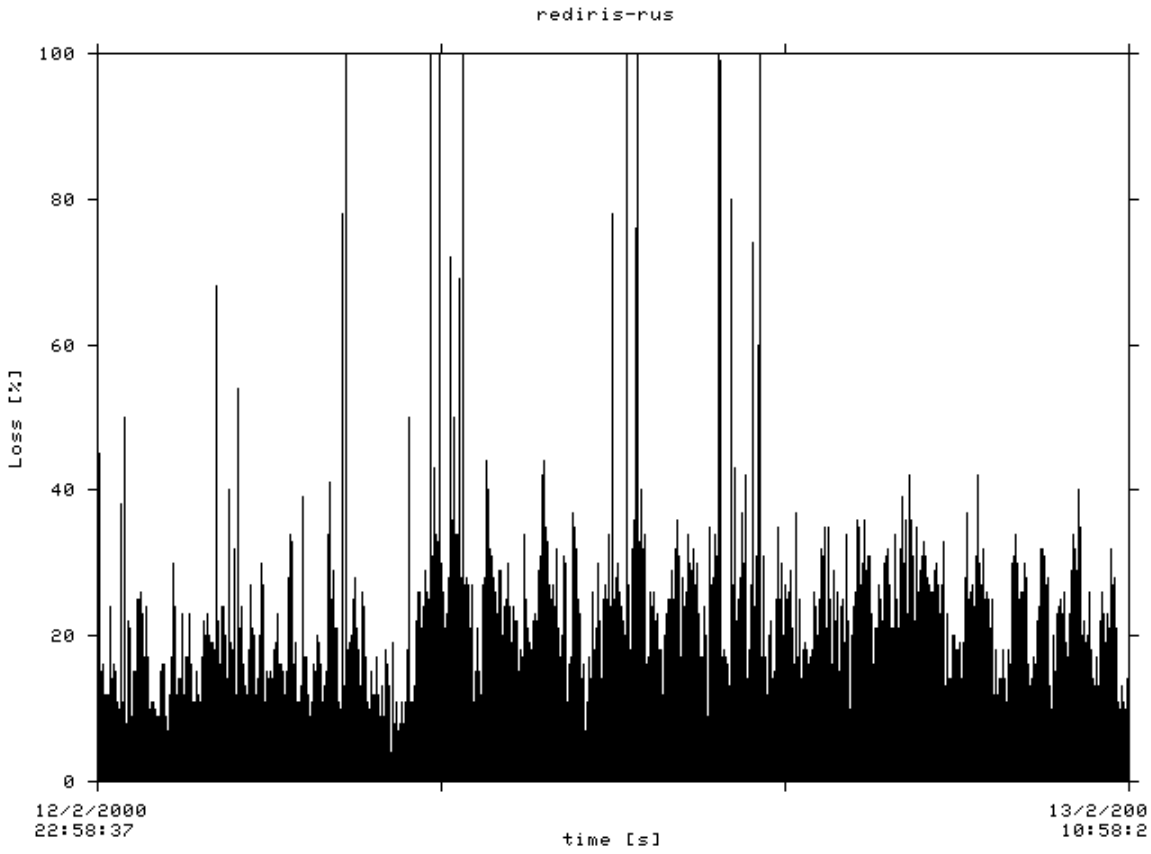


Figure 3. Graphical representation of packet losses reported during a MRM test session between RedIRIS, Madrid and RUS, Stuttgart

A more detailed description of the above described phase 1 tests is available in:
http://www-ks.rus.uni-stuttgart.de/TF-TANT/mrm-results/rus-rediris-1

3.5 Future Activities

- Future ongoing evaluation of the MASC/BGMP implementations and definition of a test description
- Phase 3 of the multicast monitoring activity, involving Cisco routers for interoperability tests and on a larger scale covering as much as possible NRNs

4 DIFFERENTIATED SERVICES AND QOS MEASUREMENT

Experiment leader: Tiziana Ferrari - CNAF-INFN

Participants: CERN, CSELT, DANTE, GRNET, IAT, REDIris, RUS, SURFnet, SWITCH, University of Twente, University of Utrecht

Keywords: AF, DSCP, EF, IP QoS, PHB, PQ, TOS, Smartbits, WFQ, WRED

4.1 Problem statement

The experiments described in this report address the problem of Quality of Service (QoS) support in IP networks. Experiments focus on the *Differentiated Services* (diffserv) QoS architecture specified in RFC 2475.
The Diffserv architecture is based on the idea of classifying IP flows into aggregates and differentiating the service given to the aggregates rather than to flows individually. The treatment of aggregates is defined by Per-Hop Behaviors (PHB) which give the conditions that packet handling mechanisms must meet for certain classes of traffic.
Packets are classified by access routers or hosts, as belonging to one of a set of behavior aggregates (BA) and this is indicated by a specific value of the DSCP (DiffServ Code Point) in the IP packet header as specified in RFC 2474. In the core PHBs operate on behavior aggregates.

The activities carried out by the TF-TANT task force and described in this report are divided in two broad categories:

- Group 1: tests for the performance analysis applied to different queuing algorithms, namely Weighted Fair Queuing (WFQ) and Priority Queuing (PQ) for the support of delay and jitter-sensitive traffic;
- Group 2: tests of WRED (Weighted Random Early Detection), a selective packet discard mechanism which allows relative traffic differentiation within a given queue. WRED can be deployed for the support of the Assured Forwarding (AF) PHB Groups defined in RFC 2597.

The metrics used for the analysis are: one-way delay, instantaneous packet delay variation, packet loss percentage and burst size for tests in Group 1; per-class aggregate throughput for WRED testing in Group 2.

Scheduling is a fundamental QoS component for traffic differentiation in packet-handling devices.
According to the WFQ algorithm [6,7] packets are scheduled in increasing order of *forwarding time*, which is an attribute computed for each packet upon arrival. It is a function of both the packet size and the weight of the queue the datagram belongs to. On the other hand, with PQ [8,9] every time the ongoing transmission is over, the priority is checked: The priority queue is selected for transmission if it contains one or more packets while the transmission control is released only when/if the priority queue is empty. While WFQ provides a fair distribution of link capacity among queues, PQ is starvation-prone.
Given the difference of the two algorithms, their performance is compared under different traffic scenarios to identify the most suitable approach for the provisioning of delay, jitter and packet loss guarantees. The goal is to identify configuration guidelines for the support of the Expedited Forwarding (EF) per-hop behaviour (PHB) for the support of services.

WRED is an additional traffic differentiation mechanism for packets belonging to a given queue. As usual the packet class needs to be identified by the label carried in the IP header (either the diffserv code-point – DSCP – or the IP precedence). WRED can be deployed for the implementation of the AF PHB Group [2], since differentiation does not require the distribution of packets among different queues so that packet reordering within a given AF group is avoided.

Different WRED configurations are studied to verify their impact on the TCP performance[1].

4.2 Objectives of the experiment

The goals of the tests here reported are many-fold:

- The definition of implementation guidelines of the EF PHB for the support of delay and jitter-sensitive traffic;
- The detailed analysis of some scheduling algorithms in different traffic and network scenarios;
- The study of the influence of traffic aggregation on the end-to-end performance of a single reference flow;
- The tuning of WRED configuration for the optimisation of TCP performance and for an effective flow isolation within a single traffic class.

The queuing analysis aims at the development of services addressing the needs of applications which are one-way delay and jitter-sensitive such as real time applications for remote hardware control, interactive applications, audio and video streaming etc. On the other hand, the WRED mechanism is suitable for the implementation of services based on relative degrees of guarantees, a type of data treatment which is particularly suitable for data applications which are packet-loss, delay and jitter-tolerant.

4.3 Outline solution

The service implementation is based on the following functions:

- *Classification* and *marking*: multi-field classification is applied to traffic in the input direction at the ingress router. In our experiments it is performed according to the content of the source and destination IP address fields and to the transport protocol in use.
- UDP is deployed both for Best-Effort (BE) and EF traffic. UDP simplifies the performance analysis related to the reference stream, since with UDP both the stream output rate and the datagram size can be easily controlled through explicit configuration. Setting of the output rate is particularly important to keep the output interface under congestion[2].
- *Policing*: with WFQ EF traffic is subject to policing in the ingress router to make sure that the departure rate configured is greater than the arrival rate and the queuing delay introduced by the queuing system is minimised. On the other hand, with PQ policing ensures that only a limited fraction of traffic is injected into the priority queue so that starvation at the egress interface is prevented.
- *Scheduling*: it is deployed to guarantee separation between traffic classes, i.e. between EF and BE in our test scenarios. The adoption of a scheduler is recommended at each egress interface that represents a potential bottleneck in the network.
- *WRED:* The length of the queue is constantly monitored to avoid congestion by selectively discarding packets according to the precedence or DSCP in the header. A set of parameters defining the drop behaviour of the class is associated to each precedence or DSCP. WRED is enabled for a given queue by configuring different discard thresholds and drop probabilities for e given set of precedence values.

Performance is estimated by injecting an EF stream, the *reference stream,* to which measurement is applied. Specialised hardware, the SmartBits 200 by Netcom Systems, is deployed: it supports packet time stamping in hardware for an accurate delay and jitter estimation.

[1]In this report we summarize the preliminary WRED test results: the analysis of WRED is an ongoing activity.
[2]In the queuing implementation under test by CISCO scheduling is activated if and only if the transmission queue of the interface is under congestion.

4.4 Resources

4.4.1 Loan

As already reported in [12] two different equipment loans and one donation were made available to several test sites:

- CISCO loan: 1 router C7200, 2 routers C7500, 1 ATM switch LightStream 1010
- Hardware distributed to: GRNET, INFN and RedIRIS
- IBM donation: 5 IBM 2216, 5 IBM 2212.
- Hardware distributed to: CERN, GRNET, INFN, Uni. of Stuttgart and Uni. of Utrecht
- Netcom Systems: 3 SmartBits 200 with GPS kit (GPS antenna and GPS rx).
- Hardware distributed to: INFN, Uni. of Twente and Uni. of Utrecht

4.4.2 Hardware available on site

In each test site dedicated test equipment was made available as listed below[3]:

- ***Test workstations***
  - Platforms: HP, Linux, Sun Solaris. Workstations with several types of network interface were available: ATM, Ethernet, Fast Ethernet and Giga Ethernet.
- ***Traffic generators***
  - 1 SmartBits equipped with 2 10/100 Ethernet interfaces (for traffic generation) and 1 ctrl Ethernet interface (for the configuration of the apparatus), 1 GPS receiver and 1 GPS antenna. The SmartBits are configured through the Windows application called SmartWindow (version 6.53.18).
- ***ATM and LAN switches***
  - 1 ATM switch per site
  - 1 Cabletron Smart Switch Router (INFN, Uni. of Utrecht)
- ***Routers deployed in the testbed***
  - One router CISCO 7200 per test site: IOS 12.0(6.5)T7 – Maintenance Interim Software -. Other router platforms being part of the network are: Cisco 7500, IBM 2216 and 2212 and FreeBDS routers (at IRISA).

4.4.3 Test partners

Sites directly involved in testing activities during Phase 2 are: CERN (CH), DANTE (UK), GRnet (GR), CSELT (IT), IAT (IT), INFN-CNAF (IT), RedIRIS (SP), SWITCH (CH), Uni. of Bologna (IT), Uni. of Stuttgart (DE), Uni. of Twente (NL), Uni. of Utrecht (NL).
The complete experimental layout including additional test sites is illustrated in Fig. 1.

4.5 Description of the experiments

4.5.1 Technical set-up

The diffserv test-bed interconnects 14 test sites as illustrated in Fig. 1. As in Phase 1, the wide area network is partially meshed and is based on CBR ATM VPs at 2 Mbps (ATM overhead included). On each VP one PVC at full bandwidth is configured. The PVC is deployed as a point-to-point connection between two remote diffserv capable routers.

[3] The hardware deployed during the second phase of the diffserv test experiments is the same described in [12].

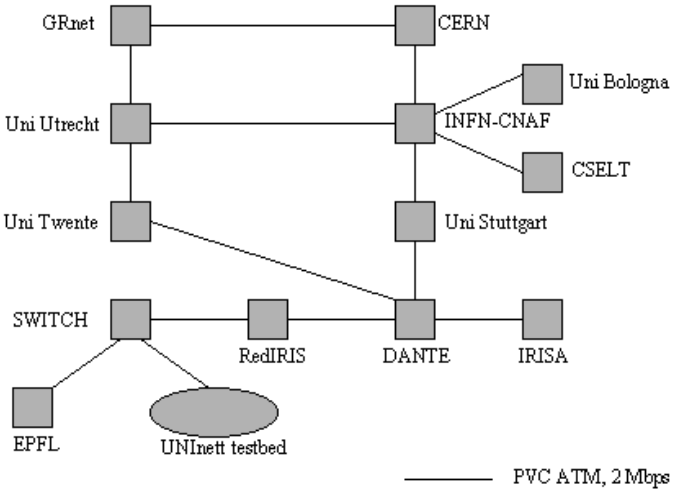


Fig.1: experimental diffserv wide area network

4.5.2 QoS metrics

In order to begin characterising EF behaviour in router implementations, we focused on two key QoS metrics: *One-way packet delay* and *instantaneous packet delay variation*. These are the key parameters for applications that have stringent QoS demands. In order to have reproducible and consistent measurements, we adopt the definitions for these quantities that were developed in the IPPM working group of the IETF [13].

Key to the IPPM definitions is the notion of *wire time*. This notion assumes that the measurement device has an observation post on the IP link: The packet arrives at a particular wire time when the first bit appears at the observation point, and the packet departs from the link at a particular wire time when the last bit of the packet has passed the observation point.

One-way Delay is defined formally in RFC 2679. This metric is measured from the wire time of the packet arriving on the link observed by the sender to the wire time of the last bit of the packet observed by the receiver. The difference of these two values is the one-way delay.

Instantaneous Packet Delay Variation (IPDV) is formally defined by the IPPM working group Draft [15]. It is based on one-way delay measurements and it is defined for (consecutive) pairs of packets. A *singleton* IPDV measurement requires two packets. If we let D_i be the one-way delay of the i^{th} packet, then the IPDV of the packet pair is defined as $D_i - D_{i-1}$.
According to common usage, IPDV-jitter is computed according to the following formula:

$$IPDV\text{-}jitter = |IPDV| \quad (1)$$

In our tests we assume that the drift of the sender clock and receiver clock is negligible given the time scales of the tests discussed in this article. In the following we will refer to IPDV-jitter simply with *ipdv*.
It is important to note that while one-way-delay requires clocks to be synchronized or at least the offset and drift to be known so that the times can be corrected, the computation of IPDV cancels the offset since it is the difference of two time intervals. If the clocks do not drift significantly in the time between the two time interval measurements, no correction is needed.

Maximum Burstiness: we define maximum burstiness the maximum number of packets instantaneously stored in a queue. Maximum burstiness can be expressed in packets or bytes.

Packet-loss Percentage: the percentage of packets lost during the whole duration of a stream.

4.5.3 Methodology

In this section we describe the measurement methodology deployed for the tests described in the following sections.

Specialized equipment for traffic generation - the SmartBits 200 by Netcom Systems (firmware v.6.50) - was deployed as single measurement point supporting packet time stamping in hardware and providing a precision of 100 nsec. It was equipped with two ML-7710 10/100 Ethernet interfaces so that Expedited Forwarding traffic could be originated from a given network card and received by the second one. In this way precise one-way delay measures can be gathered, since they are not affected by clock synchronization errors. This is the EF measurement approach adopted for all the experiments presented in this paper.

One-way delay computation is derived from cut-through latency measures collected through the application SmartWindow (v. 6.53) according to RFC 1242 [16]: it can be computed from cut-through latency by adding the transmission time of the packet, which is constant for a given packet size, as explained in Fig. 2.

In order to analyze the time interaction between best-effort and expedited forwarding traffic even in the presence of nodal congestion, constant bit rate UDP unidirectional streams were generated through the application called *mgen 3.1* [17]. While EF traffic occupies a relatively small portion of line capacity, best-effort streams were needed to add background traffic.

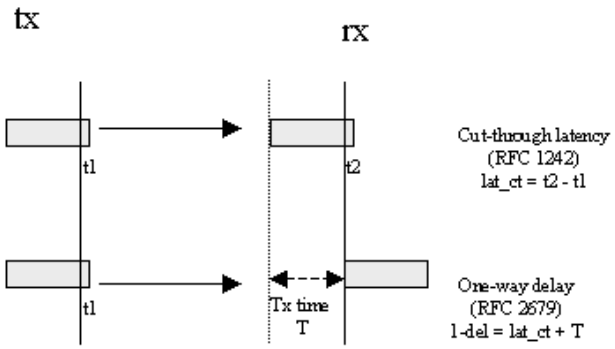


Fig. 2: relationship between cut-through latency and one-way delay.

The *maximum burstiness* is evaluated according to the following methodology. For a given test session in each router on the data path the occurrence of tail drop in the EF queue is monitored. The length *Q* of the priority queue is progressively increased until no packet loss is observed. The maximum burstiness is assumed to be equal to *Q* if during a time interval of the order of magnitude of a test session the following conditions apply: when the queue size is *Q-1* tail drop occurs, while

[4] Packet loss is relevant to burst measurement when due to tail drop.

for a queue size *Q* no packet loss[11] is observed.
In our tests the burst size in bytes can be easily derived from the equivalent metric expressed in packets as for each test the EF packet size is constant and known. Measurement is applied to a single stream, which is called the reference stream.

WRED performance is estimated by computing the overall performance of a class, i.e. the aggregate throughput of streams belonging to that class. Since WRED experiments are based on TCP traffic, the aggregate throughput offers an indirect estimation of the packet-loss per class.

4.5.4 Planned timetable and work items

A test extension is proposed in order to complete the test programme, in particular to develop the following work items:

- Comparison of Priority Queuing and WFQ performance for WFQ systems with 3 or more queues;
- Validation of the EF implementation in the wide area through the deployment of real delay and jitter-sensitive applications;
- WRED configuration and tuning for the support of different AF-based services;
- Analysis and test of several colour markers when applied to AF;
- Performance analysis of classification, marking, policing and scheduling when applied at full line rate (Fast Ethernet and OC-3c);
- Interoperability testing;
- Configuration and analysis of different approached for service validation and monitoring;
- QoS support in the MPLS architecture;
- QoS policy control applied to a intra- and inter-domain diffserv network set-up.

4.6 Results of the experiments

Paragraph 4.6.1 analyses the impact of a FIFO transmission queue when coupled with a WFQ- or PQ-based queuing system. The performance of WFQ and PQ is addressed in paragraphs from 4.6.2 to 4.6.4: PQ and WFQ are tested in different traffic scenarios to identify the most suitable queuing approach for the provisioning of delay, jitter and packet loss guarantees. In what follows we use the acronym EF to refer to delay and jitter-sensitive traffic that is subject to preferential treatment in our experiments and to which measurement is applied.
The end-to-end performance is analysed in two different scenarios:

- single-hop connections: base-line testing is performed so that QoS features are enabled just in the ingress router;
- multi-hop connections: multiple congestion and aggregation points are introduced in the network to analyse the performance of a single micro-flow within a behaviour aggregate when packets are repeatedly subject to scheduling in the network.

The problem of the departure rate configuration is developed in Paragraph 4.6.3, while in Paragraph 4.6.4 we analyse the PQ algorithm performance as a function of the background traffic.
In Paragraph 4.6.5 we focus on the problem of end-to-end performance in presence of aggregation in the wide are and in Paragraphs 4.6.6 the performance of WFQ and PQ are compared.
Finally, in Paragraph 4.6.7 we summarise the main achievement of WRED experiments.
For more detailed survey of the above-mentioned tests, refer to [26, 27].

4.6.1 Transmission queue

Often, in the design of real systems, additional stages of buffering are required after scheduling decisions are made as illustrated in Fig. 3. A typical instance of this might be where scheduling decisions are made on a card or board that is separated from the line adapter by a bus or switching fabric. There may, of course, be other designs that cause an extra buffering stage to be added.

In what follows, we will call this extra buffering stage the *transmission queue* (TX queue) and note that it generally has a simple FIFO queuing discipline. How large this queue is, and how it is handled can have a large effect on the performance on EF traffic, as we show in what follows.

If a given egress interface is congested, i.e. the input rate exceeds the line rate, the TX queue is permanently full. Let's consider two BAs (EF and a BE), such that the EF volume is a minor percentage of the line capacity. Under such assumption whatever EF scheduling algorithm is adopted, when a EF packet is serviced the TX queue is full as it is congested by background traffic: The contribution to EF one-way delay is equal to the time needed to empty the TX queue, thus it is a function of the average background packet size and of the TX queue size itself. For long transmission queues the scheduling system converges a simple FIFO queue and the effect of traffic differentiation gets lost.
One-way delay is a linear function of the TX queue. As a consequence its size has to be limited to a multiple of a few MTU-size packets through explicit configuration where possible. In the following tests the TX queue was set to 5 memory units of 512 bytes each (2560 bytes in total). For a more detailed analysis of the contribution of the TX queue to one-way delay refer to [18].

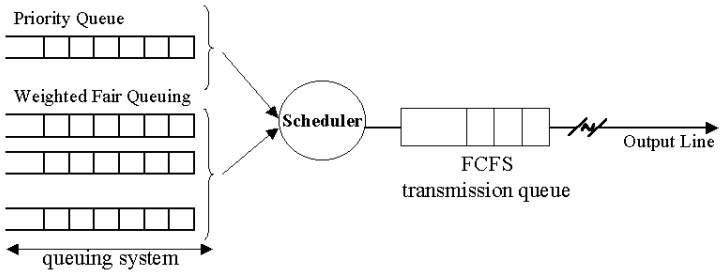


Fig. 3: architecture of the queuing system under analysis

4.6.2 Weighed Fair Queuing service rate

In this test scenario EF and BE traffic crossed a metropolitan area network as illustrated in Fig. 4. Classification, marking, policing and scheduling are tested on the CISCO 7200 in site 1. An ATM VC connection at 2 Mbps links the source to the destination. QoS features are enabled only on the first router on the data path. Both EF and BE traffic are generated by UDP constant bit rate sources. BE background packets are issued by test workstations to congest the egress ATM connection of the first router C7200. The two remaining routers C7200 are non-congested FIFO devices introducing a constant transmission delay that is only a function of the EF packet size.

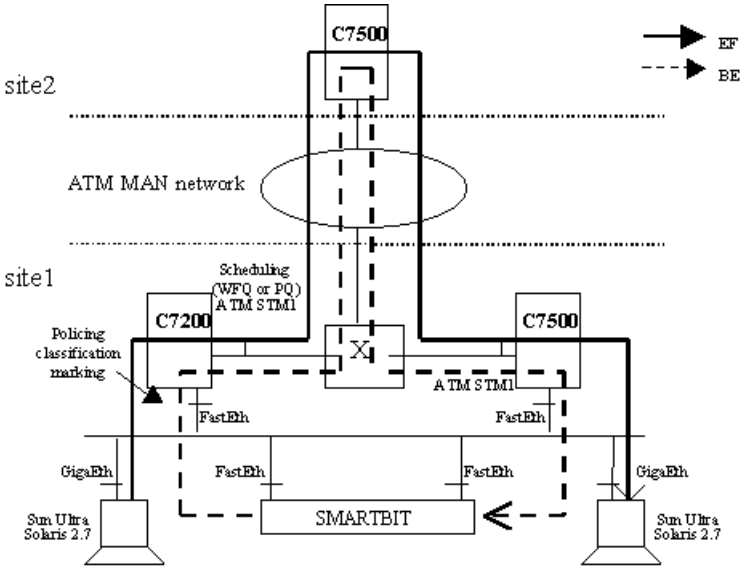


Fig. 4: metropolitan test layout for PQ and WFQ performance testing

Given an EF queue serviced according to the WFQ policy and the estimation of the input EF traffic volume, the correct configuration of the EF queue service rate is important for timely delivery of EF packets. The ratio between the arrival and departure rate is important for the minimisation of the queuing time introduced by the EF queue, as stated in RFC 2598. If the *weight* of the queue is inversely proportional to the service rate assigned to it, then the service time of a given packet *p* is computed according to the formula:

$$service\_time(p) = time\_now + length(p) * weight \quad (2)$$

As expected, the configuration of a large EF queue service rate contributes to a more timely delivery. Simulation studies reported in RFC 2598 show that the over-estimation of the service rate reduces nodal delay and jitter. Our tests confirm the simulation results. In this paragraph we address the tuning of the EF queue service rate.

Given the EF queue service rate S_r (expressed as a percentage of the line rate), the link rate l_r and the EF arrival rate A_r, RFC 2598 defines parameter *Service-To-Arrival-Ratio* (STAR) as:

$$S_r * l_r = A_r * STAR \quad (3)$$

We analysed the impact of the STAR parameter on one-way delay and IPDV for different EF packet sizes. Table 1 summarizes the parameters of the experiment.

EF traffic				BE traffic			Scheduling		TX queue Size
Load (Kbps)	**Frame Size (bytes)**	**Protocol**	**Service rate (Kbps)**	**Load (Kbps)**	**Frame size (bytes)**	**Protocol**	**Type**	**EF queue size (packet)**	**(Particles)**
300	Variable	UDP	Variable	2000	1000	UDP	WFQ	10	5

TABLE 1: test parameters for the analysis of the WFQ service rate

In this test STAR varies in the range [1-5].
As a comment, it is important to note that STAR is simply the inverse of ρ, the ratio of the arrival rate to the service rate of an arbitrary single-server queuing system. In this case, we view the WFQ class proportion of the line capacity as the service rate. For stability, ρ may not be greater than (or equal to) 1, which means that STAR should be greater than 1.
The TX queue is 5 particles (2 BE packets) to minimise delay, according to the results from the previous paragraph.

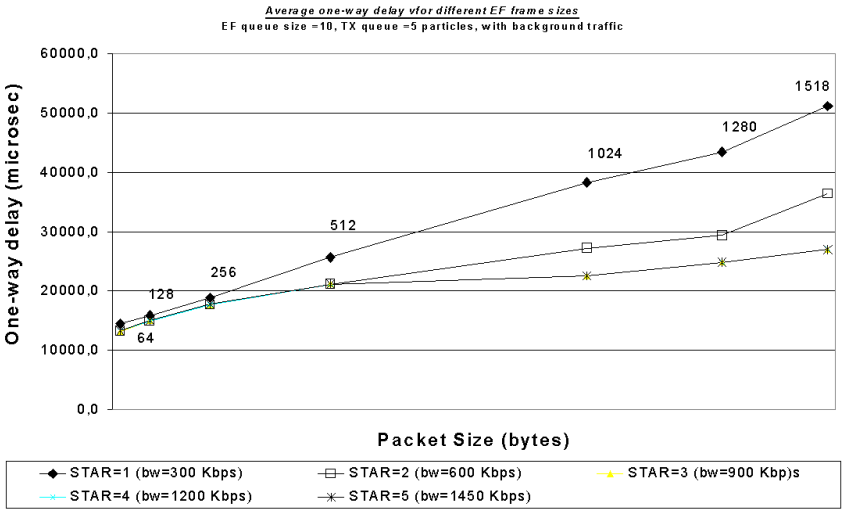


Fig. 5: relationship between EF service rate and average one-way vs EF packet sizes.

Over-provisioning of the service rate can significantly reduce one-way delay, in particular for large average EF packet sizes: In our test the gain of over-provisioning with EF frame sizes of 1500 bytes is up to 25 msec. For smaller EF packets the gain decreases since with WFQ the packet service time is directly proportional to the datagram length and in this case WFQ asymptotically converges to a priority queuing system.
As Fig. 5 shows, STAR values greater than or equal to 4 do not bring any additional improvement, since for an over-provisioning degree equal to 4 the EF queue is already empty and WFQ is equivalent to a priority queuing policy, according to which PQ packets have higher precedence than any other packet in the queuing system.
Increasing the EF service rate does not reduce IPDV: For any STAR value IPDV oscillates in the same range, whose upper bound corresponds to the transmission time of a best-effort packet or to a integer multiple of it.

4.6.3 Priority Queuing

In this section we focus on one-way delay with Priority Queuing. We see how its performance can be affected by the BE traffic profile, in particular by the packet size. For a given BE profile we derive the inverse delay probability function, so that the probability that delay is larger than a given values can be determined. The probabilistic estimation of the end-to-end delay can be useful at the application layer for playback buffer dimensioning.

The test characteristics are summarized in Table 2. In order to reduce the complexity of our analysis only two BAs run in parallel: a best-effort BA composed of multiple best-effort streams, each issuing data at constant rate, and a single constant bit rate EF flow, serviced by the priority queue.

EF traffic			**BE traffic**		
Load (Kbps)	**Frame Size (bytes)**	**Prot**	**Load (Kbps)**	**Frame size distribution**	**Prot**
300	128, 1024	UDP	> 2000	Deterministic, Geometric, Real	UDP

TABLE 2: test parameters for PQ one-way delay measurement in presence of different background traffic profiles

BE traffic profile
While in an ideal scenario priority traffic should not be affected by the profile of other classes, in practice end-to-end delay depends on the size of packets belonging to other classes for the following two reasons. Firstly, in each buffering stage at each congestion point on the data path the scheduling system (WFQ and PQ) introduces some waiting time due to the ongoing transmission of a packet from another queue. Secondly, the TX queues – if present – store packets coming from different queues.

Deterministic distribution
The BE packet size ($size_{BE}$)[12] varies from test to test in the range [100, 1450] bytes. The analysis was repeated for two EF packet sizes ($size_{EF}$): 128 and 1024 bytes.
One-way delay is proportional to $size_{BE}$ for any $size_{EF}$ (Fig. 6 (a)). For a given $size_{EF}$ both the queuing time due to the TX queue and the queuing delay introduced by the priority queue depend on the average value of $size_{BE}$ for the following two reasons. Firstly, the queue space is often allocated in data units of fixed length. This implies that the instantaneous amount of data stored in the TX queue – and consequently the time to empty that queue – varies with the average background packet size. Secondly, the average time spent in the priority queue (waiting for the completion of the current transmission) is a linear function of the average $size_{BE}$. As Figure 6(a) shows, for a given $size_{EF}$ the difference in one-way delay for 100 byte and 1450 byte BE packets is approximately 13.1 msec.

[5] Four different sources generate BE packets at different rates to avoid the effect of synchronization with EF traffic.

The average IPDV is both a function of the EF and BE packet size (Fig. 6 (b)). For a given $size_{EF}$ it is linearly proportional to ½ $size_{BE}$, since the queuing delay introduced by the priority queue oscillates between the transmission time of the whole BE packet – when the EF packet is stored in the queue at the very beginning of the current transmission – to 0 – when the packet is immediately selected for transmission -.
Given a fixed $size_{BE}$, IPDV *decreases* with $size_{EF}$. If the EF load is constant as in this test, by reducing $size_{EF}$ the EF packet rate increases. As a result, depending on the EF rate the probability that the priority queue builds up and that the TX queue contains a variable mixture of EF and background traffic increases. The increase in average IPDV is a consequence of this factor.

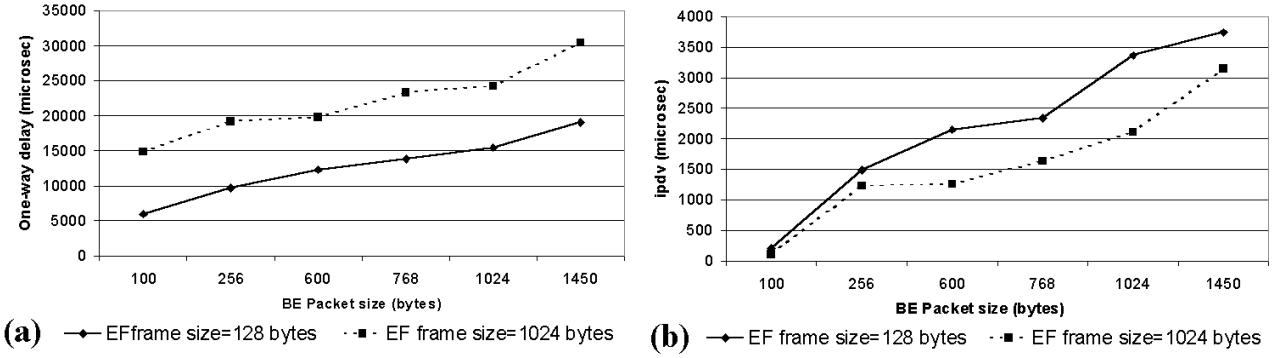


Fig. 6: impact of BE packet size on EF one-way delay (a) and IPDV (b) with PQ

Geometric distribution
Not only the average delay varies with the background traffic profile: also the frequency distribution changes considerably. Fig. 7 plots three distribution curves when the background packet size is set according to a geometric distribution, each curve corresponding to a mean value of the geometric distribution: 128, 512 and 1024 bytes. When the mean value increases the distribution range increases and delay values are more scattered around the average. For a more comprehensive study of the relationship between background traffic and priority traffic refer to [19].

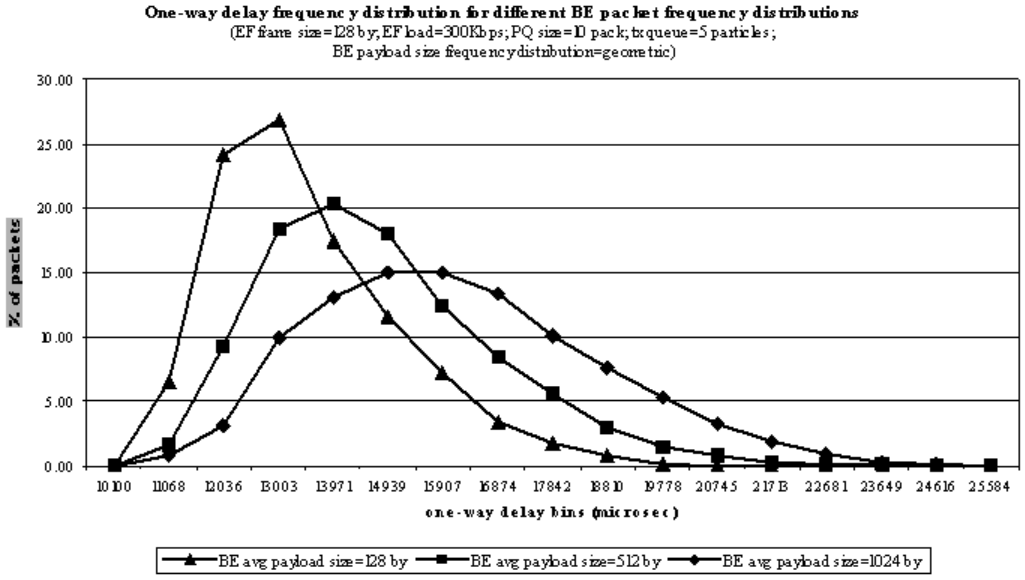


Fig. 7: one-way delay frequency distribution with BE packet size

geometrically distributed in three different cases: with mean equal to 128, 512 and 1024 bytes.

4.6.4 Comparison of WFQ and PQ

Given the network layout of Fig. 5., Table 3 summarises the parameters deployed in this of experiment.

EF traffic (UDP)		BE traffic (UDP)			Scheduling			TX queue Size (particles)
Load (Kbps)	**Frame Size (bytes)[13]**	**Load (Kbps)**	**Frame size (bytes)**	**Num streams**	**Type**	**EF queue size (pack)**	**Departure rate – WFQ – (Kbps)**	5
300	Variable: [64, 1518]	> 2000	variable	4	PQ and WFQ	10	300	

TABLE 3: test parameters for the comparison of EF and BE traffic

Average one-way delay

Priority Queuing performs better than WFQ in terms of one-way delay and its deployment is recommended for the treatment of delay-sensitive traffic. The gain with PQ increases with the packet size: With 64 byte frames14 performance in the two cases is almost equivalent, while the difference becomes relevant for MTU-size packets, as illustrated in Fig. 8: in this test scenario for 1500 byte packets one-way delay with WFQ is approximately double.

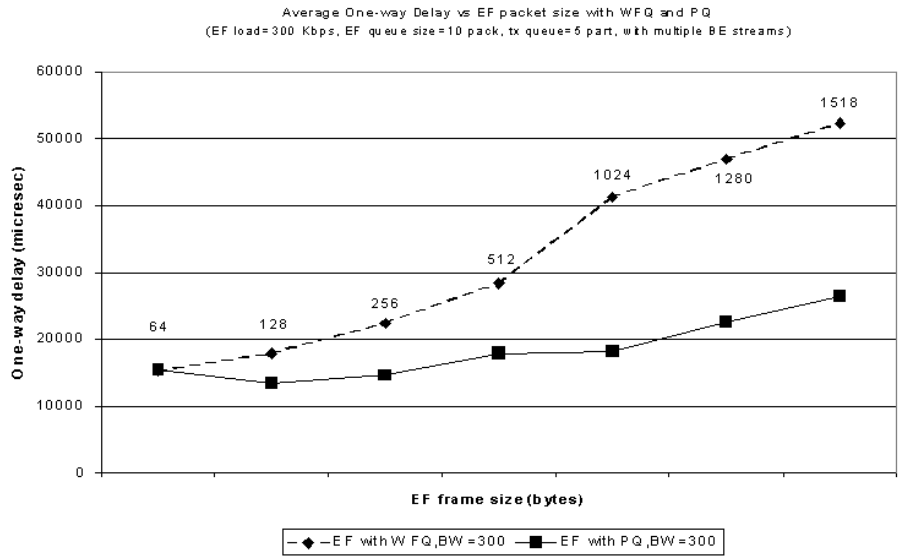


Fig. 8: comparison of average one-way delay with WFQ and PQ for different EF frame packet sizes

The equivalence of PQ and WFQ with small EF frame sizes is explained by the fact that the forwarding time is linearly proportional to the packet size under consideration: short packets experience a smaller queuing delay[15].

With PQ queuing delay is mainly introduced by the ongoing transmission of a lower-priority packet, which needs to be terminated before an EF packet is scheduled for transmission. This delay component varies in the range [0, *tx_time*] where *tx_time* is the transmission time of a lower-priority MTU-size packet. On the other hand, with WFQ an additional delay source has to be taken into

[6] ATM overhead in included.

[7] With *frame size* we refer to the size of the Fast Ethernet frame, which includes the IP payload, the IP protocol overhead and the Fast Ethernet overhead.

[8] In this test the WFQ system is only composed by two queues: the EF queue and the Best-Effort one. The performance of WFQ with a larger number of queues is the subject of future research.

account: the time needed to wait until all the packets with smaller forwarding time in the queuing system are scheduled for transmission.

One-way delay distribution

The performance of PQ depends on the EF frame size itself, as illustrated in Fig. 9(a) and (b), which compare one-way frequency distributions with PQ and WFQ for 128 byte and 1518 byte frames. For each experiment we call *delay unit* the minimum one-way delay experienced either by PQ or WFQ and we adopt it as one-way delay unit. In this test it corresponds to approximately 11 msec.

With 1518 byte frames PQ delay samples are concentrated in a relatively small interval, while with PQ delay varies greatly. This can be explained by the fact that in PQ when a small EF packet arrives, delay can be extremely different depending on the status of the current transmission: If PQ is under service then the only queuing time experienced is the time needed to wait until the EF packets ahead in the queue are transmitted. This time is relatively small since in this case EF packets are short and as a consequence do not experience the queuing delay introduced by a BE packet.

On the other hand, if a different queue is under transmission, the waiting time has two components: the time needed to finish the current non-EF transmission and the time needed to service the EF packets ahead in the PQ queue. If EF packets of MTU size are sent, the inter-packet gap is larger than with 128 byte EF frames[16]. Depending on the EF inter arrival time, it can happen that an EF packet always has to wait for the end of the ongoing BE transmission and never arrives when the PQ queue is under service. In this case the PQ queuing delay is always much higher than with shorter EF packets.

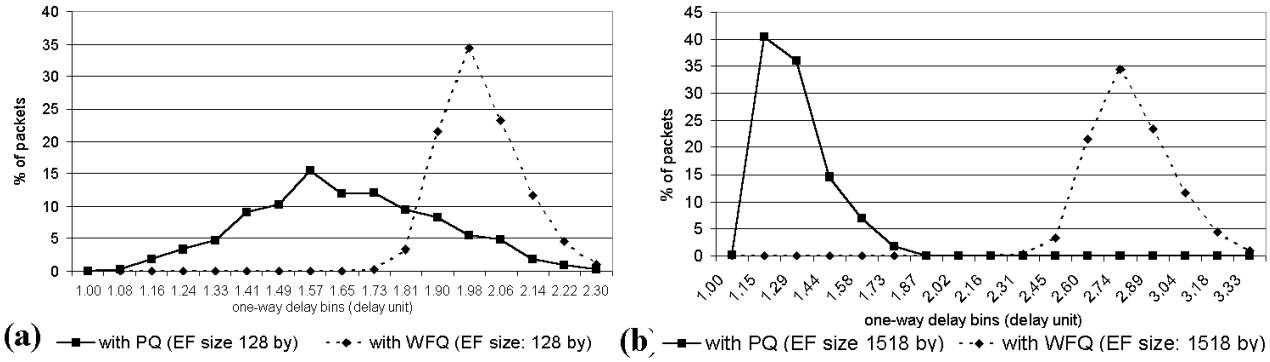


Fig. 9: one-way delay frequency distribution of WFQ and PQ for 128 byte (a) and 1518 byte (b) EF frames

Instantaneous packet delay variation

While one-way delay performance of PQ and WFQ is very different in a given range of packet sizes, the two schedulers have almost the same IPDV performance in terms of both average (Fig. 10) and frequency distribution (Fig. 11) with 128 byte and 1518 byte frames. IPDV is expressed in *transmission units*, where one transmission unit is the transmission time at line rate of the reference EF packet. The comparison of the two algorithms with more than 2 queues is subject of future study.

[9] Since the EF load is constant, by increasing the size of the EF frame we reduce the EF rate, as a consequence the inter-packet gap increases.

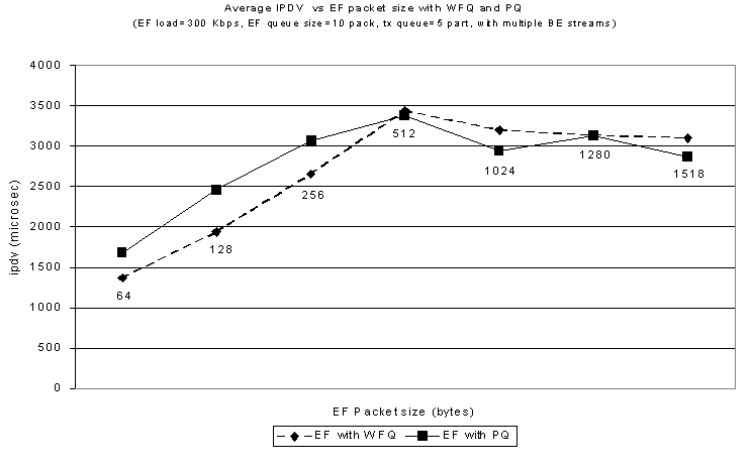


Fig. 10: IPDV sample mean of PQ and WFQ for different EF frame sizes

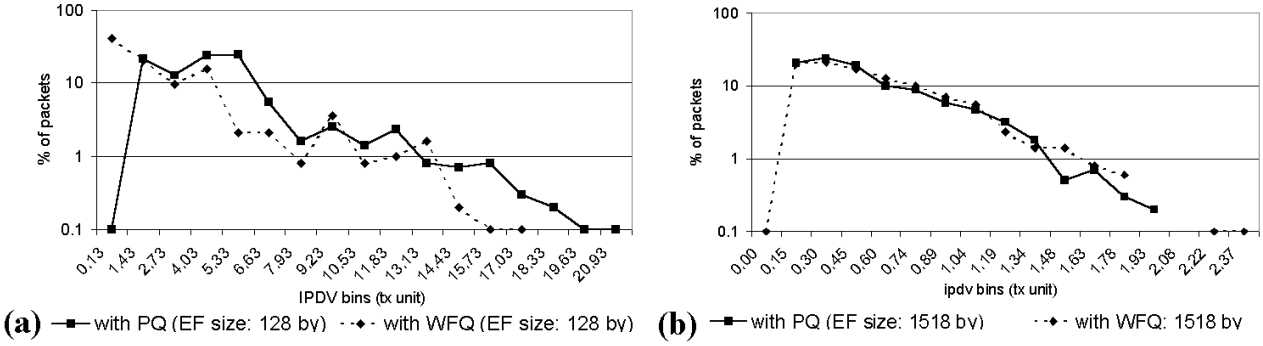


Fig. 11: IPDV frequency distributions of WFQ and PQ (EF frame size = 128 bytes)

4.6.5 PQ under traffic congestion and aggregation (multi-hop scenario)

Aggregation is one of the fundamental properties which characterize the differentiated services architecture and we want to estimate its impact to verify in which cases and to which extent the differentiated services can provide effective end-to-end services as opposed to the integrated services, which can provide per-flow performance guarantees through signaling.

In this experiment multiple EF streams are concurrently injected into the network, so that the end-to-end performance of the reference stream can be evaluated in presence of aggregation. In this case multiple packets of the same class can arrive nearly at the same time from different input interfaces and need to be directed to the same output interface. In this case depending on the speed of the input and output interfaces, the priority queue size can instantaneously hold two or more packets, as indicated in Fig. 12. If this occurs, then *packet clustering* can be observed: Priority queuing transforms a set of well-shaped input CBR streams into a bursty aggregated flow, bursts propagate on the data path to the destination and they increase the probability of packet clustering in the nodes downstream.

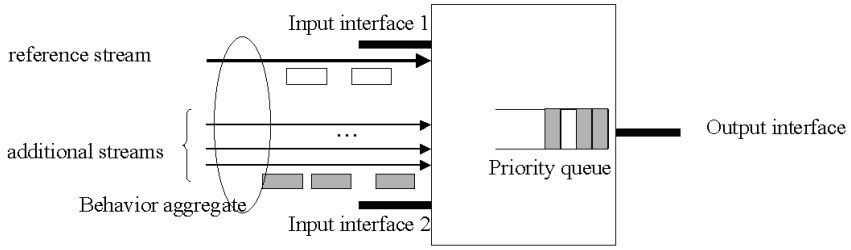


Fig. 12: burstiness with traffic aggregation

Burstiness produced by packet clustering should be avoided since it produces instantaneous non-empty queues – and consequently additional queuing delay – and can cause packet loss if the corresponding queue size is not appropriately set. In what follows we analyse the relationship between burstiness and parameters such as the number of EF streams, the EF load and the EF packet size.

In several network scenarios the instantaneous EF arrival rate can be larger than the maximum departure rate. An example is when the sum of the line rates corresponding to the interfaces from which EF traffic can be received exceeds the line rate of at least one output interface to which EF traffic is sent. This could occur for example when an ingress diffserv node collects EF traffic from multiple high-speed LAN interfaces or when in a egress diffserv node a high speed interface injects multiple EF streams to one or more lower-speed interfaces. As illustrated in Fig. 13, in our test scenario at each aggregation point (four in total) local area interfaces at 10 or 100 Mbps inject a small amount of EF traffic. EF data is then sent to the output interface, an ATM connection at 2 Mbps. EF load varies from test to test in the range [10-50] % of the ATM line rate.

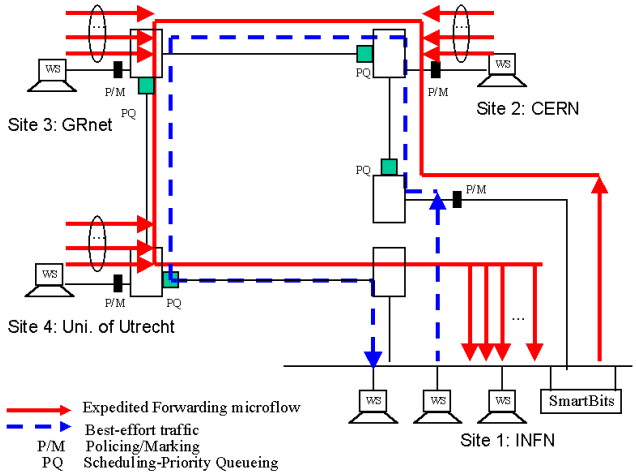


Fig. 13: EF aggregation in a wide area network set-up

Burstiness and EF load

In this test we vary the EF load, i.e. the overall EF rate injected in the network. The EF load varies in the range [10-50]% of the line rate (see Table 4.).

EF (UFP)			BE (UDP)			Scheduling type
Packet size (bytes)	Num of streams	Reference stream load (% of line rate)	Load (Kbps)	Num of streams	Frame size	
64	10 per site (40 in total)	[10, 50]	> 2000	20	Real [0,1500]	PQ

TABLE IV: test parameters for (Burstiness and EF load test)

Fig. 14 plots EF burstiness as a function of the EF load. We see that burstiness is approximately a linear function. In this test the maximum burst size can be up to 35 EF packets.

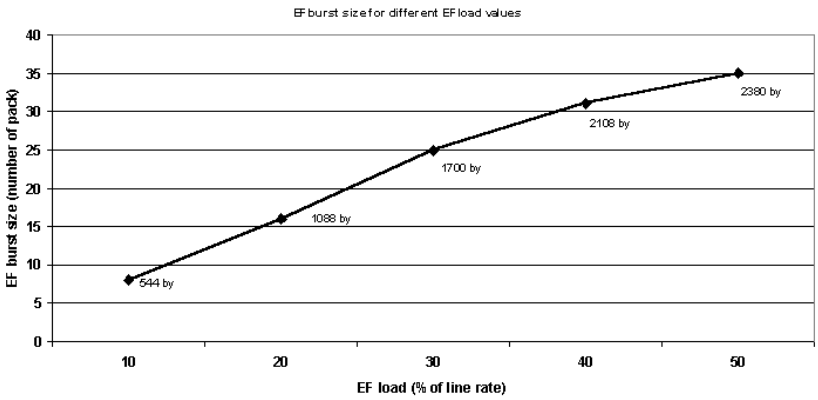


Fig. 14: EF burstiness as a function of the EF load

For each EF load we have set the EF queue length so that the whole maximum burst is completely stored in the EF queue and we have analysed the impact of this setting on delay and IPDV. This means that for each point of the curve a different EF queue size is configured. Fig. 15(a) plots the one-way delay distribution of the reference stream for different load values. Delay is expressed in delay units, i.e. 108.14 msec. The increasing EF queue length that is configured in order to avoid packet loss has a small effect on one-way delay: One-way delay slightly decreases as Figure 3 shows.
In the worst case scenario (burst size equal to 35 packets) the queuing delay introduced by the priority queue is up to 14.84 msec. However, if the presence of long EF bursts is occasional, the effect is not very visible from distributions curves, nevertheless it could have a negative impact on the end-to-end application performance.

Fig. 15(b) plots the IPDV frequency distribution; IPDV is expressed in transmission units (TX units), i.e. 0.424 msec in this test. The graphs show that in presence of increasing EF load values, i.e. of increasing EF burstiness and consequently of longer EF queues, the IPDV distribution gets slightly more spread around the average and the maximum IPDV observed increases as well.

A possible interpretation of the fact that for larger load values packets tend to cluster into longer bursts could be the following. Best-effort packets are transmitted only when the priority queue is empty. This implies that the longer the burst, the greater the number of packets which are transmitted at the same time by the priority queue without being interleaved by BE packets. This also implies that for a greater number of EF packets the delay experienced in a PQ does not considerably differ from the corresponding delay of the previous and subsequent packet in the burst, i.e. IPDV is more constant.

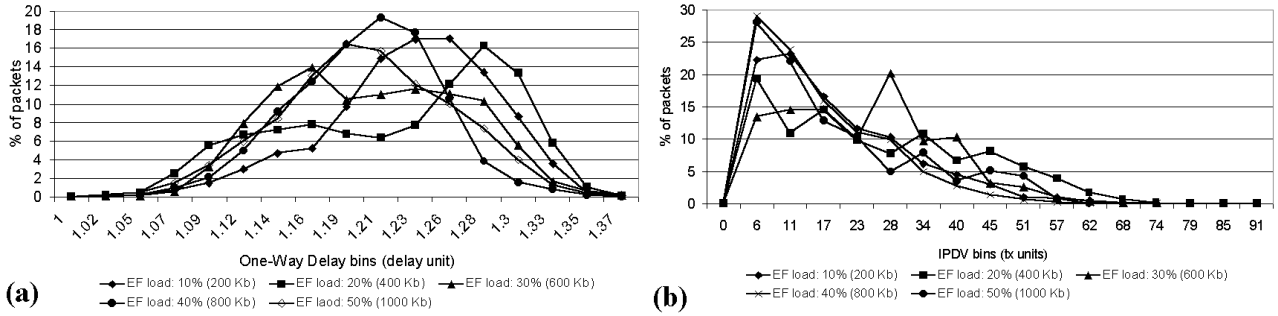


Fig. 15: one-way delay distribution (a) and IPDV distribution (b) for different burst sizes

Burstiness and number of EF streams

In this test we vary the number of EF streams, while we keep the EF load constant and equal to 640 Kbps (32% of the line rate). The test parameters are summarized in Table 5.

EF (UFP)			BE (UDP)			Scheduling type
Packet size (bytes)	Num of streams	Reference stream load (% of line rate)	Load (Kbps)	Num of streams	Frame size	
64	[1, 100]	32	> 2000	20	Real [0,1500]	PQ

TABLE V: test parameters for (Burstiness and number of EF streams)

Burstiness is only moderately influenced by the number of EF streams being part of the class as indicated in Fig. 16: While burstiness greatly increases when the number of streams varies from 1 to 8, from 8 to 100 burstiness reaches a stable point and it oscillates moderately.
While the effect of the number of streams on delay and IPDV is small, the end-to-end performance of a single stream is largely different from the corresponding performance in case of two or more flows.

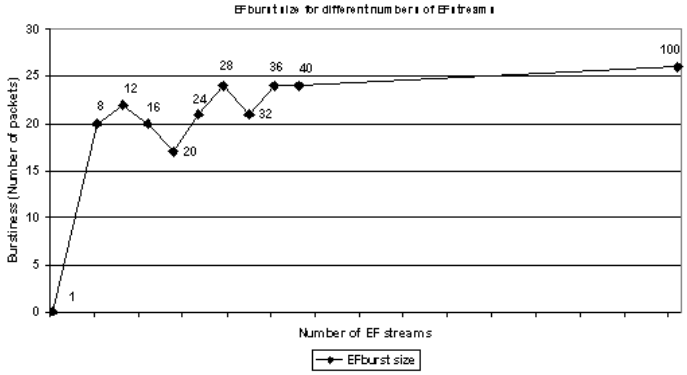


Fig. 16: EF burstiness as a function of the number of EF streams

Burstiness and EF packet size
In this test we kept the overall rate constant and the number of EF streams is also constant and equal to 40, whilst we varied the payload size of the packet in a typical range used for IP telephony, namely: [40, 80, 120, 240] bytes, as summarised in Table 6.

EF (UFP)			BE (UDP)			Scheduling type
Packet payload size (bytes)	Num of streams	Reference stream load (% of line rate)	Load (Kbps)	Num of streams	Frame size	
40, 80, 120, 140	40	32	> 2000	20	Real [0,1500]	PQ

TABLE VI: test parameters for (Burstiness and EF packet size)

Burstiness *in bytes* increases gradually from 1632 - with 40 byte-payload packets – to 1876 bytes – with 240 byte-payload packets . If we look at the same burstiness expressed in packets, we see that it decreases: the reference stream rate is constant, as a consequence if the packet size increases, the packet rate decreases.
Fig. 17(a) and 17(b) plot the one-way delay and IPDV distribution for different EF packet sizes. In this experiment one *delay unit* corresponds to 113.89 msec, while the transmission unit *TX unit* is equal to 0.424 msec.

EF payload size (byte)	40	80	120	240
Avg delay (msec)	132.629	134.834	141.325	144.715
Avg IPdV (msec)	8188	6608	4879	5416

TABLE VII: average one-way delay and jitter for different EF packet sizes

Average one-way delay increases with the EF packet size while the average IPDV decreases (Table 7). One-way delay is distributed in a interval which is inversely proportional to the packet size: With small EF packets the one-way delay range can reach lower values, while the maximum does not greatly change from case to case.
We can conclude that in this test the difference in performance is primarily a function of the number of packets sent per second (and of the background packet size) rather than of the EF packet size itself. In fact, given an overall constant rate, the increase in packet size produces a decrease in the number of packets sent per second and the packet-per-second rate has an effect on the TX queue composition.

For example, for a BE rate of 214 pack/sec and an EF rate of 720 pack/sec (40 byte packets), 1 BE packet is sent only after 3.3 EF packets, while 1 BE packet is sent every 1.1 EF packets if the EF rate is 240 pack/sec (240 byte packets). The resulting composition of the TX queue and the corresponding queuing delay are very different in the two cases (B representing a BE packet, E representing an EF packet)

- EF rate (240 bytes): 240 pack/sec, TX queue = BEBEB
  queuing time = 1.2 * 2 + 4.6 *3 = 16.2 msec
- EF rate (40 bytes): 720 pack/sec, c, TX queue = BEEEB
  queuing time = 0.424 * 3 + 4.6 * 2 = 11.747 msec

The resulting difference in queuing delay (approximately the transmission of time of one background packet) is amplified on the data path in each congested TX queue.

While delay is more limited in case of small EF packets as an effect of the increasing rate of packets per second, IPDV performance improves with large EF packet sizes, as illustrated in Fig. 17 (b)): IPDV is distributed in a larger range for smaller packet sizes. This could be explained by the burst length: The presence of longer bursts with PQ reduces IPDV since packets being part of the same burst experience a similar queuing delay and consequently delay variation in a burst is small.

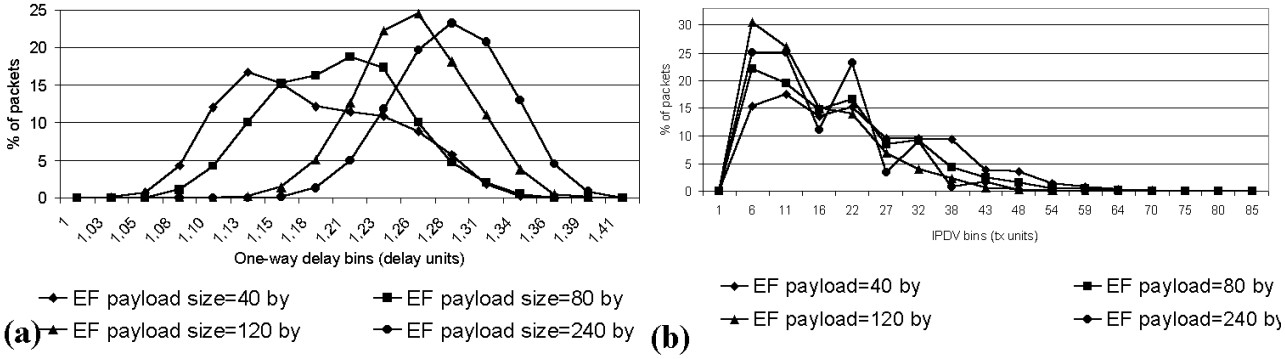


Fig. 17: one-way delay distribution (a) and IPDV frequency distribution (b) in case of burstiness produced for different EF packet-size streams

4.6.6 Comparison of WFQ and PQ in the WAN

The test described in Table 4 was repeated to compare burstiness with PQ and WFQ. Since according to the previous tests EF load is the primary parameter from which burstiness depends, we chose load as test parameter varying in the range [10, 50]% of the line rate. WFQ performance is measured for two different packet sizes: 40 bytes and 512 bytes since for small EF packet sizes the WFQ behaviour is closer to PQ as seen in test 5.1.4.

As Fig. 18 shows, burstiness with WFQ in presence of fairly large EF packet sizes is almost negligible. As expected, with WFQ the formation of large EF bursts is prevented, since unlike PQ in WFQ EF packets are interleaved by BE packets and an EF burst is not transmitted as-it downstream. Limited burstiness is important in order to reduce the end-to-end burstiness.

With shorter packet sizes (e.g. 40 bytes like in this test), WFQ converges to a PQ algorithm, since with WFQ short packets get a preferential treatment (as already stated, the forwarding time computed by WFQ is proportional to the packet size).

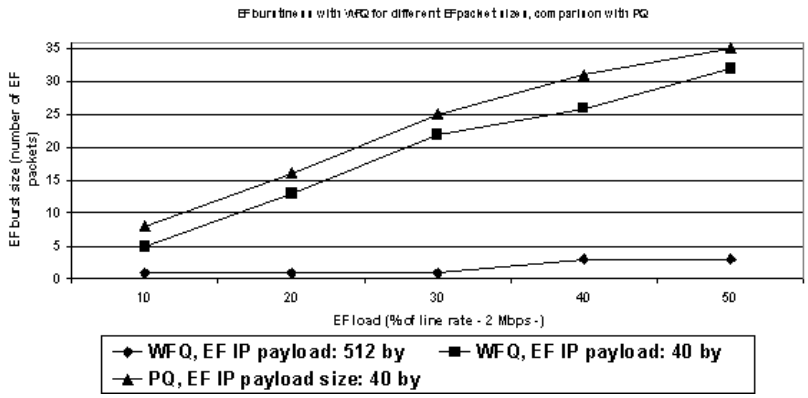


Fig. 18: EF burstiness with PQ and WFQ (40 and 512 byte EF packets)

4.6.7 WRED performance

Weighted RED is an extension of Random Early Detect (RED) [20,21], a mechanism that within a given queue provides differentiation to packets carrying different precedence values.
In WRED at any packet arrival time the *Average Queue Size* (AQS) is estimated. In addition, for each precedence value a queue length *threshold* is defined. The threshold is compared to the AQS: If AQS < threshold, then the packet is placed into the queue, otherwise the packet can be either accepted or dropped depending on the *Drop Probability* associated to the class through configuration.
In the WRED implementation being tested several parameters can be tuned:

- *Minimum Threshold* (min_th): class parameter specifying the minimum queue length (in packets) above which a given packet is evaluated for drop or transmission. If AQS < min_th a given packet is always queued, otherwise the Drop Probability is non-zero and increases linearly with AQS.
- *Maximum Thresh* (max_th): class parameter specifying the queue size limit (in packets) such that if AQS > max_th all the packets in the class are discarded.
- *Max. Drop Probability* (D_p): maximum Drop Probability value applied when AQS is equal to max_th. For each class a different drop procability can be configured.
- *Exponential weighting factor* (*W*): W is a system parameter used to compute the AQS. AQS is an exponential weighted moving average of the instantaneous queue size and is computed according to the following formula:

$$W'=1/2^{W}, \quad AQS_i=(1-W') * AQS_{i-1} + W'* Q\_size \qquad (4)$$

Parameter W' varies in the range [0-1] and defines the balance between the instantaneous queue size Q_size and the previously computed average AQS_{i-1}. If W' is close to 1, then WRED is characterised by a better burst tolerance, on the other hand, if W' is close to 0 the reaction of WRED to congestion is slower.

Congestion avoidance and precedence-based differentiation are the two main functions of WRED. However, through the setting of max_th RED can also be deployed to set an upper limit to the queuing delay experienced by all packets stored in a given queue: If we want that the maximum delay introduced by a given queue is limited to D_{max}, then max_th can be set according to the following formula:

$$max\_th = D_{max} * N \qquad \text{where} \quad N = L / (MTU\_size * 8) \qquad (5)$$

i.e. *N* is the number of MTU-size packets that can be sent in 1 sec at line rate L. min_th can be tentatively set by deriving its value from max_th according to the following rule:

min_th = ½ of max_th. (6)

Test set-up

A point-to-point ATM CBR connection at 3.5 Mbps was deployed and WRED has been enabled on a router Cisco7500 running IOS 12.0(5)XE3 and mounting a VIP2-50 and ATM Deluxe PA-A3-OC3.

WFQ is deployed at the egress interface and two queues are configured on the output interface: a best-effort queue for UDP traffic, which was used to congest the line, and a second queue for TCP traffic (see Appendix B). 25% of the reserved bandwidth is allocated to UDP, while the remaining to TCP streams, which were generated by a specialized traffic generator named *Chariot*. For each class three or more TCP streams are run in parallel, the number varied in the range: [3, 5, 10, 15, 20]. Traffic parameters are summarized in Table 8.

TCP				UDP	Scheduling (WFQ)	
Number of flows	**prec**	**Guaranteed bandwidth**	**RTT (msec)**	**Guaranteed bandwidth**	**Num queues**	**Frame size**
Variable: 3, 5, 10, 15, 20	3, 4, 5	75%	8	25 %	2	Real [0,1500]

TABLE 8: WRED test configuration

TCP traffic is partitioned into three classes, each identified by a precedence value according to the following mapping:

- AF11 → precedence 5
- AF12 → precedence 4
- AF13 → precedence 3

Marking is such that all the packets belonging to the same flow are identified by the same priority[17].

Minimum threshold *min_th*

The effect of min_th on traffic differentiation was studied by varying min_th in the range [3-100], where min_th is expressed in packets. Even though parameters min_th and max_th are the same for each class, traffic differentiation is achieved through the assignment of a different drop probability to each class as indicated in TABLE 9.

Min_th	Max_th	D_p AF11	D_p AF12	D_p AF13	Streams per class
variable	2 * min_th	10%	20%	50 %	Same number per class (3)

TABLE 9: WRED parameters (min_th)

Experiments confirm that in case of equal traffic load per class performance is inversely proportional to its drop probability: AF11 (D_p= 10%) gets 60% of the bandwidth share, AF12 (D_p = 20%) gets30% while AF13 (D_p = 50%) gets the remaining 10%.

Such a balanced bandwidth distribution is achieved if min_th varies in the range [5, 30] packets. In fact, for very small min_th values, TCP is not able to reach stability since packets are dropped too soon, or, on the other hand, for very large min_th values WRED reacts to congestion too late, so that packet loss in *each* class cannot be avoided.

Traffic load per class

The relationship between class performance (aggregate throughput) and traffic load per class is investigated in this experiment. We have measured the performance of class AF11 when streams are distributed among the classes according to what described in Table 10: for each test only one class injects up to 20 streams, while the other two classes constantly inject only 3 streams18.

[10] This kind of marking was tested for experimental purposes only. Several packet marking algorithms can be devised depending on the service to be supported and this is subject of future research. For example in each stream packets can be marked with a precedence of DSCP in a range of values depending on the instantaneous rate (RFC 2698) or burst size (RFC 2697).

[11] In this test RTT is the same for each stream. However, class performance with TCP can be greatly influenced by RTT. The relationship between WRED and RTT is subject of future research.

Min_th	Max_th	D_p AF11	D_p AF12	D_p AF13	Streams per class (AF11, AF12, AF13)
15	30	10%	20%	50 %	Case 1. (20, 3, 3) Case 2. (3, 20, 3) Case 3. (3, 3, 20)

TABLE X: WRED parameters (traffic load)

Results show that the AF11 performance varies significantly from test to test: from 60% of capacity share (the bandwidth share achieved in case of equal load per class), AF11 can get up to 90% of the whole aggregate throughput in case 1, 45% in case 2 and only 20% in case 3. In general, we can say that WRED cannot guarantee a fair throughput distribution among the classes if traffic differentiation is only based on the drop probability and load in terms of number of flows is not homogeneously distributed among the classes. Nevertheless, results show that WRED can still provide with some isolation between classes so that the performance of a lightly loaded class is not too penalized by the heavy traffic load in other classes. In a best-effort scenario without WRED in case of unbalanced traffic distribution aggregate TCP throughput is merely proportional to the number of running streams.

4.7 Summary of test results

The main achievements of the experiments conducted so far are summarised in the following list:

- If a given output interface in a diffserv node deploys a transmission queue, then it has to be limited in size through configuration to a few MTU-size packets, since the queuing delay is a linear function of the transmission queue size.
- If priority traffic is subject to WFQ scheduling, then in order to minimise one-way delay, the departure rate has to be overestimated: Experimental results indicate that a Service-To-Arrival-Ratio (STAR) equal to 3 is the optimal configuration. A further increase of parameter STAR does not bring any additional gain.
- The one-way delay performance of traffic subject to PQ scheduling depends on the profile of traffic belonging to other classes serviced by the queuing system: In particular, EF one-way delay is a function of the packet size. One-way delay increases linearly with the background packet size and delay is distributed in a range that increases with the background traffic packet size. As a consequence, for a given queuing algorithm a more timely delivery is guaranteed when the average background traffic packet size is small.
- Priority Queuing guarantees a more timely delivery that WFQ: one-way delay is considerably lower that with WFQ in particular for large packets. On the other hand, the IPDV performance of the two algorithms seems to be equivalent in terms of both average and IPDV frequency distribution. Further analysis work has to be conduced on WFQ when more than 2 queues are configured.
- If the arrival rate instantaneously exceeds the departure rate, in case of traffic aggregation PQ transforms a set of CBR well-shaped input streams into a bursty output stream. The end-to-end effect on performance introduced by aggregation with up to 100 streams is acceptable.
- EF classes with limited load, limited aggregation degree and small packet size minimise packet clustering. In particular:
  - Burstiness is a linear function of the class load.
  - The increase in the EF queue size – needed to store traffic bursts – when present, has a minor effect on one-way delay and IPDV. IPDV frequency distribution looks better in presence of bursts, probably because in PQ traffic bursts are never interleaved by packets belonging to other queues and packets within a burst experience similar delay.
  - Burstiness is proportional to the number of EF streams. It increases considerably when the number of streams increases in the range [1-8], but after that it slowly converges to a stable value.
  - Burstiness slightly increases with the packet size of the streams in the class. In addition, also both one-way delay and IPDV increase.

  - In case of aggregation, a limited EF rate expressed in packets per second in combination with larger packet produces better IPDV, while delay increases as a consequence of the TX queue, whose queuing contribution is proportional to the amount of space effectively allocated to datagrams.
- Traffic subject to WFQ is less prone to burstiness than without WFQ, in particular for large packet sizes.
- With WRED:
  - If the minimum and maximum thresholds are the same for each class, then per-class throughput can be derived from the drop probability assigned to each class, since aggregate throughput is inversely proportional to the drop probability.
  - Parameter min_th should not be too low (below 3 packets in our scenario) or too large (40 packets or more) to make sure that WRED allows TCP to reach stability by sending full-size windows without experiencing packet loss and that congestion is detected before the queue starts dropping datagrams for each class. The two extremes of the recommended interval depend on the RTT and the line rates for the data path under consideration.
  - WRED is not a mechanism for the support of bandwidth guarantees, in fact, in case of classes that are loaded by a very different number of streams, lightly loaded classes can suffer from performance penalty even when associated to high priority. Nevertheless, WRED can provide more isolation between them then in a simple best-effort non-WRED capable network.

4.8 Implications for future services

The deployment of Priority Queuing is relevant for the support of delay-sensitive traffic and applications which are packet-loss sensitive, while WFQ is important for the provisioning of bandwidth guarantees to a set of classes, where each class is serviced by a dedicated queue. PQ is the scheduling algorithm recommended for the support of the *Virtual Leased Line* service [28,29]. The service implementation problem was addressed by the task force and initial guidelines have been specified [30].

Through the combination of several QoS features such as marking (e.g. CAR or BGP policy propagation on CISCO routers), policing, queuing and WRED, a large set of complex services can be built. The implementation, performance and manageability of a service for the support of bandwidth management on congested intercontinental connection was discussed [25].

4.9 References

[1] RFC 2475: *An Architecture for Differentiated Services*; S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss
[2] RFC 2597: *Assured Forwarding PHB Group;* J. Heinanen, F. Baker, W. Weiss, J. Wroclawski
[3] RFC 2598: *An Expedited Forwarding PHB,* V. Jacobson, K. Nichols, K. Poduri
[4] *A Two-bit Differentiated Services Architecture for the Internet*; K. Nichols, V. Jacobson, L. Zhang.
[5] RFC 2474: *Definition of the Differentiated Services Field (DS Field) in the Ipv4 and Ipv6 Headers*;
[6] *Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks*; H. Zhang.
[7] *Why WFQ Is Not Good Enough For Integrated Services Networks*; J. C .R. Bennett, H. Zhang.
[8] *The bandwidth guaranteed prioritised queuing and its implementations law,* K.L.E. Global Telecommunications Conference, 1997. GLOBECOM '97., IEEE Volume: 3 , 1997, Page(s) 1445-1449 vol3.
[9] *A comparative study of parallel and sequential priority queue algorithms;*
Robert Rönngren and Rassaul Ayani; ACM trans. Model. Comput. Simul. 7,2 (Apr. 1997), pages 157 – 209.

[10] *Link-sharing and Resource Management Models for Packet Networks*; S. Floyd, V.Jacobson, ACM Transactions on Networking, Vol 3 No. 4, Aug 1995,
[11] *Implementing Real Time Packet Forwarding Policies using Streams*; I. Wakeman, A. Ghosh, J. Crowcroft.
[12] *Differentiated Services Experiment Report,* T.Ferrari (Editor), TF-TANT Task Force, Nov 03, 1999
[13] *IP Performance Metrics*; http://www.ietf.org/html.charters/ippm-charter.html
[14] RFC 2679: *One-way Delay Metric for IPPM*, G. Almes, S. Kalidindi, M. Zekauskas
[15] *Instantaneous Packet Delay Variation Metric for IPPM*, C. Demichelis, P. Chimento; ippm draft, work in progress.
[16] RFC 1242: *Benchmarking Terminology for Network Interconnection Devices*; S. Bradner.
[17] *The Multi-Generator (MGEN) Toolset*, Naval Research Laboratory (NRL), http://manimac.itd.nrl.navy.mil/MGEN/
[18] *A Measurement-based Analysis Of Expedited Forwarding PHB Mechanisms*; T. Ferrari, P. H. Chimento, IWQoS 2000, June 2000; in print.
[19] *Priority Queuing Applied to Expedited Forwarding: a Measurement-Based Analysis*; T. Ferrari, G. Pau, C. Raffaelli, submitted to QofIS 2000, April 2000.
[20] *Random Early detection Gateways for Congestion Avoidance*; S. Floyd, V. Jacobson.
[21] *Evaluation of Bandwidth Assurance Service using RED for Internet Service Differentiation*; H. Kim, W. E .Leland, S. E. Thomson.
[22] *RED behaviour with different packet sizes*; S. De Cnodder, O. Elloumi, K. Pauwels
[23] RFC 2697*: A Single Rate Three Color Maker*, J. Heinanen, R. Guerin
[24] RFC 2698*: A Two Rate Three Color Marked*, J. Heinanen, R. Guerin.
[25] *Support for Basic Bandwidth Management Techniques;* N.Papakostas, D.Kalogeras, D.Matsakis, work in progress.
[26] *TF-TANT: Differentiated Services Testing*; http://www.cnaf.infn.it/~ferrari/tfng/ds/
[27] *TF-TANT: QoS Monitoring Testing;* http://www.cnaf.infn.it/~ferrari/tfng/qosmon/
[28] *A Two-bit Differentiated Services Architecture for the Internet;* K.Nichols, V.Jacobson, L.Zhang.
[29] *QBONE Architecture v.1.0;* Internet2 QoS Working Group
[30] *The 1-nVLL Virtual Leased Line Service;* TF-TANT task force, work in progress, http://www.cnaf.infn.it/~ferrari/tfng/ds/service/vll/1-nVLL.html

4.10 Appendix

This appendix provides examples of priority queuing (Appendix A), WFQ (Appendix B) and WRED (Appendix C) for routers CISCO according to the syntax in use in IOS 12.0(6.5)T7.

A Priority queuing

Scheduler

```
class-map pre5
  match access-group 105

policy-map pre5-pq
  class pre5
     priority <rate>
   queue-limit <size>
  class class-default
    bandwidth 700
```

Classification, marking and policing

```
interface FastEthernet0/0
 description test LAN
 ip address 192.168.72.1 255.255.255.0 secondary
 no ip directed-broadcast
 rate-limit input access-group 104 640000 100000 100000 conform-action
      set-prec-transmit 5 exceed-action drop
 load-interval 30
 full-duplex
```

Attachment of the scheduler to an interface (output direction)

```
interface ATM1/0.8 point-to-point
   description to CERN (diffserv)
   bandwidth 1900
   Ip address 192.168.60.6 255.255.255.252
   no ip directed-broadcast
   pvc 8/8
      tx-ring-limit 5
      service-policy output pre5-pq
      vbr-nrt 2000 2000 1
      encapsulation aal5mux ip
```

B WFQ

Scheduler

```
class-map pre7
  match access-group 180
class-map pre6
  match access-group 181
class-map pre5
  match access-group 182
class-map pre4
  match access-group 183
class-map pre3
  match access-group 184
class-map pre2
  match access-group 185
class-map pre1
  match access-group 186

policy-map 8q-wfq
```

```
  class pre7
   bandwidth 300
   queue-limit 10
  class pre6
   bandwidth 300
   queue-limit 10
  class pre5
   bandwidth 200
   queue-limit 10
  class pre4
   bandwidth 200
   queue-limit 10
  class pre3
   bandwidth 200
   queue-limit 10
  class pre2
   bandwidth 100
   queue-limit 10
  class pre1
   bandwidth 100
   queue-limit 10
  class class-default
   bandwidth 100
```

Classification, marking and policing

```
interface FastEthernet0/0
 description test LAN
 ip address 192.168.72.1 255.255.255.0 secondary
 ip address 192.168.174.1 255.255.255.0 secondary
 ip address 192.168.184.1 255.255.255.0 secondary
 ip address 192.65.183.129 255.255.255.240 secondary
 ip address 192.168.73.1 255.255.255.0
 no ip directed-broadcast
 rate-limit input access-group 190 296000 8000 16000 conform-action
      set-prec-transmit 7 exceed-action drop
 rate-limit input access-group 191 296000 8000 16000 conform-action
      set-prec-transmit 6 exceed-action transmit
 rate-limit input access-group 192 200000 8000 16000 conform-action
      set-prec-transmit 5 exceed-action transmit
 rate-limit input access-group 193 200000 8000 16000 conform-action
      set-prec-transmit 4 exceed-action transmit
 rate-limit input access-group 194 200000 8000 16000 conform-action
      set-prec-transmit 3 exceed-action transmit
 rate-limit input access-group 195 96000 8000 16000 conform-action
      set-prec-transmit 2 exceed-action transmit
 rate-limit input access-group 196 96000 8000 16000 conform-action
      set-prec-transmit 1 exceed-action transmit
 load-interval 30
 full-duplex

access-list 180 permit ip any any precedence network
access-list 181 permit ip any any precedence internet
access-list 182 permit ip any any precedence critical
access-list 183 permit ip any any precedence flash-override
access-list 184 permit ip any any precedence flash
access-list 185 permit ip any any precedence immediate
access-list 186 permit ip any any precedence priority
access-list 190 permit udp host 192.168.174.3 host 192.168.175.3
access-list 191 permit udp any host 192.168.175.2 range 40003 40004
access-list 192 permit udp any host 192.168.175.2 range 40005 40006
access-list 193 permit udp any host 192.168.175.2 range 40007 40008
```

```
access-list 194 permit udp any host 192.168.175.2 range 40009 40010
access-list 195 permit udp any host 192.168.175.2 range 40011 40012
access-list 196 permit udp any host 192.168.175.2 range 40013 40014
```

C WRED configuration on router C7500

Marking (through a policy)

```
class-map match-all wred-prec5
   match access-group 192
class-map match-all wred-prec4
   match access-group 191
class-map match-all wred-prec3
   match access-group 190
!
policy-map setprec
   class wred-prec3
       set ip precedence 3
   class wred-prec4
       set ip precedence 4
   class wred-prec5
       set ip precedence 5
class class-default
       set ip precedence 0
```

Attachment of marking to an interface (input direction)

```
interface FastEthernet3/1
   ip address 192.168.70.1 255.255.255.0
   service-policy input setprec
```

WRED and WFQ (egress interface)

```
ip cef distributed
!
class-map match-all prec-only
   match access-group 180
!
policy-map wred-out
   class prec-only
       bandwidth 2000
       random-detect
       random-detect precedence 3 15 30 2
       random-detect precedence 4 15 30 5
       random-detect precedence 5 15 30 10
  class class-default
       bandwidth 666
!
interface ATM0/0/0.2 point-to-point
   description VP CSELT <--> INFN-CNAF
   bandwidth 3560
   ip address 192.168.77.22 255.255.255.252
   pvc infn-qos 0/100
   vbr-nrt 3560 3560
   encapsulation aal5mux ip
   service-policy output wred-out
```

Classification

```
access-list 180 permit ip any any precedence flash
access-list 180 permit ip any any precedence flash-override
access-list 180 permit ip any any precedence critical
!
access-list 190 permit tcp host 192.168.70.2 range 30000 31000 host
```

```
192.168.73.3
access-list 191 permit tcp host 192.168.70.2 range 40000 41000 host
192.168.73.3
access-list 192 permit tcp host 192.168.70.2 range 50000 51000 host
192.168.73.3
```

5 MPLS

Experiment leaders: Herve' Prigent – CRIHAN, Agnes Pouele - DANTE

Participants: CERN, CESNET INFN, GRNET, HEANET, RENATER, SURFnet, REDIRIS, DFN

Keywords: LSP, LSR, VPN, PE, Tag Switching

5.1 Introduction: an overview of MPLS

The primary goal of the MPLS working group [MPLS-1, MPLS-2] is to standardise a base technology that integrates the label switching forwarding paradigm with network layer routing. This base technology (label switching) is expected to improve the price/performance of network layer routing, improve the scalability of the network layer, and provide greater flexibility in the delivery of (new) routing services (by allowing new routing services to be added without a changing the forwarding paradigm).
The initial MPLS effort will be focused on IPv4. However, the core technology will be extendible to multiple network layer protocols (e.g., Ipv6, IPX, Appletalk, CLNP). MPLS is not confined to any specific link layer technology, it can work with any media over which Network Layer packets can be passed between network layer entities.
MPLS provides connection-oriented (label based) switching based on IP routing and control protocols. MPLS may be likened to a 'shim-layer' which is used to provide connection services to IP and which itself makes use of link-layer services from L2 (e.g. PPP, ATM, Ethernet).
MPLS makes use of a routing approach whereby the normal mode of operation is that L3 routing (e.g., existing IP routing protocols and/or new IP routing protocols) is used by all nodes to determine the routed path. MPLS provides a simple "core" set of mechanisms, which can be applied in several ways to provide a rich functionality.

5.1.1 Implications

Some of the key MPLS features are being emphasised by the IETF, vendors and users, and these are driving a major restructuring of IP networks:

- MPLS provides a clean and efficient transition towards Optical Internetworking. MPLS is not dependent of any layer-1 or layer-2 network technique, and therefore can be used to deploy services across heterogeneous infrastructures (ATM, SDH, etc.). With the integration of soft and hard QoS handling mechanisms (DiffServ, Guaranteed Bandwidth), and using "ship in the night" techniques for control planes, MPLS can simplify the use of optical networks by running directly on WDM;

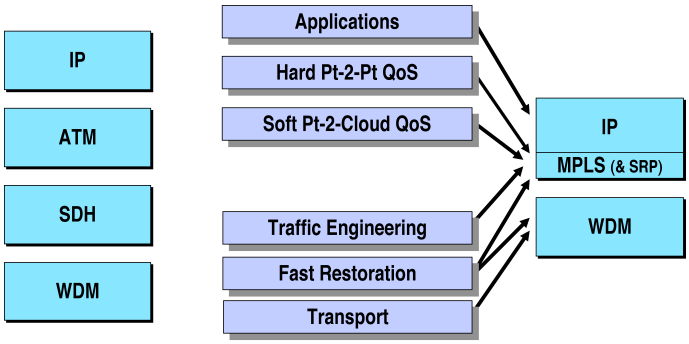


fig.1 – Layer Collapse

- Traffic Engineering is the ability to manage data flows on any underlying network architecture. Today, this is mostly done using ATM but MPLS simplifies network design by implementing Traffic Engineering features independently of layer-2 technologies used;

- New services are needed, such as voice over IP, multicast, web hosting and others. All these services can be deployed using MPLS VPNs over a shared physical infrastructure. Again, because of the many levels of abstraction provided by the use of labels, MPLS can easily be used to provide these services. Moreover, MPLS VPNs are connectionless, and they can be managed centrally and are highly scalable.

5.2 Experiments

5.2.1 Goal of the experiments

The current experiments follow-up on the 1999 TF-TANT experiments reported on in D6.1. The main objectives are to:

- gain experience of the technology
- survey existing implementations
- evaluate advanced features, the stability and performances
- prove its applicability/scalability on an European ATM backbone
- test the interoperability of available solutions

Some of these tests need to be repeated either because results were strange or bad last year, or because software has evolved and is more stable or commercially available. These are:

- MPLS fast re-route
- MPLS VPNs

Other tests planned are:

- MPLS Traffic Engineering
- Diff-Serv mapping on an MPLS network
- Interoperability between software from several vendors
- MPLS over a non ATM network

5.3 Schedule

5.3.1 February 2000 : Preparation of the tests.

- The numerous RFCs with respect to MPLS and vendor documentation were studied. A web page with up-to-date pointers has been created: http://www.crihan.fr/MPLS/mplsdoc.html.
- Because most of the tests will be done on Cisco platforms, Cisco provided technical support and organised a specific testing session in their laboratories in Paris. The aim of the tests was to prepare a full-scale laboratory simulation of the planned TF-TANT MPLS test network and test the features (and their configuration) that would be tested in the WAN. The features that were configured and tested were:
  - MPLS Fast Re-routing Features
  - Traffic Engineering with MPLS
  - MPLS over a non ATM network (a little)
  - MPLS VPNs

A detailed description of the tests is available on the web at the following URL: http://www.crihan.fr/MPLS/ciscolab/ciscolab1.html whilst a summary description of the tests follows in this report.

5.3.2 March 2000

March has been spent trying to Set-up the MPLS test Network and to propose technical information (Addressing Scheme, MPLS configuration of the equipments) to the participating sites.

5.3.3 April-June 2000

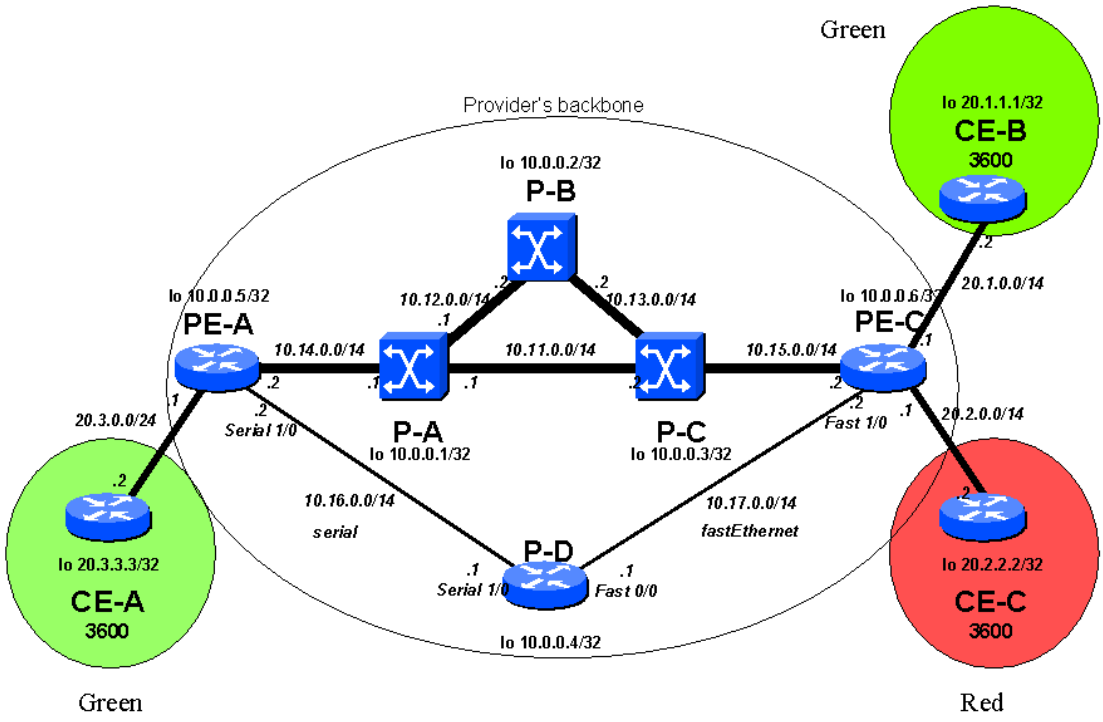
Experiments should be done between April and June. By the end of April, fast-rerouting and some VPN testing has been done.
A presentation of the tests has been done at the TF-TANT meeting in Ljubljana can be accessed at http://www.crihan.fr/MPLS/tests/pres/Ljubljana0400/ljubljana.htm. This too is summarised in the following sections.

5.4 Documentation / day-to-day operations

Information can be found on the web at the following URL: http://www.crihan.fr/MPLS/mpls.html, including technical information on the TF-TANT tests (set-up of the network, test results) and pointers to several MPLS reference pages (IETF, vendor-specific pages, etc.). This page is referenced by the TF-TANT web page at http://www.dante.net/tf-tant/.

5.5 Description of the tests done in Cisco Laboratories

The following network was set-up:



[fig. 2 – Cisco Lab platform]

The core of the Provider's Backbone is built on three Cisco 8510 MSR ATM switches (MPLS Ps) interconnected by ATM OC-3 links. At the edge of the backbone, two Cisco 7200 routers act as PEs and offer a pure IP connection to the users. In addition, an alternative (slower) link is built between the PEs through a 7200 router acting as an MPLS P node. Three clients are connected to the PEs via Ethernet interfaces.

- Routers run IOS 12.0.7 (T)
- Switches run IOS 12.0(4a)W5(11a).

Based on this, the basic configuration for a MPLS backbone was studied, including the building of VPNs and the setting up of LSPs.

5.5.1 Fast reroute on the MPLS backbone

This test was performed in the initial phase of TF-TANT and, as reported in D6.1, the results were not satisfactory in that times in the order of 4 minutes were required to recover from link failures. The same tests were repeated with updated SW and resulted in good performance, in the order of 30 seconds, which corresponds to the OSPF convergence time.

5.5.2 VPNs

In order to understand all mechanisms related to VPNs, two VPNs (Red and Green in Fig. 2) were configured.
Connectivity between the green customers was tested. It was verified that red customers could not reach green customers. Then route target policy which permits customers to be part of multiple VPNs was tested. These tests were an introduction to VPNs, and will be repeated on the MPLS test network to explore all potentiality of VPNs.

5.5.3 Traffic engineering

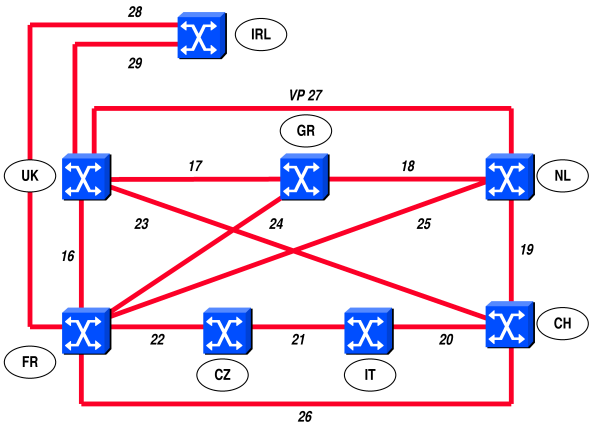
The configuration of the MPLS backbone was changed. IS-IS was used as IGP instead of OSPF. ATM-LSRs were removed and the Cisco 8510 switches were used as simple ATM switches. Routers PE-A and PE-C were connected through a ATM PVC. An LSP between PE-A and PE-C was set-up, traffic was sent in it, and a link failure was created at the ATM level to verify that the tunnel was re-routed. This is a basic test of traffic engineering as it shows that traffic can use an LSP .

5.6 TF-TANT European testbed

The TEN-155 network has been used to deploy an MPLS backbone connecting 8 countries.

- A (partial) mesh of ATM VPs is used between the TEN-155 POPs ;
- No ATM Label Switch Router (LSR) is used. The initial plan was to use one or two ATM LSRs, but production constraints were strong enough to prevent this. However, this is not too important because one of the main goal of MPLS is to be able to deploy an IP backbone without ATM and ATM LSRs do not provide additional features.

The network was fully operational in early April, 2000.



[fig. 3 - map of the network: ATM VPs]

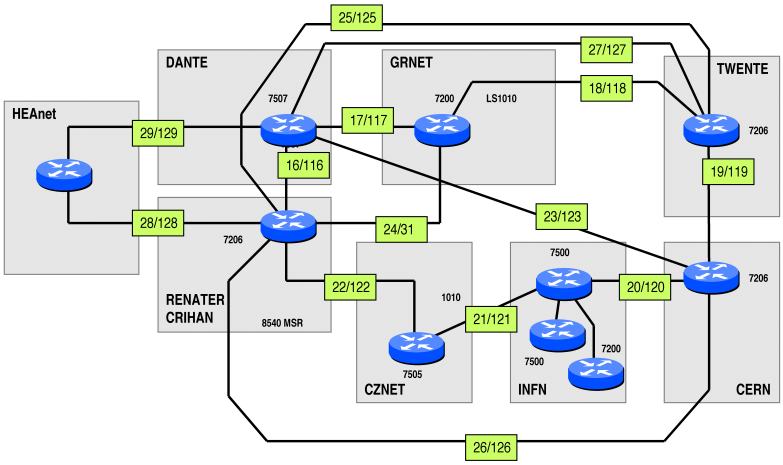


fig. 4 - map of the network: ATM PVCs

5.7 Experiments to date

5.7.1 MPLS fast re-route feature

MPLS fast re-route feature has been tested last year with mixed results. When a network connection was torn down, the convergence delay measured was as long as several minutes. This was not acceptable and was due to a software problem later identified by Cisco.
Therefore the test was repeated in the Cisco laboratories in February to make sure the problem had been corrected, then on the TEN-155 network, with similar results: convergence time was equal or less than 30s, and most of the time around 15s.
To measure the network recovery time, a large amount of ICMP packets (echo request) were sent across the network at a fixed rate. A link was then destroyed (either an optical cable was removed from a switch port or a VP switched down). The recovery time was measured by counting the number of replies that were missing and by manually measuring the elapsed time of the loss of replies. This is acceptable since the recovery time is not small (several seconds) and therefore the error is negligible.

5.7.2 MPLS VPNs

The MPLS VPN network is currently being set up. Several sites have already configured their router so that a client interface belongs to one VPN. It has been verified on these routers that a specific routing table (independent from the regular routing table enabled by default when IP routing is activated on the router) was dedicated to the Virtual Private Network.
The goal of the MPLS evaluation is to check what are the features provided by the Cisco BGP MPLS VPN software. In this implementation, Cisco uses MP BGP extensions to propagate VPN information between peers.

5.7.2.1 Description of MPLS VPNs

A level-3 VPN [MPLS-3, MPLS-4] is a collection of router interfaces sharing the same routing table across an MPLS backbone. VPNs allow for example the overlapping of network addresses between sites belonging to separate VPNs and traffic isolation through MPLS forwarding.

- **P Router**: **Provider Router**: P routers are in the core of the MPLS Cloud and run tag switching (Cisco version of MPLS, released prior MPLS normalisation); they have no knowledge of BGP or VPN routes. Any LSR can be a P router (i.e. ATM LSR.)
- **PE Router: Provider Edge Router:** PE routers connect the clients to the backbone. They receive and hold only routing information about VPNs directly connected and are MP-iBGP fully meshed.
  - Multiple Forwarding/Routing instances can be configured on a PE.
  - VRF (Virtual Routing Forwarding table) contains customers' routes.

  - P and PE routers share a common IGP and a label distribution protocol (Tag Switching).
- **CE Router: Customer Edge Router:** CE routers are client routers connected to Pes. Routes can be announced using a dynamic protocol (RIP, BGP) or statically.
- **Role of MP-BGP extended:**
  - BGP propagates only one route per destination. If two customers use the same address then MP-BGP will distinguish between them.
  - MP-BGP assigns a RD (route distinguisher) in order to propagate all.
  - MP-BGP assigns a Route Target in order for remote PE to insert such routes in the proper VRF routing table.
  - MP-BGP assigns a label for each route.

PE routers store two kinds of labels in their forwarding table:
- Labels learned through TDP and assigned to IGP routes in the global table.
- Labels learned trough MP-BGP and assigned to VPN routes, in VRFs. The MP-BGP label will be the second label in the label stack travelling the core. The label will identify the outgoing interface of the routing table to be used in order to reach the VPN destination.

When a PE router receives a route, it checks his target value. If it matches one VRF then it integrates this route in this VRF.
The label associated to this route is stored and used to send packets towards the destination.
The extended community is used for propagating the Site of Origin (SOO) and Route Target Value.

- **SOO** : Identifies one or more routers where the route has been originated.
- **Route Target** : Selects sites which should receive the route.

5.7.2.2 VPN Tests

Two VPNs will be configured on each participating site. The first one (green) will be common to all the sites. The other one (red or blue depending on the site) will allow us to check the propagation (or the non-propagation) of the routing tables, and the mechanisms that can be activated to filter between VPNs when the exchange of routing information is allowed.
The motivation for deploying VPNs can vary : a network administrator may want to distinguish different type of traffic; he may want to allow access to a specific server to a subset of users or deploy a specific application; an ISP may want to provide interconnection between bank agencies across its backbone and still be able to offer the same service to others clients. At CRIHAN, VPNs are used on the Metropolitan Area Network to be able to offer an alternative access to the Internet (Renater being the default way to do this). Therefore, Cisco's approach for building VPNs is very modular and network managers have many ways to configure their equipment. A comparison of several ways to handle VPN configurations will be done.
Another interesting test would be to check how easy it is to operate an MPLS VPN network (acting as an ISP). Cisco offers a tool that helps building VPNs. Such a tool will be invaluable for ISP and needs to be tested thoroughly, and if possible compared to third-party similar software.

5.7.3 Traffic Engineering

5.7.3.1 Introduction

MPLS traffic engineering software enables an MPLS backbone to replicate and expand upon the traffic engineering capabilities of Layer 2 ATM and Frame Relay networks.
Traffic engineering is essential for service provider and Internet service provider (ISP) backbones.
Both backbones must support a high use of transmission capacity, and the networks must be very resilient, so that they can withstand link or node failures.

5.7.3.2 MPLS traffic engineering:

- Provides an integrated approach to traffic engineering. With MPLS, traffic engineering capabilities are integrated into Layer 3, which optimizes the routing of IP traffic, given the constraints imposed by backbone capacity and topology.
- Routes traffic flows across a network based on the resources the traffic flow requires and the resources available in the network.
- Employs "constraint-based routing," in which the path for a traffic flow is the shortest path that meets the resource requirements (constraints) of the traffic flow. In MPLS traffic engineering, the traffic flow has bandwidth requirements, media requirements, a priority versus other flows, and so on.
- Recovers to link or node failures that change the topology of the backbone by adapting a new set of constraints.
- Replaces the need to manually configure the network devices to set up explicit routes. Instead, you can rely on the MPLS traffic engineering functionality to understand the backbone topology and the automated signaling process.
- Accounts for link bandwidth and for the size of the traffic flow when determining explicit routes across the backbone.
- Has a dynamic adaptation mechanism that enables the backbone to be resilient to failures, even if several primary paths are pre-calculated off-line.

MPLS traffic engineering automatically establishes and maintains a tunnel across the backbone, using RSVP[MPLS-5]. The path used by a given tunnel at any point in time is determined based on the tunnel resource requirements and network resources, such as bandwidth.
Available resources are flooded via extensions to a link-state based Interior Protocol Gateway (IPG).
Tunnel paths are calculated at the tunnel head based on a fit between required and available resources (constraint-based routing). The IGP automatically routes the traffic into these tunnels. Typically, a packet crossing the MPLS traffic engineering backbone travels on a single tunnel that connects the ingress point to the egress point.
MPLS traffic engineering is built on the following IOS mechanisms:

- Label-switched path (LSP) tunnels, which are signaled through RSVP, with traffic engineering extensions. LSP tunnels are represented as IOS tunnel interfaces, have a configured destination, and are unidirectional.
- A link-state IGP (such as IS-IS) with extensions for the global flooding of resource information and extensions for the automatic routing of traffic onto LSP tunnels as appropriate.
- An MPLS traffic engineering path calculation module that determines paths to use for LSP tunnels.
- An MPLS traffic engineering link management module that does link admission and bookkeeping of the resource information to be flooded.
- Label switching forwarding, which provides routers with a Layer 2-like ability to direct traffic across multiple hops as directed by the resource-based routing algorithm.

One approach to engineer a backbone is to define a mesh of tunnels from every ingress device to every egress device. The IGP, operating at an ingress device, determines which traffic should go to which egress device, and steers that traffic into the tunnel from ingress to egress. The MPLS traffic engineering path calculation and signaling modules determine the path taken by the LSP tunnel, subject to resource availability and the dynamic state of the network For each tunnel, counts of packets and bytes sent are kept.
Sometimes, a flow is so large that it cannot fit over a single link, so it cannot be carried by a single tunnel. In this case multiple tunnels between a given ingress and egress can be configured, and the flow is load shared among them.

5.7.3.3 Tests

Replication of the tests done at the Cisco lab in February needs to be done on the TEN-155 network: LSPs will be set-up and fast-restoration features will be tested but this time it will be based on the MPLS platform which will be basically 10 routers fully meshed via LSPs .
The focus will then move to understand which are the best scenarios with TE to avoid bottlenecks, how to load balance traffic when it's needed and/or configure bandwidth guarantees within LSPs. Attention will also be paid to tools for managing MPLS LSPs

5.7.4 QoS Mapping on an MPLS network

Soft and hard QoS handling can be mapped on an MPLS network. Along with Traffic Engineering features and CAR, this can be an alternative to an IP + ATM network infrastructure. Today, DiffServ features are available and are tested by the TF-TANT task force. Hard QoS handling should soon be available on Cisco routers (Guaranteed Bandwidth).
We are currently investigating what features can be tested.

5.7.5 Interoperability

It is not clear yet if interoperability can be tested between several vendors, because of the fast evolution of the market: vendors need to quickly solve customer problems, and tend to propose proprietary solutions that can be used immediately than to spend time on solving interoperability problems. Today, for example, Cisco uses Tag Distribution Protocol instead of Label Distribution Protocol by default, and proposes VPNs using MP BGP extensions and is concentrating on Guaranteed Bandwidth.
We are currently investigating what features can be tested.

5.8 References

[MPLS-1] A Framework for MPLS
http://www.ietf.org/internet-drafts/draft-ietf-mpls-framework-05.txt
Internet draft

[MPLS-2] Multi-Protocol Label Switching Architecture
http://www.ietf.org/internet-drafts/draft-eitf-mpls-arch-06.txt
Internet draft

[MPLS-3] BGP/MPLS VPNs
draft-rosen-rfc2547bis-00.txt (obsoletes RFC2547)
Internet draft

[MPLS-4] Core MPLS IP VPN Architecture
draft-muthukrishnan-mpls-corevpn-arch-00.txt
Internet draft

[MPLS-5] Applicability Statement for extensions to RSVP for LSP tunnels
draft-ietf-mpls-rsvp-tunnel-applicability-00.txt
Internet draft

6 FLOW-BASED MONITORING AND ANALYSIS (FLOMA)

Experiment leader: Simon Leinen – SWITCH

Participants: CERN, CESNET, DANTE, GRNET, SURFnet

Keywords: Flows, accounting, netflow

6.1 Introduction

In this work package, we look at recent developments in traffic accounting, in particular router-based accounting methods, and some possible applications of these mechanisms in the context of backbone IP networks such as National Research Networks (NRNs) or a Trans-European backbone Network (TEN).
The activities in the period covered by this report were mostly related to the exchange of experience between the participating networks, and to an analysis of accounting mechanisms, in particular Cisco's NetFlow accounting, and the tools available to operate in conjunction with them. In addition, an experimental set-up has been configured on one of the workstations at TEN-155's Geneva Point of Presence. The ``samplicator" tool is used to send it a copy of the NetFlow accounting data from the Geneva router (`ch.ten-155.net`), aggregating all traffic for CERN and SWITCH, as well as one of the ``AUCS interconnects" to the commercial part of the European Internet and some transit traffic for other NRNs.

6.2 Motivation

A large part of the costs of operating backbone networks is directly related to the required amounts of bandwidth. Thus it is useful for designers and operators of such networks to have an idea on the nature and amount of traffic transported, in order to anticipate future bandwidth demands and propose bandwidth-saving mechanisms where this seems to be useful. For instance, if the majority of external traffic is related to World-Wide Web (WWW), the promotion of caching proxies promises bandwidth savings and improved performance.
Traditional traffic measurement mechanisms either don't support the necessary level of detail, or are difficult to deploy in a backbone scenario. As an example of the former, the standard per-interface traffic counters, which represent perhaps the most widely used traffic measurement mechanism, provide information on neither the protocols of data being transferred, nor on the source and destination of the traffic, beyond the fact that a particular interface has been traversed in a given direction. Examples of the latter include network probes that must be deployed as new boxes in many places on the network, or router-based accounting mechanisms which significantly reduce the throughput of a router.

6.3 Flow-based Accounting Mechanisms

An important trade-off in accounting mechanisms is between the level of detail available to traffic analysis application, and the cost in terms of amount of accounting data and processing power required for accounting. The mechanisms studied in this work package use a *flow-based* approach to reduce the amount of accounting data without losing too much detail. In this approach, the packets under observation are classified into flows according to some rules. Depending on the actual system, those flows can be related to classical transport-level ``microflows" (either unidirectional or bidirectional), or to ``macroflows" such as the entire traffic between two regions of the network, as defined by e.g. routing table entries or autonomous system (AS) numbers.
This flow-based approach is represented by three particular mechanisms:

- Cisco's *NetFlow*
- Cabletron's *LFAP* (Lightweight Flow Admission Protocol)

- The IETF's *RTFM* (Real-time Flow Measurement) architecture

Table 1 shows a brief comparison of these different approaches. Currently, Cisco NetFlow is the most widely used in ISP contexts, in particular by the NRNs represented in TF-TANT. Therefore it is investigated in much more detail than the others.

	NetFlow	**LFAP**	**RTFM**
flow identification	fixed rules	ACLs	programmable
aggregation	fixed rules	-	-
export model	push	push	pull
export protocol	UDP-based	TCP-based	SNMP
implementation	router sw	router sw/ASIC	PC-based

Table 1. Comparison of Flow-based Accounting Mechanisms

6.3.1 NetFlow

NetFlow was originally introduced in Cisco's IOS as a route caching mode. If NetFlow is used, the router builds a cache in which ``flows" of related incoming packets are associated with forwarding treatment. In addition, a set of accounting data is kept for each flow. NetFlow has been superseded by CEF (Cisco Express Forwarding) as the preferred forwarding mode for high-traffic interfaces, but the two can coexist so that NetFlow accounting can be used with CEF. NetFlow routing has the potential to speed up more complex routing decisions that can be supported by CEF, for example filter lists or policy-based routing, which takes source addresses into account.
A ``flow", in NetFlow terms, is a set of packets traversing the router with the same values for source and destination IP address, protocol, source and destination port numbers (for TCP and UDP), and TOS (Type Of Service) byte.
If ``flow-export" is used, then this accounting data will be exported to a specified receiver using a simple UDP-based protocol. Lost packets can be detected through a sequence number mechanism, but there is no way to recover them through retransmission.

6.3.2 LFAP

RFC 2124 defines version 1.0 of Cabletron's *Lightweight Flow Admission Protocol*. The original usage context for this protocol is with routers that set up per-flow connection state based on requests from the network, and that use an external server to determine whether this set-up should take place. Referring to RAP terminology, this is similar to a Policy Enforcement Point (such as a router) requesting a decision from a Policy Decision Point.
Cabletron has continued to develop LFAP, and the version currently used on SSP routers is 2.x. The protocol is based on TCP and consists of nine message formats (in version 1.0). Most of them are related to policy-based admission control for flow set-up, but the FUN (Flow Update Notification) and their associated responses FUA (Flow Update Acknowledgement) are useful for accounting. It is possible to use only the accounting functionality, leaving the flow set-up decision to the router, i.e. a new flow context is created for every packet that doesn't below to a flow that is already active. Flows are defined according an access list; all fields that are matched by the access list will be used as keys into the flow table.
The LFAP accounting feature implements resiliency by providing for one primary and a few secondary accounting servers. Upon startup, an LFAP-capable router will attempt to connect to the primary server, and if that fails, connects to the secondary servers in the order in which they are specified. Because a reliable transport protocol (TCP) is used, intermittent network problems or restarts of the accounting software often don't result in loss of accounting information. Of course there's a limit on how much data can be buffered in the LFAP accounting sender.

6.3.3 RTFM

Realtime Traffic Flow Measurement is a set of standards defined by the RTFM Working Group in the IETF. It is based on a measurement architecture developed over a period of several years by Nevil

Brownlee at the University of Auckland. In RTFM, *meters* are configured to collect accounting information for traffic flows, where the definition of a traffic flow can be configured using *rulesets* in RTFM's ruleset language. The collected accounting values can be collected by *meter readers*. One possible protocol to access the information in a meter is SNMP, and there's a standard Management Information Base (MIB) defined for this.
An important difference between RTFM and the other accounting mechanisms is that with RTFM, accounting software has to ``pull'' accounting information from a meter, while in the other systems it is the meter (in the router) that ``pushes'' accounting information out to the accounting software.
RTFM rulesets offer a lot of flexibility with respect to the granularity at which data is collected (i.e. the definition of what constitutes a ``flow'') and to the parameters which are observed (packet and byte counts; inter-arrival times etc. This flexibility enables applications that aren't possible with the other accounting methods, but makes the system a bit harder to use (although a basic configuration need not be very complicated) and to implement. In particular, router vendors will be reluctant to integrate a system in routers that offers the possibility to configure complex accounting rules that may slow down packet forwarding.
So the only implementations of the RTFM architecture are either based on general-purpose hardware such as Unix workstations or Intel PCs under MS-DOS, or on low-speed routers. On the other hand, the host-based implementation is available as Free Software and written in an efficient way.
Using host-based RTFM meters can be attractive because this solution doesn't mandate the use of a specific type of router, and the overhead of accounting doesn't impact routing throughput. But in situations where one wants to measure traffic over several links, such as on all external interfaces of a large ISP, maintaining additional equipment for accounting will be expensive. In addition, while host-based systems interface well to shared Ethernets, where they can simply be added to the network using a hub, it is much more complicated to deploy them on switched LANs or typical wide-area connections such as leased lines, SONET/SDH circuits, or ATM circuits.

6.4 NetFlow Accounting

The following section goes into some depth on the workings of Cisco's NetFlow accounting scheme and the implications on the design and deployment of accounting software based on it.

6.4.1 Flow Definition in NetFlow

A ``flow'' in NetFlow is defined by six parameters, roughly corresponding to the keys for a transport-layer session, at least when TCP or UDP is used. The following fields are used as a key identifying a flow:

- source IP address
- destination IP address
- protocol number (e.g. ICMP=1, TCP=6, UDP=17, ...)
- source port number (for UDP and TCP; something else for ICMP)
- destination port number (for UDP and TCP; something else for ICMP)
- TOS (type-of-service) byte

The fields for source and destination port numbers can be overloaded for non-TCP/UDP protocols; for ICMP, they contain the ICMP message type and associated code. They currently don't seem to be used for other protocols.
While a flow is active, the router updates a set of statistics associated with the flow:

- time of first packet seen
- time of last packet seen
- number of packets seen
- number of bytes seen
- inclusive-or of TCP flags seen (for TCP flows)

In addition, some other values that are considered useful for traffic analysis are included in flow-export information for each flow. Those values can all be derived from the source/destination IP address by looking at the routing table.

- SNMP index of input interface
- SNMP index of output interface (0 for dropped/multicast)
- The IP address of the next hop
- prefix length of source address routing entry
- prefix length of destination address routing entry
- source address' AS (neighbour or origin)
- destination address' AS (neighbour or origin)

In principle, all of these values can change during the lifetime of a flow. We did not analyse how such changes are handled; there are basically three possibilities how this could be done:
The values are computed for every packet in a flow, and if there is a change, a new flow is created. This would result in accurate data, but at the expense of more per-packet processing. Also, the number of flows could sharply increase during route flaps.
The values are computed once, when the flow is created, and stored in the flow cache until the flow is exported.
The values are computed when a flow is exported. This would be the most efficient because they wouldn't even have to be stored while the flow is active.

6.4.2 NetFlow Export

The exporting of NetFlow accounting information happens when flow entries are removed from the cache. Entries are removed by cache maintenance according to a set of rules, which can be customised through a few variables (in recent versions).
A flow cache entry is removed:

- when a packet has been seen which seems to terminate the flow; such as a packet containing a TCP segment with the FIN flag set
- when no packet has been seen for *inactive-timeout* seconds
- when the flow has been in the cache for *active-timeout* (default is 30) minutes

In addition, when the flow cache is full and a new flow must be created, old flows will be removed from the cache to make space. The criteria that will select which existing flows will be removed in this case is not clear.
When flow entries are removed from the cache, accounting data is queued for transmission to the `flow-export` destination. Data is actually sent when the queue contains enough entries to fill an accounting datagram (thirty flows for an 1464 octet datagram in NetFlow version 5), or after a maximum age in the order of a second or so. For export, the flow entries are prepended with a header, which contains information such as

- a sequence number to permit the detection of missed accounting data
- the number of flow entries in the datagram
- the uptime of the sending engine
- the absolute time as perceived by the sending engine
- the version number describing the accounting format in use
- an engine type and ID pointing to the particular component of the router that exports the data, if Distributed NetFlow is used.

Currently, only a single UDP transport address (i.e., a destination IP address and a UDP port number) can be specified. Because of the lack of acknowledgements in the protocol, accounting data is lost when the destination address cannot be reached. There are currently several possibilities to work around this problem:

- A multicast address is used as the destination address, and accounting receivers can join the multicast group when they want to receive packets. People who tried such a configuration noticed that it didn't work for them. Additionally, care must be taken to avoid processing the same accounting data on multiple nodes and counting traffic twice (if data from multiple nodes is added together).
- Using a broadcast address would be similar, with the additional restrictions that the accounting receivers would have to live on a single broadcast-capable subnet for this to work.

- A logical (anycast) IP address can be used as the destination address. Accounting receivers are configured to listen on this address, and the IGP will always direct packets to the closest reachable node. Ideally, a node will only announce the anycast address to the IGP when the accounting receiver is actually in a state where it can receive and process packets.

A drawback of this is that some routing protocol support is needed in the machines on which the accounting receivers run, and this has traditionally been a weak point of Unix or NT workstations. With the advent of routing software such as GNU Zebra, GateD or MRT, it could be a sensible approach however.
There are several reasons why flow accounting data might be lost between an exporting router and a post-processor. The most likely is that they are lost along the network path between router and receiver. Therefore it is strongly recommended that this path not be shared with other traffic. Ideally, a separate point-to-point link on spare interfaces should be used for the accounting traffic. Another possible reason for loss of accounting data is resource shortage within the exporting router itself, such as congestion of the control plane between line cards in distributed NetFlow. The router maintains counters for these types of loss events, which can be viewed using the `show ip flow-export` command.

6.4.3 Distributed NetFlow

As an early addition to NetFlow, high-end routers with distributed architecture got the ability to run NetFlow caching and accounting on each intelligent line card, in addition to the main processor. Each line card keeps a flow cache for the traffic entering through interfaces on that card, and each line card exports an independent stream of accounting datagrams, each with its own sequence number space. Post-processors of NetFlow data should take this into account when checking for lost datagrams. In addition, the real-time clocks of the different exporters (line and processor cards) in a router are not always running synchronously. It would simplify things a lot if the clocks on line cards were synchronised from the main processor, which can in turn be synchronised with other routers' clocks using e.g. the Network Time Protocol (NTP).

6.4.4 Router-based Aggregation

In the original version of NetFlow accounting, a router would always send out accounting information for each ``microflow'' according to the NetFlow flow definition. The most widely used post-processors then aggregated this accounting data into two-dimensional matrices indexed by e.g.

- source/destination port
- source/destination IP address
- source/destination network (IP address LOGAND netmask)
- source/destination AS

Recent IOS versions introduced *router-based aggregation*, where the router itself can be configured to aggregate flow statistics into such matrices. The statistics for the resulting aggregated flows are also exported using NetFlow export, but with a new PDU format (NetFlow version 8) which could represent the different aggregation methods.
On routers which aggregate a large number of flows, router-based aggregation significantly reduces the overhead in accounting data and client-side computation, but also in terms of router processing power, because the aggregated tables are much smaller than a full NetFlow table of all active flows, leading to better locality and thus better cache usage. The flow-export function is also significantly cheaper because of the lower volume of exported data. For high-speed router line cards in the Cisco 12000 (GSR) series, Cisco plans to implement only aggregated NetFlow accounting, because the cost of accounting and exporting at the granularity of microflows is prohibitive at line rates in the Gigabit per second range.
While router-based aggregation has very favourable performance characteristics compared with full per-microflow NetFlow accounting, its range of application is severely limited. Everything which requires looking at single microflows, or which requires aggregation methods that do not subsume any of the predefined schemes supported by the router.

Post-processing software has to be extended to deal with router-based aggregation and the different accounting data formats contained in ``NetFlow v8" packets. Recent versions of Cflowd include experimental support for this, as does Cisco's NetFlow Collector.

6.4.5 Handling (Or Ignoring) Flow Start/End Times

A fundamental difficulty in processing flow-based accounting data stems from the fact that these data can refer to packets seen over a relatively long interval of the past. Using the default `active-timeout` in NetFlow, the start and end times can differ by up to thirty minutes.

All NetFlow accounting formats include per-flow fields that indicate the time when the first and last packet of the flow have been seen. These times are expressed in units of milliseconds since the initialisation of the sending engine. The fields are 32-bit unsigned quantities, so they will wrap roughly every seven weeks. In the NetFlow packet header, each engine sends its own perception of absolute time in NTP format (32-bit unsigned values for seconds and residual nanoseconds from an epoch starting on January 1, 1900), as well as the time since its reinitialisation.

As mentioned in the section on Distributed NetFlow, when multiple engines are exporting NetFlow accounting data within a single router, each engine has its own perception of time, and time can advance at different speeds between engines. This makes it quite complicated to derive absolute times from the relative flow start/end timestamps. The router's main processor can keep an accurate real-time clock through synchronisation with e.g. the Network Time Protocol (NTP), but at least in some widely used versions of IOS, the clocks in intelligent line cards are free-running, and can run up significant offsets from the actual time.

Most NetFlow post-processors avoid these difficulties by mostly ignoring flow start/end times, and treating accounting data as if all packets of a flow had been transferred at the same instant, usually the moment when the accounting record has been received or is being processed by the post-processor. This simplification gives good results as long as traffic is largely dominated by short-lived flows (such as typical HTTP transfers), or as long as traffic is averaged over intervals longer than the `active-timeout`. However, if traffic is plotted at time steps of e.g. five minutes, and there are long-lived flows representing significant traffic, those will show up as spikes at half-hour intervals. Examples of such long-lived high-traffic flows are ``newsfeed" NNTP connections between USENET News servers, or tunnels used for multicast-over-unicast or Virtual Private Networks.

One way to reduce measurement errors from such long-lived flows is to reduce the maximum lifetime of flows by setting `active-timeout` to a lower value, such as one minute instead of thirty. This generates additional overhead both for the router and the post-processor, because more flows have to be created, exported, and processed. But in most cases, traffic is dominated by short-lived flows anyway, so that the additional overhead from reducing maximum flow lifetime is relatively small.

If a post-processor does handle start/end times, then the traffic counts for long-lived flows can be distributed over the complete lifetime of the flow. The Fluxoscope system, for example, uses five-minute time slices, and can distribute flow data over up to seven time slices for flows with thirty-minute lifetimes. A drawback of this is that traffic counts in a given time slice can change for a bit more than half an hour after the nominal end of the time slice, which complicates real-time presentation of the data.

Of course, the assumption that traffic is distributed evenly between the start and end times of a flow is also a simplification. Averaging over the lifetime of a flow approximates the actual traffic pattern well as long as the rate of the flow is relatively stable. For USENET News peerings, this is usually the case because the amount of News changes very slowly over time, and data rate is often limited by bottlenecks such as disk speed or RTT/window size, which don't change quickly. However, for high-volume ``background" data transfers that traverse a saturated link, there can be significant changes in data rate, as congestion conditions change over time. A typical transatlantic link of a European NRN can change from 0% loss to 5% loss over a period of half an hour when people start work in the morning (and back to 0% in the evening). When a large file transfer (using TCP with large windows) is active over such a period, its data rate could change from several Megabits to a few tens of kilobits per second.

Even when flow start and end times are used to average traffic counts over flow lifetimes, reducing the maximum flow lifetime can improve both accurateness and timeliness of the measurement.

6.4.6 Heuristics to Determine Application Protocols

The “application mix” on important links or other parts of the network is one of the interesting types of information that can be derived from flow-based accounting data. Classic (unaggregated) NetFlow accounting data includes protocol and port number information for each flow. In many cases, this information is sufficient to deduce with great certainty the type of application to which this traffic is related; for example, SMTP servers always listen on TCP port 25, NNTP servers use TCP port 119, DNS uses both UDP and TCP under port 53, and so on.
However, many other protocols don't have such rigidly defined port numbers. The HyperText Transfer Protocol (HTTP), which today accounts for the largest part of traffic on transatlantic links, uses TCP port 80 by default, but many organisations run HTTP servers on non-standard ports such as 8000, 8080, 8888, or 81. In addition, HTTP over Secure Socket Layer (SSL) uses a different default TCP port number (443). Similarly, Internet Relay Chat (IRC) uses a couple of loosely defined TCP port numbers with the possibility of choosing any other numbers.
Other protocols don't use well-know port numbers at all, or only for an initial handshake, which is then followed by data exchange over new connections that use negotiated port numbers that cannot easily be predicted. One example of this is passive-mode FTP transfers, which don't use the officially assigned FTP-data port (TCP port 20). Because passive-mode transfers are more easily handled by firewalls, they are increasingly used in preference to traditional ``active-mode" transfers. Therefore, if one only looks at TCP ports 20 and 21 (the FTP control port), one is likely to miss the larger part of traffic that actually uses FTP.
The Fluxoscope system has experimented with heuristics to identify such passive-mode FTP transfers. The first method that was implemented used an FTP-control flow between two hosts as a trigger to reclassify *previously* accounted flows with unknown TCP port numbers between the same pair of hosts. While this has been relatively effective, it is somewhat undesirable to have to reclassify traffic that has already been accounted for. After a discussion between the authors of Fluxoscope and FlowScan, an alternate method was implemented, in which FTP-control flows are memorised, and *subsequently* received flows between the same pair of hosts with unknown TCP port numbers are classified as FTP transfers. The memorised FTP-control flows can be discarded as soon as a TCP FIN segment has been observed. This variant has shown to be even more successful at identifying passive-mode FTP transfers.
Similar mechanisms can be used for other applications such as H.323 or streaming media protocols such as RealNetworks', which can represent a high amount of traffic when network users have access to audio or video broadcasts over the network. Implementing such heuristics requires good knowledge of the protocols and their traffic signatures, as well as typical deployment scenarios, in order to make sure that the errors introduced by ``false positives" don't invalidate the increased differentiation provided by the new heuristics.

6.4.7 Overview of Existing NetFlow Post-Processors

6.4.7.1 Cflowd

The Cflowd system has been initially developed at ANS and is now maintained by CAIDA. Its features include good documentation, optimisation for large amounts of accounting data (so that it is useful even for analysing data from core parts of the Internet), and the use of a specialised storage format for the generated data files (the ARTS++ library). The current version of the system is written in C++ and includes experimental support for router-based aggregation and NetFlow v8 format.
Cflowd consists of several components:

- `cflowmuxd` receives NetFlow accounting packets from routers, and makes them available through System V Shared Memory segments.
- `cflowd` takes accounting records out of the Shared Memory buffer, aggregates them into different matrices and stores them into ARTS++ files. This daemon can also dump ``raw" flow accounting data to files for further analysis.
- A collection of smaller programs (`cfdports`, `cfdases`...) access the ARTS++ files and produce reports from the traffic matrices updated by `cflowd`.

6.4.7.2 NetFlow Collector/Network Data Analyzer

These two Cisco products are somewhat similar in capabilities to `cflowd`, but packaged as a commercial product and somewhat more flexible with respect to aggregation methods in the collector. NetFlow Collector runs on Unix workstations (currently supported are Sun's Solaris and Hewlett-Packard's HP-UX systems) and can be configured using filter and aggregation definitions. Network Data Analyzer is a Java application and can in principle be run on any Java-capable platform. The resource requirements for both the Collector and the Analyzer mandate relatively powerful machines however. The programs are licensed on a commercial basis, with evaluation licenses available at no cost.

6.4.7.3 NetFlowMet

Nevil Brownlee has extended his ``NeTraMet" implementation of the IETF RTFM architecture so that it can process Cisco NetFlow accounting streams, calling the resulting system ``NetFlowMet". Using this capability, one could use an existing RTFM meter reader to read accounting data generated using NetFlow-capable routers, specifying aggregation rules using RTFM's ruleset language, although there will be certain restrictions because NetFlow accounting doesn't contain all the information that can be matched by RTFM rules.
This should be an attractive solution for people who have been using NeTraMet and want to move to NetFlow accounting for easier deployment, or for those who want to develop accounting software which conforms to the RTFM standard proposal. NetFlowMet is included in the freely available NeTraMet distribution starting at version 4.2.

6.4.7.4 Fluxoscope

The Fluxoscope system has been written at SWITCH to provide the accounting infrastructure for volume-based charging for transatlantic traffic, as well as for purposes of traffic monitoring and long-term analysis. The main component is the *flow listener*, which is written in Common Lisp and which receives NetFlow accounting data from one or multiple routers, aggregates them according to programmed rules, and maintains traffic matrices for five-minute time slices. The aggregation method used at SWITCH uses three dimensions, namely the large-scale external (peer AS) and internal (connected site) termination point, as well as the ``application protocol" of the flow, which is computed from protocol/port number information according to a set of heuristics.
The listener writes aggregated traffic matrices to ASCII-formatted files every five minutes. Those files can then be accessed using additional utilities that present the data in different forms. A number of presentation methods have been implemented so far:

- A Perl script that generates monthly bills based on the US traffic received by each organisation, taking into account peak and off-peak hours which are charged differently.
- A Common Lisp program periodically feeds a subset of recent data into *RRDtool* databases and generates MRTG-like plots, but with per-application data.
- A CGI script (written in Perl) allows form-based access to the full aggregated traffic matrices through the Web.

6.4.7.5 FlowScan

Written by Dave Plonka at the University of Wisconsin, the FlowScan system allows for the implementation of complex analysis algorithms in Perl, which can be run on raw flow accounting data as generated through (a slightly patched version of) Cflowd. An example application is the detection of Napster-related traffic at a site boundary. This traffic must be identified using relatively complex heuristics, because Napster uses neither well-known port numbers nor well-known servers, but it has a traffic signature which involves connections to a server (from a set of known addresses) followed by TCP connections to other hosts.
The system being written in Perl, it is unclear how well it would scale to high flow rates such as those observed in backbone networks. But it seems to be excellently suited to experimentation with upcoming protocols and novel methods of processing NetFlow data.

6.4.7.6 Others

Several commercial network accounting systems such as those from XACCT, Apogee and Rodopi can process NetFlow data besides other sources. Cabletron's FAS (Flow Accounting Server) 2.0 will handle NetFlow in addition to Cabletron's LFAP accounting format.
Many organisations have written their own tools for the analysis of NetFlow accounting data:
UKERNA uses NetFlow for charging for use of JANET's transatlantic line per organisation. Distinguishing features include a highly scalable and resilient implementation based on a cluster of workstations, and the ``itemisation service". This can be used by organisations to receive a detailed bill based on a list of address prefixes submitted by the organisation, so that charges can be distributed to individual departments, institutes or other cost centres. Unfortunately documentation for the system was not available.
Tom Kosnar has written a comprehensive accounting system for CESNET, which integrates NetFlow accounting data with other sources of information, such as SNMP counters or Cisco ``IP accounting" tables. An SSL-secured Web-based interface provides access to different views on collected data.
At DANTE, the Purgatorio system uses NetFlow to produce a traffic matrix between the different National Research Networks connected to the TEN-155 backbone network, as well as some peering connections to commercial networks. The system is based on Cflowd and uses the AS-to-AS traffic matrix feature.
On the free-software side, there are numerous smaller libraries and utilities that can be used to build NetFlow-based monitoring and traffic analysis applications.

6.4.7.7 Samplicator

An example for a tool that is not really specific to NetFlow, but which can be used to facilitate the development of NetFlow post-processors, is the ``samplicator" program. The samplicator receives datagrams using UDP on a given port, and sends copies of those datagrams to one or multiple other UDP destinations. In addition, a destination can be configured to receive only a sample of received packets, i.e. one in *n* packets. This feature has been used to evaluate the influence of sampling on the accurateness of measurements, in anticipation of future high-speed routers which would only allow accounting based on a sample of the packets transmitted. Another new feature that has been implemented is that the samplicator can now use raw IP sockets to include the source address of the original sender in the copied datagrams, rather than the IP address of the station that the samplicator is running on. This allows it to be used in connection with NetFlow post-processors which talk back to the sending routers, for instance to determine interface set-up using SNMP. This is the case for Fluxoscope and recent versions of Cflowd, for example.

6.5 Conclusions

Flow-based accounting mechanisms permit a vast range of applications in traffic analysis, charging schemes, quality-of-service monitoring, monitoring for denial-of-service attacks and other security issues. There is some tension between the level of detail at which accounting information is generated and accessed, and the performance impact, in particular as one moves up the traffic aggregation scale from campus network through access provider networks to international or core backbone networks. The presence of operators of small and large NRNs, as well as TEN-155 itself, allows for an interesting exchange of ideas. Novel algorithms for analysis of accounting information can be prototyped in the context of a smaller network, and if they prove useful, adapted to backbone networks if feasible.

7 IP OVER ATM

Experiment Leader: Roberto Sabatino – DANTE

Participants: DANTE, KPNQ, Lucent Technologies

Keywords: ATM, DBR, SBR3, SBR2, ABR

7.1 Introduction

TEN-155 offers an IP service based on ATM as the underlying infrastructure. The mapping between IP and ATM is done according to RFC 1483. On TEN-155 a mix of two ATM traffic classes (ATC) are used to ensure efficient backbone usage and fair sharing of capacity between competing flows in cases of congestion. These ATCs are DBR and SBR3, which correspond to CBR and VBR-nrt (with SCR=10, PCR=line rate) respectively. These traffic classes have shown to be suitable for meeting the requirements of TEN-155 and have been thoroughly tested prior to their use on TEN-155.

Other ATCs are available, such as ABR, SBR3 (with SCR >> 0) and SBR2, and they could be suitable for use on TEN-155. Theoretically using SBR3 with SCR>>0 could be advantageous in situations where a minimum guarantee is required as well as the ability to exceed the minimum guarantee. ABR in theory meets these same requirements, except that in this case the ATM layer provides feedback to the end systems for congestion control and therefore avoids cell drops at the ATM level. SBR2 has a similar behaviour to SBR3, except that the end systems rather than the ATM switches tag the cells exceeding SCR.

7.2 Objectives

The objectives of the experiment were to understand exactly how the different ATCs behave in isolated situations and in co-existence and competition with other ATCs. This was intended to enable network managers to configure the most appropriate set of ATCs for their specific needs. On TEN-155 there is a specific interest in verifying if a different mix of ATCs is more suitable than the one currently used. The activity of this experiment therefore focused on the requirements of the TEN-155 service.

7.3 Test Plan

The test plan covered four areas as follows:

- Use of SBR3 wth SCR >>0 for best efforts IP traffic
- Use of SBR2 for best efforts IP traffic
- Use of ABR for best efforts IP traffic
- Fine tuning of SBR3 parameters

The tests were done in co-operation between DANTE, KPNQ and Lucent, and the detailed results are subject to NDA (Non-Disclosure Agreement). However it is possible to present a summary of the results.
It was not possible to test SBR2 and ABR due to the lack of end systems supporting these ATCs during the test plan schedule.
The tests were performed making use of Lucent laboratories in Hilversum, NL and Boston, MA. The tests were restricted to the Ascend CBX500 as these are the switches deployed on TEN-155.

7.3.1

7.3.2 Use of SBR3 with SCR >> 0

For the laboratory tests the following simple test set-up was implemented in the Lucent laboratories in Hilversum.

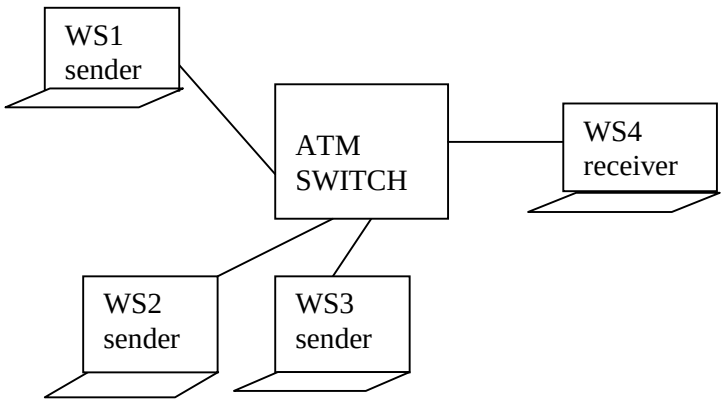


Fig. x.1 Test set-up for SBR3 (SCR >0) testing

This set up allows to create competing streams from WS1, WS2 and WS3 towards WS4, which in turn allowed to test the co-existence of different ATCs in situations of congestion. The ttcp program was used to generate artificial traffic between the workstations. The connections between workstations and ATM switch were STM-1.
The laboratory tests enabled to verify the behaviour of the various ATCs in conditions of extreme load, but with artificial traffic. Field tests on TEN-155 with live traffic are required to verify the behaviour of the ATCs with real traffic patterns. With field tests though it is neither possible nor desirable to artificially create situations of congestion

7.3.2.1 Results

These tests are described in detail in http://www.dante.net/staff/roberto/docs/1999/qtp/RS-99-10.html. ATM PVCs are set up between the sending workstations and the receiving workstation. The 3 PVCs have different SCR, but all have PCR=line rate. The sum of the SCR must not exceed the capacity of the link towards the receiving workstation.

Four sets of tests were performed:

1. all senders sending at a rate less than the corresponding SCR of PVC to receiver;
2. all senders sending at a rate > SCR, but without creating congestion towards receiver;
3. some senders sending at >SCR, others at <SCR, creating congestion towards receiver;
4. all senders sending at a rate > SCR, creating congestion towards the receiver

Tests 1 and 3 are important to verify that the ATC basically works, and that in situations of under utilisation of the PVCs, traffic on the PVCs is not affected by traffic on other PVCs that may be overloaded.

Test 2 is required to verify that in cases of extra capacity being available, this may be used by traffic flows sending at a rate greater than the SCR of the corresponding PVC.

Test 4 is required to demonstrate that in cases of severe congestion, each flow receives at least the SCR of the corresponding PVC to the receiver. As the sum of the SCR is less that line rate, the remaining capacity should be distributed amongst the competing streams.

The tests have outlined that in a situation with no congestion the ATC SBR3 with SCR>>0 behaves well, in that SCR is guaranteed and extra capacity available may be used. The tests have however outlined that the ATC does not work in situations of congestion: the SCRs are in effect not guaranteed.

The details of the results are subject to a non-disclosure agreement with Ascend. While the unexpected behaviour is being investigated, to date this problem has not been solved.

7.4 Fine tuning of SBR parameters

This activity was carried out after severe performance problems with the ATM service were discovered on the operational TEN-155 service: the ATM switches were discarding cells even on non-congested interfaces. Detailed investigation of this problem lead to the conclusion that the FCP (Flow Control Processor) module required a higher degree of parameter tuning than currently used. The parameter tuning depends heavily on the network set-up, the traffic profiles and the different mix of PVC settings used.
A network emulation was set-up in Lucent laboratories in Boston, MA, in December 1999 to simulate the TEN-155 network and identify the required parameter settings for the FCP module. These parameters include:

- the Initial Cell Rate (ICR);
- Rate increase factor, rate decrease factor (RiF, RdF);
- Available cell rate (ACR);
- Buffer discard threshold;
- Buffer congestion threshold

The exact meaning, value and interaction of these parameters are all strictly confidential and subject to NDA. They cannot, therefore, be explained in this document.

The new parameter settings have been applied to TEN-155 and the performance problems related to them have subsequently disappeared.

7.5 Future work and implications for future services

To perform tests with ABR and SBR2 it is necessary to have access to end systems with these capabilities. Tests with ABR have been performed with ATM traffic analysers, but this is viewed as a limited test as these are not systems that users use for real traffic.
If the problems detected with ATC SBR3 with SCR>>0 are resolved, it may be possible to use this ATC on circuits which often experience congestion, in order to guarantee a minimum capacity and the ability to use more in situations of spare capacity being available.
As the planned upgrade of TEN-155 will focus more on IP over SDH and as it is not planned to extend and develop the current use of ATM (except for connections to new countries), no further work is planned for the IP over ATM activity

8 IPv6

Experiment Leaders: Simon Nybroe, Alex van der Plas -Ericsson/Telebit

Participants: ACONET, CERN, CESNET DANTE, G6, JOIN, REDIRIS, SURFnet, SWITCH, Uninett, Univ. of Southampton

Keywords: IPv6, DNS, multihoming, bind

8.1 Objectives of the Experiment

The goal of this experiment is to provide the information needed to implement IPv6 connectivity as a service in the TEN-155 production network. This will provide input to DANTE with regard to a future IPv6 production service and on the same time work as a forum where NRNs (and possibly others) that have or want hands-on experience can share their findings.
Outline Solution
While IPv6 is a new protocol, the steps in providing a backbone service are very similar to those needed to follow in order to provide an IPv4 backbone service. This will hopefully shorten the learning curve and ease the adaptation of IPv6 into DANTE's existing management procedures. After the establishment of the European IPv6 test network, the experiments have been separated into four different areas each covering different aspects of the issues of moving towards an IPv6 infrastructure. The four different areas will be described in more detail in the following section. The testing within a area is performed by subsets of the participants. Testing is performed across the native IPv6 test network built as part of the experiment, locally within one participant's network infrastructure, in a test lab or any combination hereof.

8.2 Description of the Experiments

The experiment has been separated into 4 different areas:
- Interoperability
- DNS
- MultiHoming
- Applications

The task of specifying the tests within one area are currently worked on by the four groups, the following sections will report in the status of three of the four areas. With respect to applications, no significant testing has been carried out yet, therefore this is not reported on. All tests involving Cisco hardware have been performed using pre-production IOS versions.

8.2.1 Interoperability tests (JOIN)

Currently only Ericsson Telebit and Cisco routers have been used for interoperability tests. For the Cisco routers IPv6 support is still in beta, and many of the reported problems may be resolved before the first initial release. Some issued reported here are also documented in the release notes.

While there have been no overall tests of IPv6 basic compliance functionality, the group's attention focused on interoperability problems, which occurred when setting up the dedicated network to support IPv6 testing on TEN-155 (referred to as the QTPv6 network) or during local laboratory tests. The QTPv6 network consists mainly of ATM connections with BGP4+ as the routing protocol, therefor most of the occurred difficulties touch this topics. In local lab tests several other problems concerning hosts and software routers (such as MRTd and zebra) came up. The following section will describe the problems in more detail.

8.2.1.1 Telebit and Cisco

BGP negotiation.
Since IOS 12.0 Cisco does BGP capability negotiation according to draft draft-ietf-idr-bgp4-cap-net-06.txt by default. Telebit does not support this yet, so BGP sessions will not establish. A solution or workaround for this is to disable capability negotiation in the Cisco router with "override-capability-neg" for the IPv6 connection.

BGP connection status change between active and established
The BGP connection between SURFNET and the core router changes status from established to idle and back within minutes. This problem occurs only at SURFNET, other Cisco to core router connections are stable. This behaviour may be caused by different timer values, or more likely by the lack of keep-alive messages. This issue has been forwarded to Cisco but currently the problem is unresolved.

BGP4+ peering over native ATM
It was not possible to establish a BGP4+ peering between Cisco (IOS 11.3 and 12.0beta) and the Telebit routers using a native ATM path. Router-ID and/or next-hop fields got mixed up and were not properly set. This problem is resolved by a bug fix to the Telebit router software.

Exchange of BGP routes
The BGP routing information was not exchanged properly between Cisco and Telebit routers while running over ATM. The problem occurred when the ATM path configuration on the Cisco router was set to "multipoint". The problem was resolved by changing this to "point-to-point".

8.2.1.2 Cisco and Cisco

Route selection between Cisco 2501 and 4500
In a BGP peering between a Cisco 2501 and a Cisco 4500 the BGP message was sent correctly with the correct prefix, but the 4500 still picked the wrong route. This problem is still unresolved.

8.2.1.3 Cisco and MRTd

BGP negotiation
It was not able to establish a BGP peering between Cisco and the MRTd software router due to a bug in refusing to do BGP capability negotiation by MRTd. This problem was resolved by a bug fix to MRTd code as of version 2.1.1a.

Use of RIPng
It was not possible to use the RIPng component of MRTd with Cisco routers. MRTd did not conform to the RFC with respect to hop-count in announcements, and Cisco does strict conformance checking. This problem was resolved by a bug fix to MRTd code. The same problem occurs using the RIPng daemon supplied by Solaris8/PC “early access”. Later versions of the Solaris software have not been tested.

8.2.1.4 Newbridge VIVID network does not support IPv6

Ipv6 will not run properly over a Newbridge VIVID network because its bridges over VLANs have a bug whereby they may drop the first three packets of a data flow (before establishing a cut-through path). This is disastrous for certain IPv6 requirements such as router advertisements. Newbridge have been unable to fix this and has no plans to support IPv6 commercially. A possible workaround is to generate ping flows of more than four packets periodically.

8.2.1.5 Stability

Sending large ping packets caused various problems on different platforms and implementations. The symptoms were ranging from

- Data errors (e.g. Linux fragment out-of-sequence problem)
- Asymmetric ping behaviour (max data block length), depending on which platform the client is used
- Disjoint packet length intervals for which ping6 works
- Crashes on target nodes

Crashes

Especially the last point is most severe. It happens when a Linux host sends large packets over the network. A Cisco router will crash completely. This problem is caused by a reversed sending mechanism of Linux. When Linux sends a large packet over the network it will be correctly split into several fragments, but the fragments are sent in reverse order (last first) and this causes the Cisco router to crash. This problem is forwarded to both Cisco and the Linux development community, but is still unresolved. As fragmentation of packets is handled in a new manner in IPv6 (no router fragmentation, PMTU discovery), this issue needs closer observation.

8.2.2 Multihoming issues for QTPv6

The key property that makes multihoming issues more interesting under IPv6 in comparison to IPv4 is that hosts and interfaces are far more readily able to inherit and utilise multiple network addresses in a dynamic environment under the new Internet Protocol.

Under IPv6, if a host attaches to a network it will see a router advertisement (RA) message from any IPv6 router attached to the local LAN. It will then, under stateless autoconfiguration, assign an IPv6 address to the interface receiving the RA, created from the concatenation of the advertised network prefix and an EUI-64 host part based on the host's MAC address. In the event that two or more distinct RAs are made on the same LAN, a host will become multihomed on two or more networks.

While multihoming may occur through multiple addresses on one interface, it is also possible to have a multihomed scenario with multiple interfaces on a host with one or more addresses per interface. In each case, the host is faced with the choice of which interface to use (if more than one exists) and which source and destination address to use in initiating a connection to a target host (bearing in mind that that host may also itself be multihomed).

8.2.2.1 Reasons for becoming multihomed

There are many reasons why a site may choose to become multihomed, e.g.

- *Network resilience* – if one link fails, the other one can be used
- *Load balancing* – to achieve better average throughput over two service providers
- *Value in services* – selection of a provider on a per-application or per-connection basis (this may become more prevalent when, for example, IPv6 VoIP services are commonplace).

An interesting question here is how likely a requirement it will be to perform multihoming, in particular for network resilience, in an academic environment. It is unusual for an academic site to have both a link to its own academic NRN network and an independent link to a commercial ISP (and some NRNs specifically exclude sites from operating in a way which may leak commercial traffic over an academic network). It may be possible for pairs of academic sites to share connections for resilience, but in that case the network would have to be able to cope with the surge in bandwidth demanded on one link when the other link failed. It is worth noting that these issues are as applicable to IPv4 as IPv6.

8.2.2.2 Dynamic IPv6 networks

One must bear in mind that IPv6 networks have a tendency to be dynamic in terms of the prefixes and interfaces seen on them, for a variety of reasons including:

- *Router renumbering* – a network prefix may be changed at any time
- *Inheriting new prefixes and interfaces* – either via a new prefix being enabled and advertised or a new tunnel connection being established
- *Mobile hosts* – a host may migrate from one network prefix to another
- *Losing prefixes* – a link dies, or a mobile user migrates from a fixed Ethernet subnet to a different WaveLAN subnet when they roam in a building.

Dynamic changes that are unusual in IPv4 are likely to be seen as more "normal" under IPv6. Because the underlying multihomed environment on a network may change with time, robust solutions to the problems posed become more difficult to engineer.

8.2.2.3 Routing efficiency

It is very important for global routing efficiency (for keeping to the IPv6 address aggregation principle) that unaggregated addresses are not leaked out onto the global IPv6 network (though if they were, they would almost certainly be filtered by most backbone routers). Thus any multihoming solutions must operate at most by peering arrangements at a site's parent provider level. For example, while a /48 network may not be propogated to the backbone, it may be exchanged between two parent providers which themselves allocate /48's to their clients.

8.2.2.4 Multihoming environments under QTPv6

The consensus amongst QTPv6 partners was that the most natural way to trial multihoming scenarios was to use two prefixes, the first being the 6bone prefix 3ffe:8030::/28 assigned to the QTPv6 IPv6 test-bed network, the other being the real SubTLA allocated to the participating NRN.

Each partner on the network has a /34 prefix, e.g. Southampton has 3ffe:803c::/34, under which each site has allocated addresses according to its own local address plan. At Southampton we have made those allocations using the Blanchet Internet Draft "A method for flexible IPv6 address assignments" [QTPV6-01] in which the assignments are made from the middle of each subnetwork part.

Such an assignment scheme allows for flexibility in moving the subnetwork boundaries. The /48 subnetwork allocations from under 3ffe:803c::/34 could thus be handled as follows:

```
             ----SLA----------
3ffe:803c:00 00 0000 1000 0000:0  or  3ffe:803c:80::/48
3ffe:803c:00 00 0001 0000 0000:0  or  3ffe:803c:100::/48
3ffe:803c:00 00 0001 1000 0000:0  or  3ffe:803c:180::/48
3ffe:803c:00 00 0000 0110 0000:0  or  3ffe:803c:60::/48
3ffe:803c:00 00 0000 1100 0000:0  or  3ffe:803c:c0::/48
3ffe:803c:00 00 0001 0100 0000:0  or  3ffe:803c:140::/48
3ffe:803c:00 00 0001 1100 0000:0  or  3ffe:803c:1c0::/48
3ffe:803c:00 00 0010 0000 0000:0  or  3ffe:803c:200::/48
etc.
```

For multihoming work, a second network prefix allocation is required. The RIPE NCC has allocated SubTLA status to many of the QTPv6 NRNs. These include:

```
CH-SWITCH-19990903            2001:0620::/35
AT-ACONET-19990920            2001:0628::/35
UK-JANET-19991019             2001:0630::/35
DE-DFN-19991102               2001:0638::/35
NL-SURFNET-19990819           2001:0610::/35
```

```
GR-GRNET-19991208              2001:0648::/35
FR-RENATER-20000321            2001:0660::/35
```

Although the allocation is a /35, the "real" provision is of a /29.

8.2.2.5 Multihoming at Southampton

As an example, Southampton falls under JANET-UK, and has address space under 2001:0630::/35. With network connectivity from JANET and from QTPv6, using a different prefix on each, a multihomed scenario can be created, with some hosts on the site resident in both networks. Similarly, other NRNs are also able to have similar multihomed sites (on the assumption that the network connectivity is distinct for each prefix).

The Southampton network inherits three provider prefixes:

Network	*Trial Prefix*	*Native IPv6 Connectivity*	*Partners*
QTPv6	3ffe:803c:80::/48	512Kbit ATM PVC	QTPv6
Bermuda	2001:0530:1fff::/48	2Mbit ATM PVC	UCL, Lancaster, BT
UUNET	3ffe:1108:800::/40	64Kbit X21	UUNET-UK

The network connections to Southampton are run on Cisco (UUNET) and Telebit (QTPv6, Bermuda) routers. To enable hosts within the internal network to become multihomed, subnetworks are created by breaking out multiple /64 networks using PC-based FreeBSD routers with quad Ethernet cards installed (such a router platform costs less than 800 euros). We run with FreeBSD 4.0 and are currently experimenting with the GNU Zebra router.

8.2.2.6 Planned experiments

Experiments at participant sites involved with multihoming work are currently still at early stages. This is in part because the IETF documents on the subject are themselves only at draft status, but also because most NRNs have only recently begun issuing address space under their new SubTLA allocations.

The multihoming work under QTPv6 will continue as we look to identify scenarios for study by the participants. This may include testing work suggested by the IETF, or alternative methods that have been tabled (e.g. the use of just site-local addresses within a site, or making some use of the global uniqueness of the EUI-64 part of an autoconfigured IPv6 address).

8.2.2.7 Multihoming work within the IETF

There are three Internet Drafts in the IETF IPng working group:

Default Address Selection for IPv6 [QTPV6-02]
This draft investigates the issue of how a host might select source and destination addresses for a new TCP/IP connection. Algorithms are proposed which consider factors such as address scope (link local, site local, global), longest match prefix (the longer the match, the better the selection) and deprecation status (avoid using deprecated prefixes). Mechanisms for administrators to set policies to override those algorithms are discussed, as is the issue of trying multiple source-destination address combinations at session start-up.

Multihomed routing domain issues for IPv6 aggregatable scheme [QTPV6-03]
A good overview of multihoming methods is presented in this draft. The "mutual peering" scenario illustrates one technique by which a site with two providers can inherit prefixes from both providers, advertise both prefixes to both providers, who in turn peer their network prefixes between each other.

The result offers resilience in the event of a single link failure, but does not necessarily scale to more than two providers. An interesting and novel technique described in the draft makes use of IPv6 mobility methods to enable an existing TCP/IP connection to continue operating when one link into a multihomed site fails. An internal host can learn not to use the prefix of the broken link for a new connection by the fact that the link prefix will be advertised as deprecated. An external host will not use the broken link inbound as it should try all addresses returned by the DNS and the broken link address will fail while the still-working link address will succeed. By using IPv6 mobility, an internal communicating host using a source address in the broken link range can in theory allow a remote host to believe that it has, in effect, migrated to the other (working) prefix in the local multihomed network (and thus the session can continue).

IPv6 Multihoming with Route Aggregation [QTPV6-04]
Here a method is proposed whereby a site nominates a primary ISP connection, but it also connects to a secondary ISP. The site uses a network prefix from its primary ISP, and advertises that prefix to its secondary ISP, who also advertises it to the primary ISP. Should the link to the primary ISP fail, data will continue to flow via the secondary site.

8.2.2.8 Conclusions

Multihoming experiments under QTPv6 are still at an early stage. However, the frameworks for trials to begin are now in place at a number of sites, including Southampton and ACOnet.

There are many complex problems related to multihoming, not least handling persistent TCP/IP connections when endpoints move or disappear (e.g. through router renumbering). There are many issues with services such as (dynamic) DNS and DHCPv6 that impact on multihomed sites.

We envisage that participants may soon be able to attempt to test the feasibility and robustness of the techniques described in at least the above IETF Internet Drafts. We will also continue to study other issues as and when they are raised and where they can be practically investigated.

8.2.3 DNS

Initially, the plan was to use a "production-grade" version of bind on a stable platform, i.e. bind 8.2.2_P5 on RS6000 and with an IPv6-aware operating system (AIX).
However, while bind 8.2.2 does support AAAA Resource Records, it does not support the more modern RR (resource record) types that are being introduced specifically to support IPv6:

- A6 RR type
- DNAME RR type
- binary labels

Bind 8.2.2 could not be accessed using IPv6 packets to submit queries, even though the platform itself was IPv6-aware.
In order to track more up-to-date developments, the focus of the efforts was moved to different platforms (FreeBSD and Solaris8 on Intel PCs) and to the most recent bind versions (bind 9 beta releases). The reasons for that are mainly:

- Early in 2000, bind-9 became available as beta-1, and later beta-2, including support for A6, DNAME and "binary labels". In particular DNAME RRs and "binary labels" are seen as essential in supporting re-numbering and ReverseDNS. Both of these aspects are of paramount interest for the QTPv6 environment where both pTLA address space as well as sTLA address space is being deployed.

- While AIX was one of the first (commercially available) operating systems to support IPv6 end-node facilities, most of the more recent development efforts seems to have gone into open source environments (Linux, FreeBSD, KAME) and Solaris on Sun and Intel hardware.

8.2.3.1 On-going tests and achievements

Testing of IPv6 DNS facilities was approached from different directions, in particular:
- bind compatibility with various platforms
- bind functionality
- bind operational stability
- client access to bind and service verification

Bind compatibility with various platforms:

Bind-9 beta-1 was not easy to install (i.e. compile) on some of the platforms in use at our site. Other problems surfaced after the installation itself when we tried to access the DNS services by using IPv6 as the transport packet format. Due to the fact that bind-9 beta-2 has been available for quite a while, and beta-3 is due shortly, there is no point in going into specific details of beta-1 for the purposes of this report.

Bind-9 beta-2 turns out to be more compatible with different environments and IPv6 stack implementations. Also, most of the deficiencies that were either documented in the release notes or reported as bugs for beta-1, have been fixed.

As of the time of submitting this report, bind-9 beta-2 has been successfully installed on the following platforms:
- Solaris8 Intel PC (early access)
- Solaris8 Sun Sparc (early access)
- FreeBSD 3.4 + KAME stack
- FreeBSD 4.0

Bind functionality

Bind 9 beta 2 is supposed to support the IPV6-specific new RR types A6, DNAME and more generally, "binary labels".

In the context of this research program, DNAME and "binary label" functionality has been verified to work in principle. Deployment of A6 records was not yet successful, but this may be due to poor documentation and/or background information available to the project group. This is one item for further investigation.

Bind-9 beta-2 supports access to DNS services by using IPv6 packet formats for communication between the client (resolver, nslookup) and the server (bind process). However, it turns out to be quite cumbersome to verify this functionality on various platforms, and with different applications), because:

- Some platforms still lack full support for IPv6 transport implementations in the resolver libraries
- At least one platform has even reverted to supporting IPv4 transport only (most notably Cisco 12.0 based IPv6 beta IOS). While this restriction is documented in release notes or other documentation shipped with the code beta version code, it is still seen as a major stumbling block for widespread IPv6 deployment!
- Some platforms do not (easily) allow the specification of (an) IPv6 nameserver address(es) in the resolv.conf file.

- Due to the requirement to interoperate with older bind versions (which do not support IPv6 transport), usually the end system manager has to configure both IPv6 addresses (where supported) and IPv4 addresses in the resolv.conf file. Mixing IPv4 and IPv6 addresses adds another level of complexity to the debugging and verification scenario.

Bind operational stability:

With bind-9 beta-2 the operational stability of the name server code itself was improved considerably and in situations of correct configuration crashes or outages of the bind service itself were not observed. However this cannot not be considered proof of operational quality because the load on the servers was very low, and no stress testing with regard to performance or resilience against malformed queries was performed.
This is left for further investigation, in particular stress-testing against malformed queries should be high up on the list of future tests.

Even with bind-9 beta-2 it is easily possible to crash the name server when trying to load malformed zone files that make use of the more modern RR types (binary labels and DNAMES). This problem has been reported to the bind-9 developers and is waiting for resolution. Until this flaw has been rectified, there is no point in deploying bind-9, even for a pre-production environment.

8.2.3.2 Problems identified

User Interface Problems / DNSv6

Problem: cisco 12.0 based beta IOS no longer allows the name resolver to use IPv6 Addresses for the name servers.
Status: not yet resolved
Additonal Note: this functionality was present in the 11.3 based version

API problems

Problem: Various applications are not yet capable to use IPv6 name services, though those applications work when supplied with IPv6 addresses.
Status: waiting for resolution in updated operating system versions.

Problem: various resolver libraries are not yet capable to use IPv6 packet formats to interact with name servers.
Status: waiting for resolution in updated operating system versions.

Problem: some platforms do not yet support IPv6 packet formats to interact with name servers
Status: waiting for resolution in updated versions of operating system software.

Stability problems

Problem: bind 9 beta versions accessible at the time of submitting this summary are still vulnerable to incorrect configuration and/or zone files.
Status: to be checked against more recent beta or pre-production versions of bind 9 code.

Problem: bind 9 beta versions accessible at the time of submitting this summary still require major effort to install on some commercially available operating system versions and/or to make use of the IPv6 protocol stack as provided.
Status: to be checked against more recent beta or pre-production versions of bind 9 code.

8.2.3.3 Next steps

For the imminent future, follow-up on the following aspects is planned and is regarded to be of high priority:

- Combine both stateless, as well as stateful (DHCPv6) autoconfiguration facilities, with DNS services, if possible in combination with multi-homing tests;
- Verify proper operation of the IPv6-specific RR types in support of renumbering from one IPv6 address range (e.g. pTLA address space) to a different IPv6 address range (e.g. sTLA address space);
- Follow up on compatibility with and stability on operating platforms being used at many sites (most notably Linux on Intel hardware) and new beta older final releases of operating system distribution (most notably Solaris8 on Intel and Sun hardware);
- Follow up on the API and application issues to assess the quality of integration, performance and stability for deployment in an "end-user" environment (as compared to a test-lab and proto-typing environment);
- Configure the primary name server (managed by DANTE) and 2 secondary nameservers (managed by Vienna University/ACOnet, AT and by Southampton University, UK) for the reverse zone and to obtain delegation of the reverse zone for the QTPv6 address space in pTLA space
- Devise and deploy test scenarios and mechanisms to stress test IPv6 DNS functionality with regard to performance and resilience against malformed requests;
- As soon as a stable configuration is achieved and verified to work, make the various configurations and zone files available to help in widespread deployment within the academic community.

8.3 References

[QTPV6-01] IETF Internet draft, draft-ietf-ipngwg-ipaddressassign-00.txt
[QTPV6-02] IETF Internet draft, draft-ietf-ipngwg-default-addr-select-00.txt
[QTPV6-03] IETF Internet draft, draft-ietf-ipngwg-multi-isp-00.txt
[QTPV6-04] IETF Internet draft, draft-ietf-ipngwg-ipv6multihome-with-aggr-00.txt

9 SDH ISSUES

Activity leader: Victor Reijs, SURFnet bv, The Netherlands

Participants: All TF-TANT participants.

Keywords: SDH, concatenation, DWDM

9.1 Description

Many connectivity providers in Europe provide their SDH services using equipment that is not able to provide concatenation of multiple STM-1 links (to realise one clear link of Nx155Mbps). This leads to the effect that it is not possible to procure a clear link of e.g. 644 Mbit/s (STM-4c) between two routers. In most instances a 622Mbps circuit is presented as an STM-4 link, which is in effect 4 separate links of 155 Mbit/s.

9.2 Goals

The goals of this activity address two aspects:

- To determine what the effects of SDH implementations (mainly concerning concatenation) will have on aggregated IP traffic over these SDH links;
- To make SDH providers aware of the problems arising from the lack of SDH concatenation implementations.

9.3 Effects of SDH implementations on aggregated IP traffic

9.3.1 European SDH/STM implementations

Most implementations of SDH networks in Europe are internally based on the switching of STM-1 (155 Mbit/s) links. The resilience (APS: Automatic Protection Switching) within these SDH networks is also based on STM-1 links. As long as one has traffic streams of less than or equal to 155 Mbit/s, there is no problem. For traffic streams higher than 155 Mbit/s problems may arise.
In the present day SDH networks in Europe, e.g. a 622 Mbit/s link will be made by multiplexing 4 paths of 155 Mbit/s, called STM-4. If no load balancing is done by the IP equipment on this STM-4 interface, one traffic stream is still not able to utilise more than 155 Mbit/s. Load balancing on packet level by the IP equipment sounds like a solution. But if the SDH network is still doing its (resilience) switching on the bases of STM-1, the delays in the different link can in principle be different. This may increase the number of misaligned IP packets, which will seriously hamper IP traffic.
To overcome these problems the SDH network must support concatenated links of 622 Mbit/s, called STM-4c. ('c' of concatenation).
So when talking about 622 Mbit/s links it is always important to check what the SDH network provides and what the IP equipment supports: STM-4 or STM-4c. For applications wanting to use more than 155 Mbit/s, STM-4c is mandatory.
The major SDH manufacturers provide systems that can support STM-4c, but this does not say that the real implemented SDH network have this more advanced equipment! Most American based IP equipment manufacturers can provide STM-4 or STM-4c interfaces for their routers and switches.
In principle this problem will also arise when using STM-16(c) or DWDM. In DWDM the links are 2.4 Gbit/s (and in the future 10 Gbit/s). Concatenation is not yet defined in DWDM, but happily it will also take some time before single applications will need more than 2.4 or 10 Gbit/s...

9.3.2 American SDH/OC implementations

Interfaces that can support OC-12c are in most cases compatible with STM-4c (sometimes small software configuration changes are need). All OC interfaces already have the 'c' incorporate (like OC-3c <- STM-1 and OC-12c <- STM-4c, etc.).

9.4 Raising awareness of implications of SDH concatenations issues

Through DANTE and TF-TANT meetings the NRNs been made aware of the STM-4 and STM-4c difference. Furthermore talks have been held with IP and SDH equipment manufacturers (e.g. Cisco, Cabletron, Alcatel, KPN Telecom, Siemens)

10 VIRTUAL PRIVATE NETWORK SERVICE

Activity leader: Victor Reijs, SURFnet bv/HEAnet, The Netherlands/Ireland

Participants: All TF-TANT participants (for the actual MPLS-VPN test; see the overview of the MPLS activity).

Keywords: MPLS, VPN

10.1 Description

Virtual Private Networks are becoming more and more important (such as extra-nets over WAN, tele-working, CoS related services, etc.). It is therefore important to know what the features of IP VPNs are and how they will effect wide area IP networks. Also the interaction on generic services such as identification, authorisation and accounting is important, so that these generic services can be placed at the correct logical location in the WAN and/or LAN.

10.1.1 What is a VPN

The acronym VPN is used in many ways, and stands for Virtual Private Network. The general idea in all it implementations is that it is possible to obtain a network service which resembles a private network but which shares a public infrastructure with other organisations who in turn build their own VPN.
With a VPN service, it is guaranteed that nobody outside the VPN can access the information being exchanged within the VPN. Another aspect of VPNs is their capability to support operational management facilities which are effective only within the VPN, such as CoS facilities.
Two types of VPNs are briefly described: Access VPNs and Intranet/Extranet VPNs

10.1.1.1 Access VPN

Access VPNs encompass two architecture options: client-initiated or network access server (NAS)-initiated connections. With client-initiated Access VPNs, users establish an encrypted IP tunnel from their client across a service provider's shared network to the corporate network.
The alternative architecture for Access VPNs defines tunnels initiated from the NAS. In this scenario, remote users dial into a service provider's point of presence (POP). The service provider, in turn, initiates a secure, encrypted tunnel to the corporate network. With a NAS-initiated architecture, service providers authenticate the user to allow initial access to the corporate network. Businesses, however, retain control of their own security policy, authenticating users, authorising access privileges, and tracking user activity on the network.
Technologies that can be used for Access VPNs are: L2F, L2TP and IPSec. This type of VPNs is not studied by TF-TANT.

10.1.1.2 Intranet and Extranet VPNs

Intranet and Extranet VPN services link remote offices and potentially link suppliers, partners, customers, or communities of interest over a shared infrastructure with the same policies as a private network. The following methods are possible:

- IP tunnels, which can be based on the IP Security Protocol (IPSec), generic routing encapsulation (GRE).
  These technologies use standards to establish secure, point-to-point connections in a mesh topology that is overlaid on the service provider's trusted IP network or the public Internet. An IPSec architecture, includes the IETF proposed standard for IP-based encryption and enables encrypted tunnels from the access point to and across the intranet or extranet.
  This type of VPN was not studied by the TF-TANT group.

- Virtual circuits based on ATM or Frame Relay. With this architecture, privacy is accomplished with permanent virtual circuits (PVPCs/PVCCs) instead of tunnels. Encryption more commonly is applied as needed by individual applications. Virtual circuit architectures provide VPNs prioritisation through quality of service (QoS) for ATM and committed information rate (CIR) for Frame Relay. It is possible to set-up VPNs using DANTE's MBS service which is based on ATM PVCs.
- Multiprotocol Label Switching (MPLS)-based services that enable secure, business-quality VPN solutions that scale to support tens of thousands of VPN customers over a multiprotocol backbone.
  MPLS forwards packets using labels, VPN-based addresses analogous to a postal office zip code. The VPN identifier in the label isolates traffic to a specific VPN. In contrast with IP tunnel and virtual-circuit architectures, MPLS-based VPNs enable connectionless routing within each VPN community. Encryption more commonly is applied as needed by individual applications. TF-TANT will only look into the MPLS-based services

10.2 Goals

- Gain knowledge on the standardisation/implementation status of VPNs in IP networks;
- Understand the effects of VPNs on the wide area IP networks;
- Understand the security issues (identification, authorisation, accounting and encryption) that are relevant to VPNs;
- Determine the use of and policies on VPNs in national research networks;
- To carry out testing of VPNs in the wide-area using TEN-155 and the MBS.

10.3 Standardisation and implementation status of VPNs in IP networks

Several internet drafts address MPLS VPNs, and the most significant ones are listed and summarised:

- MPLS VPN Interworking[1] Virtual private networks (VPNs) based on Multiprotocol Label Switching (MPLS) are called 'MPLS VPN'. This document discusses motivation and a model of interworking among MPLS VPNs. It then proposes functional capabilities for the interworking such as realisation of security, mapping of the QoS class, dynamic routing information distribution;
- Core MPLS IP VPN Architecture[2]: This draft presents an approach for building core VPN services in a service provider's MPLS backbone. This approach uses MPLS running in the backbone to provide premium services in addition to best effort services. The central vision is for the service provider to provide a virtual router service to their customers. The keystones of this architecture are ease of configuration, user security, network security, dynamic neighbour discovery, scaling and the use of existing routing protocols as they exist today without any modifications;
- BGP/MPLS VPNs[3]: This document describes a method by which a Service Provider may use an IP backbone to provide VPNs for its customers. MPLS is used for forwarding packets over the backbone, and BGP is used for distributing routes over the backbone. The primary goal of this method is to support the case in which a client obtains IP backbone services from a Service Provider or Service Providers with whom it maintains contractual relationships. The client may be an enterprise, a group of enterprises which need an extranet, an Internet Service Provider, another VPN Service Provider (even one which uses this same method to offer VPNs to clients of its own), an application service provider, etc. The method makes it very simple for the client to use the backbone services. It is also very scalable and flexible for the Service Provider, and allows the Service Provider to add value;
- Constraint-Based LSP Set-up using LDP[4]: Label Distribution Protocol (LDP) takes care of the distribution of labels inside one MPLS domain. One of the most important services that may be offered using MPLS in general and LDP in particular is support for constraint-based routing of

traffic across the routed network. Constraint-based routing offers the opportunity to extend the information used to set-up paths beyond what is available for the routing protocol. For instance, an LSP (Label Switched Path) can be set-up based on explicit route constraints, QoS constraints, and other constraints. Constraint-based routing (CR) is a mechanism used to meet Traffic Engineering requirements. These requirements may be met by extending LDP for support of constraint-based routed label switched paths (CR-LSPs). Other uses for CR-LSPs include MPLS-based VPNs. This draft specifies mechanisms and TLVs for support of CR-LSPs using LDP.

The MPLS VPN protocol looks after the following services:

- It provides scalable VPNs (from hundreds to thousands of sites) over one physical IP infrastructure. In fact, MPLS VPNs provide a connectionless VPN service whereby the user makes a connection only to the first provider router (called PE-router), without having to make numerous point-to-point connections as in others type of VPN implementations: IPsec, L2TP, L2F, GRE, Frame relay or ATM virtual circuits;
- MPLS VPNs provide the same level of security as point-to-point based VPNs, because it guarantees that packets will only reach end systems belonging to (and trusted by) the same VPN;
- IP addresses only have to be unique within one VPN, so there is no conflict of addresses when using local addresses over WANs.
- In future MPLS VPN implementations it will be possible to take advantage of the MPLS TE and MPLS CoS services, to optimise routing and provide dedicated CoS within a particular VPN.;

Implementations are available from Cisco 12.07(T)

10.4 Effects of VPNs in WAN environments

The effects of VPNs inside a WAN are quite small. Only new software is needed, which is already quite stable at this moment.

10.5 Security issues

The security issues (identification, authorisation, accounting and encryption) issues were not studied because they were felt to be part of the access side and not of the WAN side. The WAN expects that all packets injected by connections to the VPN belonging to that particular VPN, thereby pushing the security issues more into the connected access network (generic services and policy control).

10.6 VPNs in National Research Networks

In the NRNs there is not yet much use for MPLS VPN. The idea though exists that MPLS VPN could be provided to certain groups within the total user group of an NRN that have special needs (on security or CoS, clustering of supercomputers, etc.). MPLS VPN seems at this moment to be more a requirement for public ISPs than for NRNs.
HEAnet Ltd (Ireland) plans to make us of MPLS VPNs: The backbone of the HEAnet network will become a 34 Mbit/s VPN within the Eircom infrastructure. To implement this, HEAnet and Eircom are testing (as part of the HEAnet National Backbone project in 2000) MPLS VPN, to check if sound and secure VPN services can be provided. If so, the Eircom infrastructure could be opened for use by other customers, which will share the same physical infrastructure.

10.7 Testing VPNs in the TF-TANT network environment

Tests are conducted within the MPLS activity.

10.8 References

[1] MPLS VPN Interworking
Junichi Sumimoto, e.a.
draft-sumimoto-mpls-vpn-interworking-00.txt
Expires August 8,2000

[2] Core MPLS IP VPN Architecture
Karthik Muthukrishnan e.a.
draft-muthukrishnan-mpls-corevpn-arch-01.txt
Expires November 1, 2000

[3] BGP/MPLS VPNs
Eric C. Rosen, e.a.
draft-rosen-rfc2547bis-00.txt
Expiration Date: September 2000

[4] Constraint-Based LSP Set-up using LDP
Bilel Jamoussi (editor),
draft-ietf-mpls-cr-ldp-03.txt
Expiration Date: March 2000

11 WAVE DIVISION MULTIPLEXING (WDM)

Activity leader: Victor Reijs, SURFnet bv, The Netherlands

Participants: All TF-TANT participants.

Keywords: WDM, MPLS, MPLambdaS, restoration, performance monitoring

11.1 Description

Wave Division Multiplexing is an emerging technology to get high bandwidth on a single fiber (pair), upto at least 640 Gbit/s. Over this technology it is possible to transport IP over SDH over WDM (the present way of doing it) and, in the future, IP over WDM. This activity will primarily streamline experience exchange obtained on a national or local environment.

11.2 Goals

- Exchange information between the participants on the standardisation process of WDM technology and on the use of WDM in WAN networks.
- Test in a small environment, WDM on international links (if possible)
- Get an idea on fiber quality, management and resilience issues of WDM provision.

11.3 Standardisation process and use of WDM in WANs

11.3.1 Documentation

There are several Internet drafts on optical networking:

- IP over Optical Networks - A Framework [1]
- Multi-Protocol Lambda Switching: Combining MPLS Traffic Engineering[2] and Issues in Combining MPLS Traffic Engineering Control With Optical Cross-connects [3]
- Performance Monitoring in Photonic Networks in support of MPL(ambda)S http://search.ietf.org/internet-drafts/draft-ceuppens-mpls-optical-00.txt [4]

11.3.2 Activities within the NRNs

An overview of the NRN/local activities on optical networking is hereby listed:

- **DFN**: DFN information on the gigabit test beds in Germany can be found at: http://www.dfn.de/projekte/gigabit/presentation-tnc.html;
- **GARR-INFN**: INFN-GARR is planning a WDM based backbone in Italy as a replacement to the present backbone of the GARR-B network (see http://www.garr.it/garr-b-home-engl.shtml). In this project it is intended to replace initially only the core network, passing from 4 core nodes of GARR-B to (at least) 5 core nodes. A minimum of 6 dark fibers will be interconnected with WDM multiplexers equipped with 2.5 Gbit/s and with a minimum of 8 and a maximum of 32 colours send/receive modules. The peripheral infrastructure of the present GARR-B, based on 16 PoPs, will remain initially unchanged. INFN-GARR started already the process of selection of multiplexers and routers. A pilot network should be running by mid 2000. The official call for tender will be issued immediately afterward and the working GARR-G is expected to run by March 2001;

- **Grnet**: detailed information on the GRnet WDM experiments available at http://www.grnet.gr/reports/wdm_en.html;
- **INFN**: INFN has a project called QUADIS for the study of the deployment of WDM for applications such as data acquisition in the high-energy physics experiments. In this project the deployment of WDM in larger scale networks will be studied, for example in the metropolitan area. See http://www.cnaf.infn.it/CNAF/progetti/quadis/index-eng.html for more information;
- **SURFnet bv**: SURFnet bv is planning an DWDM based backbone in the Netherlands as a follow up to SURFnet4. A call for tender has been issued for this transmission and network infrastructure. A pilot network (initially based on STM-16c) will be operational by mid-2000. Experiments will be carried out with regard to MPLS FRR and DWDM and the manageability of DWDM networks. More information is available at http://www.gigaport.nl/en/en_network.html;
- **TERENA**: http://www.terena.nl/tech/wave-workshop/index.html contains the minutes of a workshop on optical networking held at TERENA in November 1999;
- **UCL**: Activities of UCL with regard to this technology is available at http://www.cs.ucl.ac.uk/research/jif/

11.4 Test environment

No test environment has been implemented in the time frame of TF-TANT. An international dark fiber pair exists between The Netherlands and Belgium (part of the Electra project) which could be used for these test, but no equipment and manpower was available for doing test on this dark fiber pair. A Wavelength Workshop in November 1999 has been organised by TERENA. Due to unavailability of international links, this did not evolve into an optical European test network.

11.5 Management and resilience issues

Important issues for WDM-like technologies:
- On schedule wavelength provisioning;
- MPLS FRR for fast IP restoration;
- Signalling in case of lost wavelengths/fibres;
- Optimisation of bandwidth provisioning for customers;
- Class based restoration;
- Traffic monitoring tools

Recent work in the IETF [2,3] is starting to indicate a direction for the first 3 items above, though this work is still in its beginning stages. However, it looks like there is an initiative to use MPLS signalling for signalling on the OXC level (MPLS FRR).
In order to provide these functions the need to do traffic monitoring is important. There are several aspects of traffic monitoring that could be useful as input to developing models for WDM functions in IP networks. Specifically, the daily variation in traffic demand at GigaPoPs and access to GigaPoPs, and the overall traffic matrix that shows the magnitude (coarse scale) of flow from all points in the network to all other points in the network.

11.6 References

[1] IP over Optical Networks - A Framework,
James Luciani e.a., draft-ip-optical-framework-00.txt,
Expiration Date: Sept., 10, 2000

[2] Multi-Protocol Lambda Switching: Combining MPLS Traffic Engineering Control With Optical Crossconnects,
Daniel O. Awduche, e.a.,
draft-awduche-mpls-te-optical-01.txt,

Expiration Date: May 2000

[3] Multi-protocol Lambda Switching: Issues in Combining MPLS Traffic Engineering Control With Optical Cross-connects,
Debashis Basak, e.a.,
draft-basak-mpls-oxc-issues-01.txt,
Expires: August, 2000

[4] Performance Monitoring in Photonic Networks in support of MPL(ambda)S,
Luc Ceuppens, e.a., d
raft-ceuppens-mpls-optical-00.txt ,
Expiration Date: September 2000

12 GLOSSARY OF TERMS

AAA	Authentication, Authorisation and Accounting
ABR	Available Bit Rate
ACR	Available cell Rate
AF	Assured Forwarding
ARP	Address Resolution Protocol
APS	Automatic Protection Switching
AS	Autonomous System
ATC	ATM traffic class
ATM	Asynchronous Transfer Mode
BA	Behaviour Aggregate
BE	Best Effort
BGP	Border Gateway Protocol
BGMP	Border Gateway Multicast Protocol
CAC	Call Admission Control
CAR	Committed Access Rate
CBR	Constant Bit Rate
CB-WFQ	Class Based Weighted Fair Queuing
CE	Customer Edge
CLIP	Connection-Less Internet Protocol
CLNP	Connection-Less Network Protocol
CoS	Class Of Service
CR	Constraint-based Routing
DBR	Deterministic Bit Rate
DHCP	Dynamic Host Configuration Protocol
Diffserv	Differentiated Services
DNS	Domain Name Service
DoS	Denial of Service
DSCP	Differentiated Service Code Point
DMTF	Distributed Management Task Force
DVMRP	Distance Vector Multicast Routing Protocol
DWDM	Dense-Wave Division Multiplexing
EF	Expedited Forwarding
FIN	Close connection (TCP message)
FRR	Fast Route Restoration
FUA	Flow Update Acknowledgement
FUN	Flow Update Notification
GPS	Global Positioning System
GRE	Generic Routing Encapsulation
iBGP	internal BGP
ICMP	Internet Control Message Protocol
ICR	Initial Cell rate
IETF	Internet Engineering Task Force
IGP	Internal Gateway Protocol
ILMI	Interim Local Management Interface
IP	Internet Protocol
IPDV	Instantaneous Packet Delay Variation
IPSec	Internet Protocol SECurity
ISP	Internet Service Provider
IS-IS	Intermediate System to Intermediate System
L2F	Layer Two Forwarding

L2TP	Layer 2 Tunneling Protocol
LAN	Local Area Network
LANE	Local Area Network Emulation
LFAP	Lightweight Flow Admission Protocol
LIS	Logical IP sub-network
LDP	Label Distribution Protocol
LRU	Least Recently Used
LSP	Label Switched Path
LSR	Label Switching Router
MAC	Media Access Protocol
MASC	Multicast Address-Set Claim
MBGP	Multiprotocol BGP
MBS	Managed Bandwidth Service (of the TEN-155 network)
MIB	Management Information Base
MP-BGP	Multiprotocol BGP
MPLS	Multi Protocol Label Switching
MPLamdbaS	Multi-Protocol Lambda Switching
MRTG	Multi Router Traffic Grapher
MRTd	Multithreaded Routing Toolkit Daemon
MRM	Multicast Routing Monitor
MSDP	Multicast Source Distribution Protocol
MTU	Maximum Transfer Unit
NDA	Non Disclosure Agreement
NHRP	Next Hop Resolution Protocol
NRN	National Research Network
OADM	Optical Add and Drop Multiplexer
OC	Optical Carrier
OSPF	Open Shortest Path First
OXC	Optical X-cross Connect
PATH	Path message for RSVP
PCR	Peak Cell Rate
PE	Provider Edge
PHB	Per Hop Behaviour
PIM-SM	Protocol Indipendant Multicast – Sparse Mode
PPP	Point to Point Protocol
PQ	Priority Queuing
PVC	Permanent Virtual Circuit
pTLA	pseudo Top Level Area
QoS	Quality of Service
RA	Router Advertisement
RdF	Rate decrease Factor
RiF	Rate increase Factor
RIPng	Routing Information Protocol next generation
RSVP	Resource Reservation protocol
RTP	Real Time Protocol
RTCP	RTP Control Protocol
RTFM	Real Time Flow Measurement
QTP	Quantum Test Programme
SBR2	Statistical Bit Rate-2
SBR3	Statistical Bit Rate-3
SDH	Synchronous Digital Hierarchy
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical NETwork
SOO	Site of Origin
sTLA	sub Top Level Area

STM	Synchronous Transfer Module
SVC	Switched Virtual Circuit
TCP	Transmission Control Protocol
TDP	Tag Distribution Protocol
TLV	Type, Length, Value
TX Queue	Transmission Queue
UDP	User Datagram Protocol
VBR-nrt	Variable Bit Rate non real time
VLAN	Virtual LAN
VoIP	Voice over IP
VPN	Virtual Private Network
VRF	Virtual Routing Forwarding table
WAN	Wide Area Network
WDM	Wave Division Multiplexing
WFQ	Weighted Fair Queueing
WRED	Weighted Random Early Discard
WWW	World Wide Web