Appendix A

9 TESTBED CONFIGURATION

9.1 Introduction

This document describes the MESCAL testbed that has been used during the development and validation phases 1, 2 and 3 defined in [D2.1]. This testbed is located in FTR&D premises, in Caen, France.

The same platform is used for carrying out these 3 phases. From phase 1 to phase 3 the core of the platform (AS topology, links, addressing, core technologies) remains the same but evolves, mainly in terms of configuration in order to support the specific constraints and requirements of each of these phases.

Phase 1 aims at validating the overall testing environment including: hardware, software, traffic generators, BGP and QoS configuration. This phase needs a large number of Linux and non-Linux features to be activated but does not include any of the new features MESCAL has to develop. This phase allows verifying that the selected environment is suitable for supporting phases 2 and 3 and definitively confirm the pertinence of MESCAL technical choices for this testbed. Additionally, the experience gained during this phase will help to identify and/or improve all appropriate tools for maintaining the testbed in an efficient way.

In particular, this phase will allow:

- To validate the inter-domain routing with ZebOS running on Linux PCs by setting-up several ASs and by configuring BGP between those domains.
- To validate, at the data plane level, the DSCP swapping (marking/remarking) between ASs in order to signal a given meta-QoS-class.
- To validate the implementation of l-QCs in each domain using Linux traffic control features.

Phase 2 aims at validating the loose service option. The same testbed will be used for validating q-BGP implementation and the enhanced Linux IP forwarding. Depending on the tests that will be performed the number and the definition of local-QoS-classes in each AS will be adapted together with the pSLS definitions. This will be on per test basis.

Phase 3 aims at validating the Path Computation System (PCS) function. For this purpose, the PCS implementation will be uploaded in an appropriate set of Linux routers but the overall infrastructure will remain the same as for the phase 1. No MPLS features will be deployed since the project will focus only on the computation of inter-AS paths for establishing inter-domain LSPs.

More precisely, this document provides information on:

- The AS topology
- The deployed hardware,
- The links set-up between routers,
- The IP addressing scheme,
- The BGP and q-BGP configurations,
- The QoS policies defined for each domain, including a description of the local-QoS-classes deployed in each AS,
- The tools developed for maintaining, configuring and monitoring the platform,
- The tools used for generating and analysing IP traffic.

9.2 Autonomous system topology

The testbed supports 8 autonomous systems. A single router implements each AS except AS4, which is composed of 3 routers. AS4 and AS6 represent Tier-one providers and have more network resources allocated. AS1 represents a Tier-3 provider. The remaining ASs are Tier-two providers.

In phase 1, BGP is activated at the boundaries of each domain and iBGP is activated within AS4 domain. In phase 2 q-BGP and q-iBGP will be activated instead.

These ASs are interconnected as roughly shown in Figure 94 and depicted in more details in Figure 95.

Some inter-AS interconnections have been doubled in order to allow more sophisticated inter-domain routing tests and to evaluate eventual load balancing features especially between:

- AS4 and AS6
- AS3 and AS4

Even if this topology is far from the real Internet it will nevertheless allow learning different QoS routes thanks to ad-hoc activation of pSLS. If necessary, a maximum of 6 crossed AS can be reached provided the appropriate configuration.

Figure 95 gives a more accurate view of the testbed. It groups on the same picture different level of information:

- AS numbers
- Links between ASs
- IP address of the interfaces, with their interface number on each machine
- DS code point values used to signal local-QoS-classes
- Agreed DS code point values used to signal meta-QoS-classes between domains together with the capacity provisioned for each class.
- Role of the remaining interface: management interface or customer interface

Smartbit interfaces are connected to each router via a dedicated interface. These interfaces are mainly used for injecting load traffic in the testbed. Some of the routers (AS1 and AS8) have additional Smartbit interfaces connected in order to inject customer traffic used for measurement purposes.

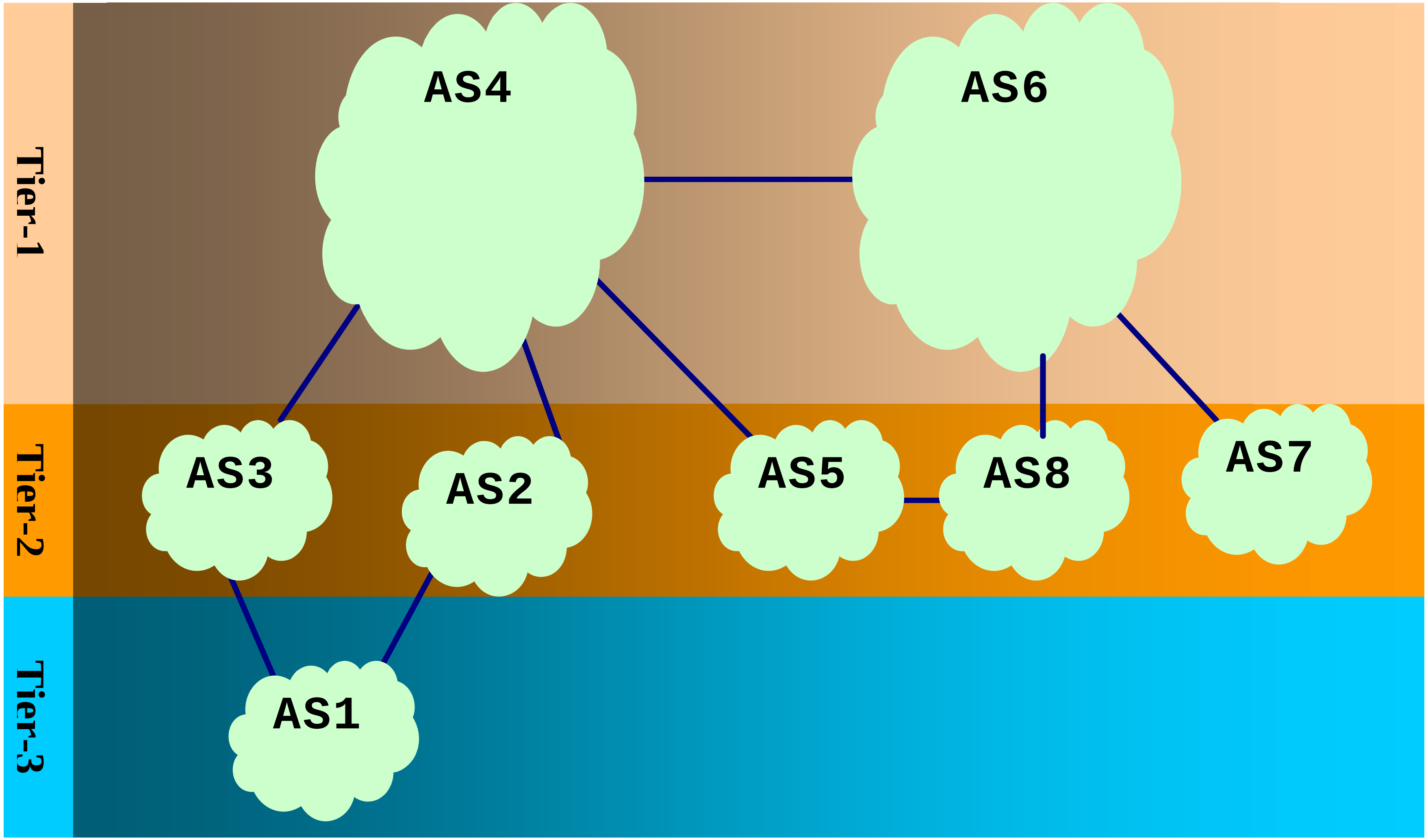

Figure 94: FTR&D MESCAL testbed: hierarchical view

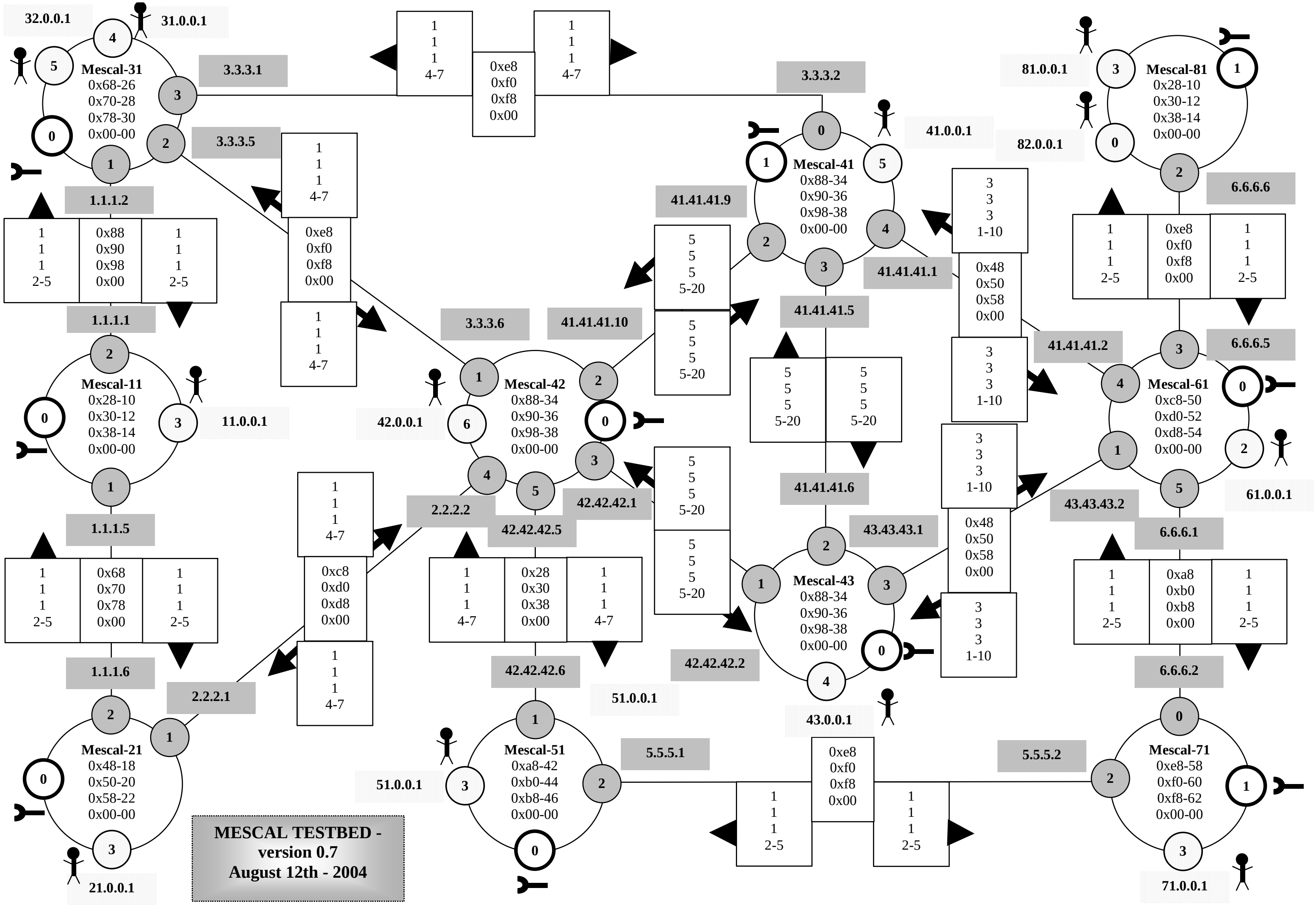

Figure 95: FTR&D MESCAL testbed: detailed architecture

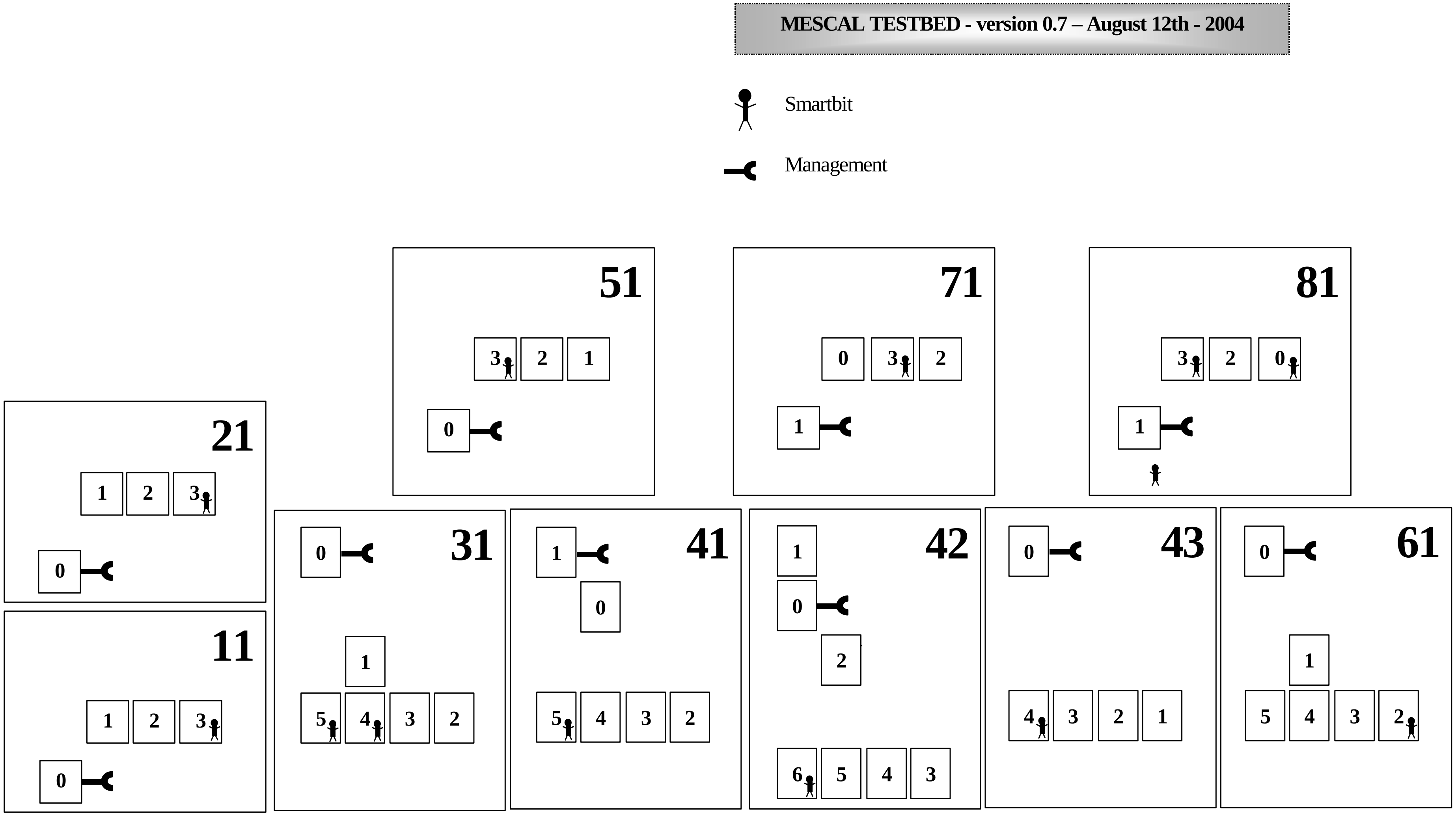

Figure 96: FTR&D MESCAL testbed: Network interfaces schema

Copyright © MESCAL Consortium, July 2005

9.3 Testbed components

The testbed deployed by France Telecom R&D in its premises contains many devices including network elements (routers, hubs…), workstations and test equipments. In this current section, we list and describe both the hardware and software components used for building the FTR&D MESCAL testbed.

9.3.1 Hardware components

9.3.1.1 PCs

PCs used in the testbed are of four types:

- Type 1: Intel® Xeo, CPU 1.7 GHz, 17 GB hard disk drive, 512 MB RAM
- Type 2: Intel® PIII, CPU 1.0 GHz, 20 GB hard disk drive, 512 MB RAM
- Type 3: Intel® P4, CPU 1.7 GHz, 20 GB hard disk drive, 512 MB RAM
- Type 4: Intel® PIII, CPU 500 MHz, 8 GB hard disk drive, 256 MB RAM

Name	Type	Available PCI Slots
Mescal11	1	4
Mescal21	3	3
Mescal31	3	3
Mescal41	2	3
Mescal42	1	4
Mescal43	2	3
Mescal51	1	4
Mescal61	1	4
Mescal71	1	4
Mescal81	3	3
PC-admin	4	4

Table 24: PC characteristics

Except PC-admin, all these PCs are used as PC-based routers.

Additional PCs could be added to the testbed and would be used to emulate customer premises.

9.3.1.2 Traffic Generators

9.3.1.2.1 Smartbits

Two *SmartBits* chassis (SMB 2000 and SMB600) are available in the lab. These equipments are used to inject traffic in the network and to carry out measurements.

The SMB 2000 is equipped with 20 Ethernet cards:

- 10 of them are 10/100 Mbit/s full duplex cards,
- The others are 10 Mbit/s half duplex cards.

The SMB 6000 is equipped with 2 cards of 2 x 10/100 Mbit/s Ethernet cards. Those cards support additional features the SMB 2000 card does not support and should be preferred when QoS measurements need to be carried out.

9.3.1.2.2 QARobots

In addition, *QARobot* will be used to generate BGP messages in order to test routing features. This tool will be used to validate the behaviour of q-BGP and more especially the conformity of BGP messages.

9.3.2 Software components

9.3.2.1 Operating system

Linux Red Hat version 9 (kernel version of 2.4.20-8) is installed on all PC-based routers and PC-admin.

9.3.2.2 Software information

Software	Version	Company/Package	Description
GCC	3.2.2	GNU	C++ Compiler
ZebOS	5.3.1	IPInfusion	Routing stack including BGP
Ethereal	0.9.8		Traffic analyzer
MGEN	4.2	Naval Research Laboratory	Traffic generator
Jnettop	0.9	GNU	Real-time interface bandwidth measurement
NTP	4.2		Time server

Table 25: Software information

During phase 1, the ZebOS routing stack is used without any modification. During phase 2, it is enhanced to support QoS related messages and information. Phase 3 relies on the q-BGP implementation realised during phase 2.

MGEN and *TG* are traffic generators that could be used as a complement to the *Smartbits*. These tools can generate UDP and TCP traffic (especially *TG*). They can set the DS code point on a per flow basis.

9.4 Configuration for phase 1

9.4.1 User' accounts

Only two accounts have been created in all PCs: "**root**" and "**mescal**".

9.4.2 Remote connection

Remote connections to the routers are achieved via SSH. There are no restrictions between testbed components. From outside the testbed, connections are filtered by an external firewall. Only **PC-admin** has been made accessible. The firewall ensures a network address translation for this latter machine. The corresponding public IP address will be provided by FTRD on request. Firewall rules will be updated to allow external connexions from MESCAL partners.

In addition, the **ftpd** service has been enabled in **PC-admin** and in the routers.

9.4.3 Internet access

An Internet access is configured in all machines involved in the testbed. All Internet connexions go through **PC-admin**, which acts as a HTTP/FTP proxy.

9.4.4 Firewall rules

Firewall rules have been added in order to control communications from/to testbed machines.

9.4.5 Time synchronisation

PC-admin is configured as a NTP server. It synchronizes its clock with a public NTP server. All routers in the testbed synchronise their clock with PC-admin.

9.4.6 Printer

A postscript printer, called **`Gutenberg`**, is available for all the machines of the testbed.

9.4.7 AS identifiers

Hereafter is listed the AS number affected to each AS. This AS number will be used when configuring BGP instances.

AS	AS number
AS1	1
AS2	2
AS3	3
AS4	4
AS5	5
AS6	6
AS7	7
AS8	8

Table 26: AS numbers

9.4.8 LANs

This table summarizes the addressing plan for each local area network.

Subnet	Address/Mask	Device/Interface	Address	Description
Administrative Network	192.168.66.0/24			
		PC Admin /Eth0	.73	AN2
		Firewall	.1	AN1

Table 27: Administrative network addressing

9.4.9 Customer addresses

The following table lists the IP network addresses used by customers connected to each AS (via a direct physical connection to each AS)

AS	Customers	IP addresses realm
AS1	Customer11	11.0.0.0/10
	Customer12	12.0.0.0/10
	Customer13	13.0.0.0/10
	Customer14	14.0.0.0/10
AS2	Customer21	21.0.0.0/10
	Customer22	22.0.0.0/10
	Customer23	23.0.0.0/10
	Customer24	24.0.0.0/10
AS3	Customer31	31.0.0.0/10
	Customer32	32.0.0.0/10
	Customer33	33.0.0.0/10
	Customer34	34.0.0.0/10
AS4	Customer41	41.0.0.0/10
	Customer42	42.0.0.0/10
	Customer43	43.0.0.0/10

	Customer44	44.0.0.0/10
AS5	Customer51	51.0.0.0/10
	Customer52	52.0.0.0/10
	Customer53	53.0.0.0/10
	Customer54	54.0.0.0/10
AS6	Customer61	61.0.0.0/10
	Customer62	62.0.0.0/10
	Customer63	63.0.0.0/10
	Customer64	64.0.0.0/10
AS7	Customer71	71.0.0.0/10
	Customer72	72.0.0.0/10
	Customer73	73.0.0.0/10
	Customer74	74.0.0.0/10
AS8	Customer81	81.0.0.0/10
	Customer82	82.0.0.0/10
	Customer83	83.0.0.0/10
	Customer84	84.0.0.0/10

Table 28: Customers IP address realms

9.4.10 Network addresses announced by each AS

This table lists the IP network addresses announced by each AS.

AS	Customers	IP addresses realm
AS1	Network11	193.251.128.0/19
	Network12	212.167.0.0/21
	Network13	62.42.0.0/16
	Network14	193.251.240.0/20
AS2	Network21	194.52.168.0/21
	Network22	202.167.22.0/24
	Network23	194.199.98.0/24
	Network24	128.223.60.102/20
	Network25	193.41.227.0/24
	Network26	192.176.123.0/24
	Network27	192.94.149.0/24
AS3	Network31	216.191.64.0/20
	Network32	154.101.11.10/24
	Network33	132.150.224.0/24
	Network34	155.234.165.12/19
	Network35	199.79.131.0/24
	Network36	198.26.215.0/24
	Network37	196.11.196.0/24
AS4	Network41	62.177.143.254/20
	Network42	62.216.31.254/19
	Network43	146.188.61.109/22
	Network44	195.69.144.12/18
	Network45	192.70.132.0/24
	Network46	216.169.114.0/24
	Network47	216.116.175.0/24
	Network48	214.3.214.0/24
	Network49	205.237.35.0/24
	Network410	204.222.17.0/24
	Network411	204.116.187.0/24
	Network412	203.116.188.0/24
	Network413	216.103.190.0/24
	Network414	216.84.141.0/24
	Network415	213.239.59.0/24
	Network416	213.205.25.0/30

	Network417	198.205.10.0/24
AS5	Network51	198.32.247.85/20
	Network52	12.129.192.134/20
	Network53	170.170.0.0/16
	Network54	203.34.233.0/24
AS6	Network61	205.204.1.0/24
	Network62	216.255.39.0/24
	Network63	193.100.167.0/24
	Network64	193.19.208.0/24
	Network65	198.129.78.23/20
	Network66	193.111.167.0/24
	Network67	192.197.12.0/24
	Network68	192.58.243.0/24
	Network69	192.43.226.0/24
	Network610	200.108.174.0/24
	Network611	200.10.200.0/24
	Network612	194.215.31.0/24
	Network613	195.177.120.0/24
	Network614	193.108.167.0/24
	Network615	170.163.0.0/16
	Network616	217.108.191.0/24
	Network617	217.10.217.0/24
	Network618	216.95.141.0/24
	Network619	212.78.144.0/24
AS7	Network71	209.215.12.0/24
	Network72	208.147.73.0/24
	Network73	207.245.48.0/27
	Network74	192.228.46.0/24
AS8	Network81	198.69.130.0/24
	Network82	193.16.208.0/24
	Network83	192.35.226.0/24
	Network84	192.236.46.0/24

Table 29: Customers IP address realms

9.4.11 Routing configuration

This section provides routing information, as currently configured in the testbed. The routing protocol used in the network is BGP (Border Gateway Protocol). Each AS has at least one e-BGP session with its adjacent ASs. i-BGP sessions have been established between all AS4 border routers.

Hereafter is an example of a BGP configuration (example of **MESCAL420**)

```
!
!
! Config for ZebOS version 5.3.1:03312003-Main (i686-pc-linux-gnu)
! Written 2004/05/19 17:59:55
!

!
banner motd Welcome to MESCAL42 Router :))
if-arbiter
!
interface lo
 ip address 42.1.1.1/32
!
interface eth0
 ip address 192.168.66.69/24
!
interface eth1
 ip address 3.3.3.6/30
```

Copyright © MESCAL Consortium, July 2005

```
!
interface eth2
 ip address 41.41.41.10/30
!
interface eth3
 ip address 42.42.42.1/30
!
interface eth4
 ip address 2.2.2.2/30
!
interface eth5
 ip address 42.42.42.5/30
!
interface eth6
!
router bgp 4
 bgp router-id 42.42.42.1
 bgp log-neighbor-changes
 bgp scan-time 10
 network 41.0.0.0/10
 network 42.0.0.0/10
 network 43.0.0.0/10
 network 44.0.0.0/10
 network 62.177.128.0/20
 network 62.216.0.0/19
 network 146.188.60.0/22
 network 192.70.132.0/24
 network 195.69.128.0/18
 network 198.205.10.0/24
 network 203.116.188.0/24
 network 204.116.187.0/24
 network 204.222.17.0/24
 network 205.237.35.0/24
 network 213.205.25.0/30
 network 213.239.59.0/24
 network 214.3.214.0/24
 network 216.84.141.0/24
 network 216.103.190.0/24
 network 216.116.175.0/24
 network 216.169.114.0/24
 redistribute connected
 redistribute static
 timers bgp 10 15
 neighbor 2.2.2.1 remote-as 2
 neighbor 2.2.2.1 advertisement-interval 5
 neighbor 2.2.2.1 prefix-list FLAN out
 neighbor 3.3.3.5 remote-as 3
 neighbor 3.3.3.5 advertisement-interval 5
 neighbor 3.3.3.5 prefix-list FLAN out
 neighbor 41.41.41.9 remote-as 4
 neighbor 41.41.41.9 advertisement-interval 5
 neighbor 41.41.41.9 prefix-list FLAN1 out
 neighbor 42.42.42.2 remote-as 4
 neighbor 42.42.42.2 advertisement-interval 5
 neighbor 42.42.42.2 prefix-list FLAN2 out
 neighbor 42.42.42.6 remote-as 5
 neighbor 42.42.42.6 advertisement-interval 5
 neighbor 42.42.42.6 prefix-list FLAN out
!
ip route 3.3.3.0/30 41.41.41.9 2
ip route 3.3.3.0/30 42.42.42.2 3
ip route 41.0.0.0/30 41.41.41.9 2
ip route 41.0.0.0/30 42.42.42.2 3
ip route 41.41.41.0/30 41.41.41.9 2
ip route 41.41.41.0/30 42.42.42.2 3
ip route 41.41.41.4/30 41.41.41.9 2
ip route 41.41.41.4/30 42.42.42.2 2
ip route 43.0.0.0/30 42.42.42.2 2
ip route 43.0.0.0/30 41.41.41.9 3
ip route 43.43.43.0/30 42.42.42.2 2
ip route 43.43.43.0/30 41.41.41.9 3
ip route 192.168.1.0/24 192.168.66.1
!
ip prefix-list FLAN seq 5 deny 192.168.66.0/24
ip prefix-list FLAN seq 10 deny 192.168.1.0/24
ip prefix-list FLAN seq 15 permit any
ip prefix-list FLAN1 seq 5 deny 3.3.3.0/30
ip prefix-list FLAN1 seq 10 deny 41.41.41.0/30
ip prefix-list FLAN1 seq 15 deny 192.168.66.0/24
ip prefix-list FLAN1 seq 20 deny 192.168.1.0/24
ip prefix-list FLAN1 seq 25 permit any
ip prefix-list FLAN2 seq 5 deny 43.43.43.0/30
ip prefix-list FLAN2 seq 10 deny 192.168.66.0/24
ip prefix-list FLAN2 seq 15 deny 192.168.1.0/24
ip prefix-list FLAN2 seq 20 permit any
```

9.4.11.1 e-bgp

An e-BGP session is configured in ZebOS routers as follows (example of AS4 declaring AS2 as neighbour):

```
router bgp 4
 bgp router-id 42.42.42.1
 neighbor 2.2.2.1 remote-as 2
```

9.4.11.2 i-bgp

In the testbed, AS4 is made of three routers. Each router has to declare the other two routers as i-BGP neighbours, to do so the following configuration has to be added (example of **`MESCAL420`**):

```
router bgp 4
 bgp router-id 42.42.42.1
 neighbor 41.41.41.9 remote-as 4
 neighbor 42.42.42.2 remote-as 4
```

9.4.11.3 Networks

In order to advertise networks prefixes, the command "**`network`**" is used as listed below:

```
router bgp 4
 network 214.3.214.0/24
 network 216.84.141.0/24
 network 216.103.190.0/24
 network 216.116.175.0/24
 network 216.169.114.0/24
```

9.4.11.4 Static routes

We made the decision not to activate an IGP protocol within AS4 domain. This is motivated by the fact that we prefer not having to modify IGP related processes feeding the FIBs when q-BGP is deployed and enabled. Thus, we will make use of static routes in order to simplify the development phase.

Copyright © MESCAL Consortium, July 2005

Static routes can be configured with the command "**ip route**". The example below illustrates the static routes that have been configured in **MESCAL420** for joining other AS4 intra-domain destination.

```
ip route 41.41.41.0/30 42.42.42.2 3
ip route 41.41.41.4/30 41.41.41.9 2
ip route 41.41.41.4/30 42.42.42.2 2
ip route 43.0.0.0/30 42.42.42.2 2
ip route 43.0.0.0/30 41.41.41.9 3
ip route 43.43.43.0/30 42.42.42.2 2
ip route 43.43.43.0/30 41.41.41.9 3
```

9.4.11.5 Prefix list

The ZebOS implementation can filter network prefixes announcements on a peer-by-peer basis thanks to the use of the "**prefix-list**" command. The configuration bellow allows to send all configured network prefixes except the 43.43.43.0/30, 192.168.66.0/24 and 192.168.1.0/24 Addresses.

```
ip prefix-list FLAN2 seq 5 deny 43.43.43.0/30
ip prefix-list FLAN2 seq 10 deny 192.168.66.0/24
ip prefix-list FLAN2 seq 15 deny 192.168.1.0/24
ip prefix-list FLAN2 seq 20 permit any
```

9.4.11.6 Fast link failover detection

ZebOS has been configured to support fast link failover detection.

9.4.11.7 BGP timers

The BGP timers that can be configured are: "**keepalive**", "**holdtime**" and "**connect**". We configured the two first timers to 10s and 15s respectively. This is achieved with the following command:

```
Bgp router 4
 timers bgp 10 15
```

The motivation behind these values is to decrease the time needed for the detection of invalid routes.

9.4.11.8 Route selection process

ZebOS allows selecting a type of route selection process. We selected the "**rfc-1771-path-selection**" for all routers.

9.4.12 Local QoS class DSCP values

This table summarizes the l-QC DSCP values that are used in each AS.

AS	Local QC	DSCP	b-DSCP	x-DSCP	d-DSCP	b-DS	x-DS	d-DS
AS1	l-QC11	001010	0000-1010	0xa	10	0010-1000	0x28	40
	l-QC12	001100	0000-1100	0xc	12	0011-0000	0x30	48
	l-QC13	001110	0000-1110	0xe	14	0011-1000	0x38	56
	l-QC14	000000	0000-0000	0x0	0	0000-0000	0x0	0
AS2	l-QC21	010010	0001-0010	0x12	18	0100-0100	0x48	72
	l-QC22	010100	0001-0100	0x14	20	0101-0000	0x50	80
	l-QC23	010110	0001-0110	0x16	22	0101-1000	0x58	88
	l-QC24	000000	0000-0000	0x0	0	0000-0000	0x0	0
AS3	l-QC31	011010	0001-1010	0x1a	26	0110-1000	0x68	104
	l-QC32	011100	0001-1100	0x1c	28	0111-0000	0x70	112
	l-QC33	011110	0001-1110	0x1e	30	0111-1000	0x78	120
	l-QC34	000000	0000-0000	0x0	0	0000-0000	0x0	0

AS4	l-QC41	100010	0010-0010	0x22	34	1000-1000	0x88	136
	l-QC42	100100	0010-0100	0x24	36	1001-0000	0x90	144
	l-QC43	100110	0010-0110	0x26	38	1001-1000	0x98	152
	l-QC44	000000	0000-0000	0x0	0	0000-0000	0x0	0
AS5	l-QC51	101010	0010-1010	0x2a	42	1010-1000	0xa8	168
	l-QC52	101100	0010-1100	0x2c	44	1011-0000	0xb0	176
	l-QC53	101110	0010-1110	0x2e	46	1011-1000	0xb8	184
	l-QC54	000000	0000-0000	0x0	0	0000-0000	0x0	0
AS6	l-QC61	110010	0011-0010	0x32	50	1100-1000	0xc8	200
	l-QC62	110100	0011-0100	0x34	52	1101-0000	0xd0	208
	l-QC63	110110	0011-0110	0x36	54	1101-1000	0xd8	216
	l-QC64	000000	0000-0000	0x0	0	0000-0000	0x0	0
AS7	l-QC71	111010	0011-1010	0x3a	58	1110-1000	0xe8	232
	l-QC72	111100	0011-1100	0x3c	60	1111-0000	0xf0	240
	l-QC73	111110	0011-1110	0x3e	62	1111-1000	0xf8	248
	l-QC74	000000	0000-0000	0x0	0	0000-0000	0x0	0
AS8	l-QC11	001010	0000-1010	0xa	10	0010-1000	0x28	40
	l-QC12	001100	0000-1100	0xc	12	0011-0000	0x30	48
	l-QC13	001110	0000-1110	0xe	14	0011-1000	0x38	56
	l-QC14	000000	0000-0000	0x0	0	0000-0000	0x0	0

Table 30: l-QC DSCP values

9.4.13 Inter-domain Meta-QoS-classes DSCP values

This table summarizes DSCP values used between ASs in order to signal meta-QoS-classes. These values are used in both directions (i.e. upstream and downstream).

AS-AS	MC	b-DSCP	b-DSCP	x-DSCP	d-DSCP	b-DS	x-DS	d-DS
AS1-AS2	MC1	011010	0001-1010	0x1a	26	01101000	0x68	104
	MC2	011100	0001-1100	0x1c	28	01110000	0x70	112
	MC3	011110	0001-1110	0x1e	30	01111000	0x78	120
	MC4	000000	0000-0000	0x0	0	00000000	0x0	0
AS1-AS3	MC1	100010	0010-0010	0x22	34	10001000	0x88	136
	MC2	100100	0010-0100	0x24	36	10010000	0x90	144
	MC3	100110	0010-0110	0x26	38	10011000	0x98	152
	MC4	000000	0000-0000	0x0	0	00000000	0x0	0
AS2-AS4	MC1	110010	0011-0010	0x32	50	11001000	0xc8	200
	MC2	110100	0011-0100	0x34	52	11010000	0xd0	208
	MC3	110110	0011-0110	0x36	54	11011000	0xd8	216
	MC4	000000	0000-0000	0x0	0	00000000	0x0	0
AS7-AS5	MC1	111010	0011-1010	0x3a	58	11101000	0xe8	232
	MC2	111100	0011-1100	0x3c	60	11110000	0xf0	240
	MC3	111110	0011-1110	0x3e	62	11111000	0xf8	248
	MC4	000000	0000-0000	0x0	0	00000000	0x0	0
AS3-AS4	MC1	111010	0011-1010	0x3a	58	11101000	0xe8	232
	MC2	111100	0011-1100	0x3c	60	11110000	0xf0	240
	MC3	111110	0011-1110	0x3e	62	11111000	0xf8	248
	MC4	000000	0000-0000	0x0	0	00000000	0x0	0
AS4-AS5	MC1	001010	0000-1010	0xa	10	00101000	0x28	40
	MC2	001100	0000-1100	0xc	12	00110000	0x30	48
	MC3	001110	0000-1110	0xe	14	00111000	0x38	56
	MC4	000000	0000-0000	0x0	0	00000000	0x0	0
AS4-AS6	MC1	010010	0001-0010	0x12	18	01001000	0x48	72
	MC2	010100	0001-0100	0x14	20	01010000	0x50	80
	MC3	010110	0001-0110	0x16	22	01011000	0x58	88
	MC4	000000	0000-0000	0x0	0	00000000	0x0	0
AS6-AS7	MC1	101010	0010-1010	0x2a	42	10101000	0xa8	168
	MC2	101100	0010-1100	0x2c	44	10110000	0xb0	176

Copyright © MESCAL Consortium, July 2005

	MC3	101110	0010-1110	0x2e	46	10111000	0xb8	184
	MC4	000000	0000-0000	0x0	0	00000000	0x0	0
AS6-AS8	MC1	111010	0011-1010	0x3a	58	11101000	0xe8	232
	MC2	111100	0011-1100	0x3c	60	11110000	0xf0	240
	MC3	111110	0011-1110	0x3e	62	11111000	0xf8	248
	MC4	000000	0000-0000	0x0	0	00000000	0x0	0

Table 31: Inter-domain meta-QoS-class DSCP values

9.4.14 Bandwidth thresholds per Meta-QoS-class

This table illustrates the amount of bandwidth that is negotiated between two adjacent ASs and per direction. This amount is expressed in Mbit/s.

AS-AS	MC	BW (Mbit/s) →	BW (Mbit/s) ←
AS1-AS2	MC1	1	1
	MC2	1	1
	MC3	1	1
	MC4	Maximum is 5 Min is 2	Maximum is 5 Min is 2
AS1-AS3	MC1	1	1
	MC2	1	1
	MC3	1	1
	MC4	Maximum is 5 Min is 2	Maximum is 5 Min is 2
AS2-AS4	MC1	1	1
	MC2	1	1
	MC3	1	1
	MC4	Maximum is 7 Min is 4	Maximum is 7 Min is 4
AS7-AS5	MC1	1	1
	MC2	1	1
	MC3	1	1
	MC4	Maximum is 5 Min is 2	Maximum is 5 Min is 2
AS3-AS4	MC1	1	1
	MC2	1	1
	MC3	1	1
	MC4	Maximum is 7 Min is 4	Maximum is 7 Min is 4
AS4-AS5	MC1	1	1
	MC2	1	1
	MC3	1	1
	MC4	Maximum is 7 Min is 4	Maximum is 7 Min is 4
AS4-AS6	MC1	3	3
	MC2	3	3
	MC3	3	3
	MC4	Max is 10 Min is 1	Max is 10 Min is 1
AS6-AS7	MC1	1	1
	MC2	1	1
	MC3	1	1
	MC4	Maximum is 5 Min is 2	Maximum is 5 Min is 2
AS6-AS8	MC1	1	1
	MC2	1	1
	MC3	1	1

	MC4	Maximum is 5 Min is 2	Maximum is 5 Min is 2

Table 32: Bandwidth threshold per meta-QoS-class and per pSLS

9.4.15 Maximum bandwidth per local-QoS-class

This table summarizes the amount of bandwidth allocated per local QoS class. AS4 only is concerned since it is the sole AS in the platform to have intra-domain links.

AS	l-QC	BW in Mbit/s
AS4	l-QC1	5
	l-QC2	5
	l-QC3	5
	l-QC4	Maximum is 20 Min is 5

Table 33: Bandwidth threshold per local-QoS-class

9.4.16 DiffServ-related configuration

The implementation of the classes of service in the testbed will be achieved thanks to the activation of the HTB (Hierarchical Token Bucket), or the priority queuing discipline coupled with a HTB.

Preliminary tests we achieved shown that the Linux CBQ implementation had difficulties to handle more than 1.5Mbit/s of IP traffic. In addition, the TBF Linux implementation has a 1Mbit/s limitation.

In order to ease the configuration and the debugging operations, the following structure is followed for all routers present in the testbed.

Two files are created for each interface: qsi-ethx and qsHTB-ethx.

- **qsi-ethx**: this file contains the ingress related DiffServ policy configuration
- **qsHTB-ethx**: this file contains the egress related DiffServ policy configuration

In addition, the following files are used to execute the configuration of all interfaces.

- **qsi**: This script launches the ingress related DiffServ policy configuration for all relevant interfaces.
- **qse**: This script launches the egress related DiffServ policy configuration for all relevant interfaces.
- **qsa**: This script launches the ingress and the egress related DiffServ policy configuration for all relevant interfaces.
- **qsdel**: This script deletes all ingress and egress policies.

Below are listed some of these files that are used to configure DiffServ policies on **MESCAL110**:

9.4.16.1 qsa

```
#!/bin/bash

HOME_DSMARK=/home/mescal/scripts/DSmarking
$HOME_DSMARK/qsdel
$HOME_DSMARK/qsi
$HOME_DSMARK/qse
```

9.4.16.2 qsi

```
#!/bin/bash
```

```

HOME_DSMARK=/home/mescal/scripts/DSmarking
$HOME_DSMARK/qsi-eth1
$HOME_DSMARK/qsi-eth2
```

9.4.16.3 qse

```
#!/bin/bash

HOME_DSMARK=/home/mescal/scripts/DSmarking
$HOME_DSMARK/qshtb-eth1
$HOME_DSMARK/qshtb-eth2
```

9.4.16.4 qsdel

```
#!/bin/bash
tc qdisc del dev eth1 root
tc qdisc del dev eth2 root
tc qdisc del dev eth1 ingress
tc qdisc del dev eth2 ingress
```

9.4.16.5 qsi-eth1

```
#!/bin/bash

####
#Interfaces AS1-AS2
#______________________________
INGRESS=eth1

####
# Masks
#______________________________
MASK1=0xfc
MASK2=0x03
MASK3=0xff

####
# Local QoS classes
#______________________________
lQC1=0x28
lQC2=0x30
lQC3=0x38
lQC0=0x00

###
# Meta-QoS-classes as signaled by peers
#______________________________
ICP210=0x00
ICP211=0x68
ICP212=0x70
ICP213=0x78

####
# Meta-QoS-classes as should be signaled if similar codes are
# used in both ways
#______________________________
ICP120=0x00
ICP121=0x68
ICP122=0x70
ICP123=0x78
```

```
####
# Rates for policing purposes
#_____________________________
RATE_Total=10Mbit
RATE_ICP210=2.5Mbit
RATE_ICP211=2.5Mbit
RATE_ICP212=2.5Mbit
RATE_ICP213=2.5Mbit

####
# Attach an ingress qdisc to the $INGRESS interfaces
#_____________________________
echo $INGRESS "ingress policies configuration started........."
tc qdisc add dev $INGRESS handle ffff: ingress

#####
# Definition of filter that will put MC1 traffic in the relevant
# class
#_____________________________
tc filter add dev $INGRESS parent ffff: protocol ip prio 1 u32 \
match ip tos $ICP211 $MASK1 \
police rate $RATE_ICP211 buffer 10k \
drop flowid :1

#####
# Definition of filter that will put MC2 traffic in the relevant
# class. This class is dedicated to TCP traffic
#_____________________________
tc filter add dev $INGRESS parent ffff: protocol ip prio 1 u32 \
match ip tos $ICP212 $MASK1 \
police rate $RATE_ICP212 buffer 10k \
drop flowid :2
#match ip protocol 6 0xff \

#####
# Definition of filter that will put MC3 traffic in the relevant
# class. This class is dedicated to UDP traffic
#_____________________________
tc filter add dev $INGRESS parent ffff: protocol ip prio 1 u32 \
match ip tos $ICP213 $MASK1 \
police rate $RATE_ICP213 buffer 10k \
drop flowid :3

#####
# Definition of filter that will put MC0 traffic in the relevant
# class.
#_____________________________
tc filter add dev $INGRESS parent ffff: protocol ip prio 1 u32 \
match ip tos $ICP210 $MASK1 \
flowid :4

#####
# Definition of filter that will drop all other types of traffic
#_____________________________
tc filter add dev $INGRESS parent ffff: protocol ip prio 2 u32 match ip tos
0x0 0x0 police mtu 1 drop flowid :4

echo $INGRESS "....................................finished"
```

9.4.16.6 qsHTB-eth1

```
#!/bin/bash

####
#Interfaces AS1-AS2
#_____________________________
EGRESS=eth1

####
# Masks
#_____________________________
MASK1=0xfc
MASK2=0x03
MASK3=0xff

####
# Local QoS classes
#_____________________________
lQC1=0x28
lQC2=0x30
lQC3=0x38
lQC0=0x00

###
# Meta-QoS-classes as signaled by peers
#_____________________________
ICP210=0x00
ICP211=0x68
ICP212=0x70
ICP213=0x78

####
# Meta-QoS-classes as should be signaled to peers if similar codes are
# used in both ways
#_____________________________
ICP120=0x00
ICP121=0x68
ICP122=0x70
ICP123=0x78

ICP130=0x00
ICP131=0x88
ICP132=0x90
ICP133=0x98
####
# Rates for policing purposes
#_____________________________
RATE_Total=10Mbit

RATE_ICP120=2500Kbit
RATE_ICP121=2500Kbit
RATE_ICP122=2500Kbit
RATE_ICP123=2500Kbit

echo $EGRESS "egress policies configuration started....."
#####
# Attach a dsmark to the eth1 interface
#_____________________________
tc qdisc add dev $EGRESS handle 1:0 root dsmark indices 8
```

```
#####
# Definition of four classes: MC1, MC2, MC3 and MC0
#_____________________________
tc class change dev $EGRESS classid 1:1 dsmark mask $MASK2 value $ICP121
tc class change dev $EGRESS classid 1:2 dsmark mask $MASK2 value $ICP122
tc class change dev $EGRESS classid 1:3 dsmark mask $MASK2 value $ICP123
tc class change dev $EGRESS classid 1:4 dsmark mask $MASK2 value $ICP120


####
# Definition of filters that will be invoked in order to put the ingress
# traffic in the right class
#_____________________________
tc filter add dev $EGRESS parent 1:0 prio 1 prot ip tcindex pass_on
tc filter add dev $EGRESS parent 1:0 prio 1 prot ip handle 1 tcindex
classid 1:1
tc filter add dev $EGRESS parent 1:0 prio 1 prot ip handle 2 tcindex
classid 1:2
tc filter add dev $EGRESS parent 1:0 prio 1 prot ip handle 3 tcindex
classid 1:3
tc filter add dev $EGRESS parent 1:0 prio 1 prot ip handle 4 tcindex
classid 1:4

####
# Definition of filters that will be invoked in order to put the local
# traffic in the right class
#_____________________________
tc filter add dev $EGRESS parent 1:0 protocol ip prio 3 u32 match ip tos
$lQC1 $MASK1 classid 1:1
tc filter add dev $EGRESS parent 1:0 protocol ip prio 3 u32 match ip tos
$lQC2 $MASK1 classid 1:2
tc filter add dev $EGRESS parent 1:0 protocol ip prio 3 u32 match ip tos
$lQC3 $MASK1 classid 1:3
tc filter add dev $EGRESS parent 1:0 protocol ip prio 3 u32 match ip tos
$lQC0 $MASK1 classid 1:4

#####
# Definition of filters that will be used in order to put the traffic
# generated by local interfaces in the relevant egress classes
# Must add other realms if used to identify local interfaces
# Case of 1.1.1.0/30 and 1.1.1.4/30 realm
#_____________________________
tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP121 $MASK1 match ip src 1.1.1.1/30 classid 1:1
tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP122 $MASK1 match ip src 1.1.1.1/30 classid 1:2
tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP123 $MASK1 match ip src 1.1.1.1/30 classid 1:3
tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP120 $MASK1 match ip src 1.1.1.1/30 classid 1:4

tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP121 $MASK1 match ip src 1.1.1.5/30 classid 1:1
tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP122 $MASK1 match ip src 1.1.1.5/30 classid 1:2
tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP123 $MASK1 match ip src 1.1.1.5/30 classid 1:3
tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP120 $MASK1 match ip src 1.1.1.5/30 classid 1:4
```

```
tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP131 $MASK1 match ip src 1.1.1.1/30 classid 1:1
tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP132 $MASK1 match ip src 1.1.1.1/30 classid 1:2
tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP133 $MASK1 match ip src 1.1.1.1/30 classid 1:3
tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP130 $MASK1 match ip src 1.1.1.1/30 classid 1:4

tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP131 $MASK1 match ip src 1.1.1.5/30 classid 1:1
tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP132 $MASK1 match ip src 1.1.1.5/30 classid 1:2
tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP133 $MASK1 match ip src 1.1.1.5/30 classid 1:3
tc filter add dev $EGRESS parent 1:0 protocol ip prio 4 u32 match ip tos
$ICP130 $MASK1 match ip src 1.1.1.5/30 classid 1:4
####
# A generic filter that will put other traffic in the BE class
#______________________________
tc filter add dev $EGRESS parent 1:0 prio 5 prot ip handle 0 tcindex
classid 1:4

####
# Definition of an HTB qdisc that is used to simulate a virtual link
#______________________________
tc qdisc add dev $EGRESS parent 1:0 handle 2:0 htb default 1
tc class add dev $EGRESS parent 2:0 classid 2:1 htb rate $RATE_Total burst
15kb

####
# Definition of an HTB qdisc that will be used to share bw between
# classes
#______________________________
tc qdisc add dev $EGRESS parent 2:1 handle 3:0 htb default 1
tc class add dev $EGRESS parent 3:0 classid 3:1 htb rate $RATE_Total burst
10kb

####
# Definition of an HTB class in order to shape the MC1 traffic.
# A pfifo/bfifo/sfq is attached to this class
#______________________________
tc class add dev $EGRESS parent 3:1 classid 3:10 htb rate $RATE_ICP121
burst 1500b prio 0
tc qdisc add dev $EGRESS parent 3:10 handle a:0 pfifo limit 128

#####
# Definition of an HTB class in order to shape the MC2 traffic.
# A pfifo/bfifo/sfq is attached to this class
#______________________________
tc class add dev $EGRESS parent 3:1 classid 3:11 htb rate $RATE_ICP122
burst 1500b prio 1
tc qdisc add dev $EGRESS parent 3:11 handle b:0 pfifo limit 128

####
# Definition of an HTB class in order to shape the MC3 traffic.
# A pfifo/bfifo/sfq is attached to this class
#______________________________
tc class add dev $EGRESS parent 3:1 classid 3:12 htb rate $RATE_ICP123
burst 1500b prio 2
tc qdisc add dev $EGRESS parent 3:12 handle c:0 pfifo limit 128
```

Copyright © MESCAL Consortium, July 2005

```
####
# Definition of an HTB class in order to shape the MC0 traffic.
# A pfifo/bfifo/sfq is attached to this class
#______________________________
tc class add dev $EGRESS parent 3:1 classid 3:13 htb rate $RATE_ICP120 ceil
$RATE_Total burst 1500b prio 3
tc qdisc add dev $EGRESS parent 3:13 handle d:0 pfifo limit 128

#####
# Definition of filter that will put the traffic in the relevant class
#______________________________
tc filter add dev $EGRESS parent 3:0 protocol ip prio 1 handle 1 tcindex
classid 3:10
tc filter add dev $EGRESS parent 3:0 protocol ip prio 1 handle 2 tcindex
classid 3:11
tc filter add dev $EGRESS parent 3:0 protocol ip prio 1 handle 3 tcindex
classid 3:12
tc filter add dev $EGRESS parent 3:0 protocol ip prio 1 handle 4 tcindex
classid 3:13

echo $EGRESS "................................finished"
```

9.4.17 Backup

In order to limit the impact of a system failure or a possible configuration error, we put in place a two-level backup procedure, which save most of the configuration files of the testbed.

The shared directory **/home/mescal/backup** has been created in **PC-admin** for this purpose. This folder is mounted in every router under: **/mnt/backup**. Routers data are saved using the **backup** command, which can be executed from each router.

In other hand, the **PC Admin** data is saved in **MESCAL110**.

On **PC-admin** side the "**backupall**" command will achieve a backup operation of all routers.

The **/home/mescal/backup** folder contains one sub-folder per router identified with the hostname of the router.

This folder contains again sub-folders containing data saved during a single backup operation. The name of each of these latter folders respects the following structure: **xxxx-DATE-TIME.** "**xxxx**" is set to "local" or "global" depending on the type of backup operation invoked. When backups are achieved with the **backup** command from a single router, **xxxx** takes the value "local" otherwise it is a general backup and it takes the value "global".

- The **backup** command saves:
- The **DSmarking** folder
- The **Zebos.conf** file
- The **hosts** file

The second backup level consists in saving all **PC-admin** data in **MESCAL110** router. The following data are saved:

- The **common** folder
- The **scripts** folder
- The **hosts** file
- The **backup** folder

9.4.18 Logs

In order to verify quickly that all routers are correctly running (interface configured and services launched), every router is configured to report its BGP (bgpd and nsm daemons) and TC status.

A **`crontab`** list has been configured in each router that triggers a reporting every 10 min.

An example of the log file is provided below (example of **`MESCAL110`** router):

```
Tue May 25 17:21:00 CEST 2004-MESCAL110: bgpd running...
Tue May 25 17:21:00 CEST 2004-MESCAL110: nsm running...
Tue May 25 17:21:01 CEST 2004-MESCAL110: Warning An HTB root hasn't been
configured in interface lo
Tue May 25 17:21:01 CEST 2004-MESCAL110: a Warning An HTB class hasn't been
configured for the interface lo
Tue May 25 17:21:01 CEST 2004-MESCAL110: Warning A DSMARK qdisc hasn't been
configured for the interface lo
Tue May 25 17:21:01 CEST 2004-MESCAL110: Number of configured HTB root
classes is OK for interface eth0
Tue May 25 17:21:01 CEST 2004-MESCAL110: Number of configured HTB classes
is OK for the interface eth0
Tue May 25 17:21:01 CEST 2004-MESCAL110: Number of configured DSMARK
classes is OK for the interface eth0
Tue May 25 17:21:01 CEST 2004-MESCAL110: Number of configured HTB root
classes is OK for interface eth1
Tue May 25 17:21:01 CEST 2004-MESCAL110: Number of configured HTB classes
is OK for the interface eth1
Tue May 25 17:21:01 CEST 2004-MESCAL110: Number of configured DSMARK
classes is OK for the interface eth1
Tue May 25 17:21:01 CEST 2004-MESCAL110: Warning An HTB root hasn't been
configured in interface eth2
Tue May 25 17:21:01 CEST 2004-MESCAL110: a Warning An HTB class hasn't been
configured for the interface eth2
Tue May 25 17:21:01 CEST 2004-MESCAL110: Warning A DSMARK  qdisc hasn't
been configured for the interface eth2
Tue May 25 17:21:01 CEST 2004-MESCAL110: Warning An HTB root hasn't been
configured in interface eth3
Tue May 25 17:21:01 CEST 2004-MESCAL110: a Warning An HTB class hasn't been
configured for the interface eth3
Tue May 25 17:21:01 CEST 2004-MESCAL110: Warning A DSMARK  qdisc hasn't
been configured for the interface eth3
Tue May 25 17:21:01 CEST 2004-MESCAL110: Warning An HTB root hasn't been
configured in interface eth4
Tue May 25 17:21:01 CEST 2004-MESCAL110: a Warning An HTB class hasn't been
configured for the interface eth4
Tue May 25 17:21:01 CEST 2004-MESCAL110: Warning A DSMARK  qdisc hasn't
been configured for the interface eth4
Tue May 25 17:21:01 CEST 2004-MESCAL110: Warning An HTB root hasn't been
configured in interface eth5
Tue May 25 17:21:01 CEST 2004-MESCAL110: a Warning An HTB class hasn't been
configured for the interface eth5
Tue May 25 17:21:01 CEST 2004-MESCAL110: Warning A DSMARK qdisc hasn't been
configured for the interface eth5
Tue May 25 17:21:01 CEST 2004-MESCAL110: Warning An HTB root hasn't been
configured in interface eth6
Tue May 25 17:21:01 CEST 2004-MESCAL110: a Warning An HTB class hasn't been
configured for the interface eth6
Tue May 25 17:21:01 CEST 2004-MESCAL110: Warning A DSMARK qdisc hasn't been
configured for the interface eth6
```

Copyright © MESCAL Consortium, July 2005

9.4.19 Check the sanity of the test bed

In order to verify that all routers in the testbed are configured as expected, two scripts that check the status of the configuration have been developed: **`pfc`** and **`pfcheck`**. These scripts must be executed from **`PC-admin`**.

9.4.20 Configuration scripts

The table below lists useful scripts that are used for configuring the testbed. A description of the service they provide is also given.

Location	Script	description
Routers	pcm	This script allows mounting the log, transfer, common and backup folders.
	pcu	This script allows un-mounting the log, transfer, common and backup folders.
	qsa	This script configures all ingress and egress policies for a given router.
	qsdel	This script deletes all ingress and egress policies for a given router.
	qse	This script configures egress policies for a given router.
	qsi	This script configures ingress policies for a given router.
	qsi-ethj	This script configures ingress policies of the interface ethj of a given router.
	qshtb-ethj	This script configures egress policies of the interface ethj of a given router.
	qsstat	This script displays real-time egress TC related statistics
	qsstati	This script displays real-time ingress TC related statistics
	ifstat	This script displays the interface's statistics as reported by the kernel. If the argument "-s" is given the script sets to zero all statistics.
	bgpstart	This script starts the bgpd and nsm daemons
	bgpstop	This script stops the bgpd and nsm daemons
	isalive	This scripts returns the status of the bgpd and nsm daemons
	svty	This script launches the VTY. If the argument "-l" is given, the scripts loads the configuration stored in /usr/local/etc/ZeboS.conf
	pfgping	This scripts displays reachability information of all operational interfaces
	cgping	This scripts displays reachability information for all customers
	confstatus	This scripts returns the status of the configuration of a router
	backup	This script achieved a backup operation.
	pcabackup	This script can only be executed from MESCAL110. It achieved a PC-admin backup operation
	clean	This script recursively deletes all files ending with "~" in the /home/mescal folder.
PC-admin	lup	This script activates a given interface of a router
	ldown	This script deactivates a given interface of a router
	lstatus	This script returns the status of all routers interfaces

	rtrupdate	This script achieves an automatic update for a remote router
	updateall	This script achieves an automatic update for all routers
	rtrbackup	This script achieves an automatic backup for a remote router
	backupall	This script achieves an automatic backup for all routers
	pbackup	This script achieves a PC-admin backup
	pfc	This script displays the status of the testbed configuration without details and based on differed log files.
	pfcheck	This script displays the detailed status of the testbed configuration.
	initqos	This script initializes the QoS configuration of all routers
	delqos	This script deletes the QoS configuration of all routers
	egqos	This script initializes the egress QoS configuration of all routers
	ingqos	This script initializes the ingress QoS configuration of all routers
	iifstat	This script initializes the interface statistics of all routers
	lcap	This script configures the capabilities of Ethernet cards: lcap -s: restarts a negotiation between two back to back interfaces lcap -10: sets an interface to 10Mbit full duplex. lcap -100: sets an interfaces to 100Mbit full duplex.

Table 34: List of useful scripts

9.5 Specific Configuration for phase 2

In the phase 2, the testbed architecture will be the same. The major difference will be the q-BGP activation.

The following scripts have been created for phase 2 purposes:

- initqbgp: this scripts initialises all q-BGP routers.
- stopqbgp: this scripts stops all q-BGP routers.
- qbgpstart: this scripts initialises a q-BGP router.
- qbgpstop: this scripts stops a q-BGP router.
- qisalive: this scripts tests if a q-BGP process is active.
- qsvty: this scripts launches an VTTY terminal for configuring q-BGP router.
- createPSLS: this scripts manages pSLSs.
- psls2qbgp: this scripts configures q-BGP router according to a given pSLS.

9.6 Specific Configuration for phase 3

In this phase, the configuration will be the same as for the phase 2. Nevertheless, we will install a PCE (Path Computation Element) in every AS. The table below shows the locations of these PCEs:

PCE identifier	Router ID	IP address	PCSId
PCE11	MESCAL110	14.0.0.2	11.11.11.11
PCE21	MESCAL210	24.0.0.2	21.21.21.21
PCE31	MESCAL310	34.0.0.2	31.31.31.31
PCE41	MESCAL410	44.0.0.2	41.41.41.41
PCE51	MESCAL510	54.0.0.2	51.51.51.51
PCE61	MESCAL610	64.0.0.2	61.61.61.61
PCE71	MESCAL710	74.0.0.2	71.71.71.71
PCE81	MESCAL810	84.0.0.2	81.81.81.81

Table 35: PCE locations