AGAVE

A liGhtweight Approach for
Viable End-to-end IP-based QoS Services

IST-027609

D4.1: Test Specification and Experimentation Plan

Document Identifier: AGAVE/WP4/D4.1	
Deliverable Type: Report **Deliverable Nature:** Public	**Contractual Date:** M18 **Actual Date:** 6 July 2007

Editor:	Luigi Iannone, UCL.be
Authors:	*TID:* Maria L. Garcia Osma, Jorge Rodriguez Sanchez *FTR&D:* Bruno Decraene, Pierre Alain Coste *Algo:* Eleni Mykoniati, Kostas Kavidopoulos, Panos Georgatsos *UCL.uk:* David Griffin *UniS:* Ning Wang, Mina Amin, Kin-Hon Ho *UCL.be:* Luigi Iannone, Bruno Quoitin, Olivier Bonaventure

Abstract: This document specifies the experimentation activities to be undertaken by the AGAVE project. The tests fall into three main themes: *Network Plane Engineering Experiments* for testing the solutions and mechanisms specified for realising Network Planes within IP Network Provider domains; *Inter-domain Routing Experiments* for testing inter-domain routing and resilience mechanisms, algorithms and protocols for realising Parallel Internets across multiple network providers; and *Integrated Parallel Internet Engineering experiments* investigating the interworking and cooperation of intra- and inter-domain techniques to provide end-to-end service differentiation.

The experimentation activities are described in terms of: their objectives, the methodology adopted, the test-execution environment, control parameters, and performance metrics, based on which the validity and the performance of the entity/component under test will be evaluated.

Keywords: Experimentation, Network Plane, Parallel Internet, Routing, Simulation, Testbed

Telefónica Investigación y Desarrollo	TID	Co-ordinator	Spain
France Telecom Research and Development	FTR&D	Partner	France
Algonet SA	Algo	Partner	Greece
University College London	UCL.uk	Partner	UK
The University of Surrey	UniS	Partner	UK
Université catholique de Louvain	UCL.be	Partner	Belgium

Information Society and Media

Project funded by the European Community under the
“Information Society Technology” Programme (2002-2006)

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

1.1 Experimentation Approach

WP4, *Validation and Experimentation*, undertakes the experimentation activities of the project for assessing the validity and cost-effectiveness of the proposed solutions. It involves the setting-up of the required experimentation infrastructure, testbeds and simulators, the specification of appropriate evaluation scenarios and their execution.

Project experimentation aims at validating, demonstrating and assessing the performance of the solutions specified within the overall AGAVE framework - separation of SP and INP roles, notions of CPAs, NPs and PIs and relevant interface models, agreements handling functions, intra- and inter-domain engineering techniques, algorithms and protocols. To this end, a number of 'proof-of-concept' prototypes, each addressing specific aspects of the proposed framework, are built and evaluated for their functional validity and performance.

Prototype experimentation is carried out in either a testbed or simulated environment, as appropriate for the entity under test. Testbed-oriented experimentation aims at validating and demonstrating, through proof-of-concept implementation and deployment, that the specified functionality can be deployed in physical routers. It mainly concerns lower-level functional aspects such as QoS routing/forwarding or resilience mechanisms. Simulation-based testing allows for more extensive and larger scale experiments -regarding network topologies, traffic patterns and operating variables- to be undertaken than could reasonably, if not practical at all, be performed in physical testbeds. It mainly concerns interface, algorithmic and protocol behaviour aspects and combinations of them.

In the above context, WP4 is concerned with the coordination of the different experimentation activities so that to ensure a consistent evaluation methodology and common sets of input; the setting-up and maintenance of the appropriate testbeds and simulators; the integration of the prototypes implementing essential aspects of the connectivity service provisioning interface and of the Network Planes and Parallel Internets in testbed and simulation platforms; the testing and validation of the functionality and the performance of the developed mechanisms, algorithms and protocols in testbed and simulation platforms; and finally, to the extent possible, the comparison of alternative approaches for engineering Network Planes and Parallel Internets.

In particular, work in WP4 is organized around the following three activities, presenting the main steps of the project experimentation approach;

AC4.1 Specification of Tests: it specifies how the mechanisms, algorithms and protocols developed by WP2 and WP3 are evaluated in order to satisfy the test requirements documented in D2.1 and D3.1. The required testing resources are identified in terms of testing tools, traffic generators, testbed equipment, simulator customized functionality and configuration. Proof of concept test scenarios are produced to validate the connectivity service provisioning interface for selected service use cases. Reference topologies and traffic characteristics are defined for simulation-based experimentation, taking into account the service connectivity requirements. As several techniques are developed within WP3, it is important to be able to compare their performance by using the same topologies and traffic characteristics. Initial specification of the experimentation platform and tests has been documented in I4.1.

> *AC4.2 Testbed-based Prototype Evaluation;* it organizes and maintains testbed platforms, integrating the prototypes developed within WP2 and WP3 activities. It undertakes the testbed-based tests defined by AC4.1 in order to verify the feasibility of the approach. The test scenarios carried out within AC4.2 are used for demonstrating the capabilities of the Parallel Internets and the benefits of the AGAVE approach.
>
> *AC4.3 Performance Evaluation of Simulation-based Protocols and Algorithms;* it undertakes the simulation-based tests defined by AC4.1, in order to evaluate the performance of the mechanisms, algorithms and protocols developed by WP3. Simulation-based tests focus on assessing scalability, stability and cost/benefit performance aspects.

1.2 Scope and Structure of the Deliverable

The present document includes the specification of the experimentation activities to be undertaken, in terms of its objectives, tests involved and platforms, testbeds and simulators, based on which the tests will be executed.

Project experimentation is distinguished into three lines, following the logical classification of the implementation activities per major functional area, as reported in the implementation plan [D3.1].

NP engineering experiments (chapter 2): It describes experiments for testing the solutions and mechanisms specified for realising Network Planes in an INP domain. In particular, it includes:

> MTR (Multi-Topology Routing) experimentation in a simulation environment: MTR is a means for engineering several routes to the same destinations and thus for engineering route-differentiated Network Planes over the same physical topology, based on the definition of logical topologies, maintaining numerous adjacencies, etc.
>
> MRDV (Multi-Path Routing with Dynamic Variance) experimentation in testbeds and simulation environment: MRDV is a routing technique to enable intra-domain multi-path routing, used for realising NPs with different QoS levels, reacting to dynamically measured congestion levels.
>
> NP Emulation Platform experimentation: The platform presents a snapshot of an integrated IP Network Provider system, embodying the essential aspects of the project work –CPAs, NPs, PIs and respective engineering guidelines. It assumes Diffserv/MPLS IP network capabilities for realising NPs with different QoS levels.

Inter-domain routing experiments: It describes experiments for testing the specified inter-domain routing and resilience mechanisms, algorithms and protocols for realising PIs. In particular, it includes:

> q-BGP experimentation in a simulation environment: q-BGP is a means, enhancing traditional BGP, that contributes to realising PIs with different QoS levels.
>
> Resilience-aware BGP/IGP TE interactions in a simulation environment: an approach for maintaining optimised traffic distribution condition and QoS assurance on network link failures.
>
> BGP Planned Maintenance experimentation in testbed: an approach for improving availability on disruptions due to maintenance by carrying out planned BGP session shut-downs.
>
> IP tunnelling experimentation in testbeds and simulation environment: an approach for inter-domain routing and traffic control through IP tunnelling mechanisms between cooperative remote domains; the work is being aligned with the ongoing work at IETF/IRTF on locator/identifier separation.

Integrated PI engineering experiments: It describes experiments regarding the integration of NP engineering and inter-domain techniques in order to provide end-to-end service differentiation. These experiments will check the validity/investigate how a selected subset of the techniques specified can be used to realise NPs and to horizontally bind these NPs, belonging to different INPs, to form Parallel Internets. Experimentation will address the following two concrete use cases:

> PI engineering with overlay routing in a simulation environment: an approach for realising QoS-aware PIs through overlay routing techniques.
>
> PI engineering with MTR and q-BGP in a simulation environment: an approach for realising QoS-aware PIs across MTR-capable domains interconnected by q-BGP.

The above experimentation activities are described in terms of:

- Their objectives.
- The methodology adopted, outlining the different types of tests involved.

- The test-execution environment (testbed equipment and configuration, simulator, topology/traffic generators, software routing daemon etc.).
- Control parameters, set of parameters influencing the operational behaviour of the component/entity under test; they may relate to the network environment or to the operation of the component itself.
- Performance metrics, based on which the validity and the performance of the entity/component under test will be evaluated; these metrics capture in a tangible way the results of the operation of the entity/component under test and the effect that it yields in the network.

The deliverable concludes the work of activity AC4.1. This work will be used by the other WP4 activities in order to integrate the prototypes and actually execute the tests, gather and analyse their results. The deliverable will be superseded by deliverable D4.2, due at the end of the project, which will include the results of the testbed and simulation-based tests specified in this document.

2 NP ENGINEERING EXPERIMENTS

2.1 Multi-Topology Routing

2.1.1 Objectives

The objective of this simulation part is to study the empirical performance of the proposed algorithms for NP engineering using multi-topology IP routing (MTR) protocols. Specifically, we will evaluate how the proposed scheme is able to achieve traffic engineering objectives, such as minimizing maximum link utilization, as well as QoS requirements (e.g. edge-to-edge delay) for each individual NP. Performance comparison will be carried out between the proposed schemes with existing approaches, including:

- Link weight setting with inversed bandwidth capacity;
- The actual link weights (when the GEANT topology [GEANT] is used);
- The link weight optimisation scheme by Fortz et al. [FORT00];
- Multi-paths Routing with Dynamic Variance (MRDV, see section 2.2);
- Optimal performance calculated through linear programming (formulated as the Multi-Commodity Flow problem, which can be solved by the TOTEM toolbox [TOTEM]).

2.1.2 Environments

All the experiments will be performed on top of a standalone simulation platform. As it can be seen in D3.1, there exist two key components in the NP provisioning and maintenance functional block, namely offline link weight setting and online traffic engineering. The Offline TE component in NP provisioning and Maintenance is mainly responsible for MTR configuration, including topology partitioning (optional) and link weight optimisation.

There are two major inputs to be fed into the Offline TE component in order to obtain the optimised MTR configuration:

- The physical network topology;
- The overall traffic matrix coming from the functional block of NP mapping [1].

In our planned experiment, we will use the GEANT set-up (see Section 5) and also the public intra-domain traffic matrices that have been captured in the GEANT network [UHLI06] for simulation purposes.

In addition to the GEANT set-up, random generated network topologies (generated by GT-ITM [GT-ITM] or BRITE [BRITE]) will also be used for performance evaluation in larger sized networks. In this scenario, both POP level and router level topologies will be used for experimentation.

Online traffic engineering is a complementary component to its offline counterpart in NP provisioning. In our experiment for online TE, we will use the same network topology as the offline scenario. In order to evaluate the proposed algorithm for online control of the incoming traffic, an important task is to derive time-dependent traffic dynamics according to the “fixed” traffic matrix. This traffic dynamics will be stored in a dedicated script to be fed into the online TE engine together with the same network topology. As it is described in [UHLI06], traffic matrices are obtained every 15 minutes from network monitoring on top of the GEANT network topology, and this enables us to emulate

[1] This is only an optional input. We will also consider MTR link weight setting without knowing the traffic matrix a priori.

traffic dynamics in a more accurate fashion. By looking at “adjacent” traffic matrices (e.g., four adjacent TMs within one hour), it is possible to generate a script for emulating traffic dynamics during the period. The traffic matrices to be used will be selected during the period between January 1 and April 29, 2005. Finally, the optimised MTR configuration from the offline TE (virtual topologies and MT-IGP link weights and initial traffic assignment to individual topologies) is also fed into the online TE. In the test suite where random network topology is used, the corresponding traffic matrices will also be generated following the patterns of the public GEANT traffic matrices with some random deviations. The overview experiment methodology for offline/online traffic engineering in NP provisioning and maintenance is illustrated in Figure 1.

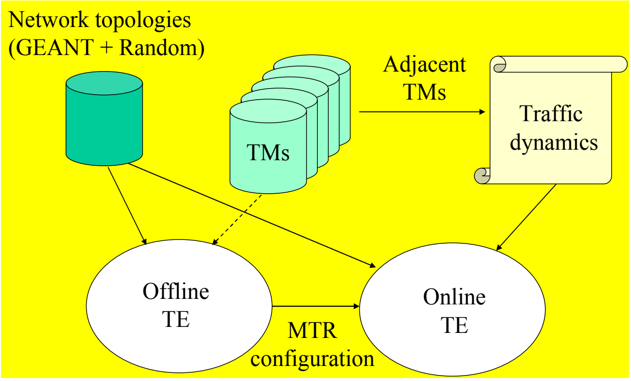


Figure 1 - Illustration of MTR simulation experiments.

2.1.3 Control Parameters

Control Parameters		
The offline TE component	*MAX_LINK_WEIGHT*	The maximum MT-IGP link weight to be set.
	LINK_CAPACITY	Bandwidth capacity of each link. This parameter will only be used when randomly generated topologies are used, as in GEANT the link capacity is known.
	MAX_LINK_WEIGHT	Propagation delay of each link.
	MAX_NUM_TOPOLOGIES	The maximum number of IGP routing topologies to be used for each NP.
	NETWORK_SIZE	The number of nodes and links in the network topology. This parameter will be tuned when random topologies are used for performance evaluation.
	TRFFIC_DEMAND	The volume of traffic demand in the traffic matrix. This parameter will be tuned when random topologies are used for performance evaluation.
	DELAY_BOUND	The maximum edge-to-edge delay required by the NP for each routing topology. This parameter will vary when random topologies are used for performance evaluation.
	Algorithm iteration	Algorithm iteration counter, part of the algorithm software.

The online TE component	*SPLIT_POLICY*	The policy that regulates the behaviour of ingress points for splitting incoming traffic across multiple equivalent routing topologies.
	SPLIT_GRANULARITY	The granularity of traffic split across multiple equivalent routing topologies.
	INITIAL_SPLITTING_RATIO	Initial traffic splitting ratio at the bootstrap phase.
	TRAFFIC_BURST_PATTERN	The patter of creating traffic burst based on real traffic matrices.

2.1.4 Performance metrics

Performance Metrics		
The offline TE component	*DEGREE_OF_INVOLVEMENT*	The number of times a particular link is involved in all routing topologies. This metric is used to indicate the overall path diversity achieved by the offline component.
	MAX_PATH_LENGTH	The max number of hops between ingress-egress points according to the IGP routing in each MTR topology.
	AVG_PATH_LENGTH	The average number of hops between ingress-egress points according to the IGP routing in each MTR topology.
The online TE component	*MAX_UTILISATION*	Maximum link utilization.
	NETWORK_COST	Overall network cost according to piece-wise linear cost function.
	RUNNING_TIME	Time for computing traffic split ratio for each incoming traffic matrix.

2.2 MRDV

2.2.1 Objectives

The work TID has planned for WP4 consists on deploying a Testbed, implementing and testing the proposed extension of MRDV (Multipath Routing with Dynamic Variance) [RAMO02], as well as a new load distribution module, on a simulation tool, and to implement MRDV (both original and extended version) in a router software (Quagga) [QUAGGA].

The first subsection describes the simulations that TID plans to carry out in order to evaluate the performance of the new MRDV (Multipath Routing with Dynamic Variance) extension that supports QoS by means of path differentiation, i.e. DiffRouting (Differentiated Routing) and to study the applicability of this MRDV extension to build Network Planes. The second subsection presents the steps to be followed to implement MRDV in Quagga routers and proposed Testbeds, while the last subsection presents the new MRDV load distribution module.

The objective of the simulations that will be carried out is to verify not only that MRDV can be used to implement Network Planes, but also that the required performance can be obtained by means of this algorithm.

2.2.2 Simulation Methodologies

MRDV will be verified simulating with NS-2 (The Network Simulator) [NS], initially using basic scenarios as the one presented on D3.1 and then experimenting with more sophisticated ones. For the latter simulations it is planned to use two different scenarios, one representing a metropolitan network and the other a core network:

- A scenario based on the Madrid Subway network, as used in [AN], which is a highly meshed metropolitan network.
- The GEANT set-up (see Section 5). By using this scenario, it will be possible to compare the obtained results with those obtained by simulations that employ other mechanisms proposed to implement Network Planes.

As aforementioned, several scenarios have been selected for the simulation activities that have been done and that will be carried out during the timeframe of this WP, namely one basic scenario and two advanced ones.

The first type of simulations have already concluded and have succeeded on verifying that the proposed extension to MRDV algorithm behaves as expected and that its performance fulfils the given requirements.

The following figure represents the basic network, which is composed by 7 core routers and 7 edge routers connected to these, apart from 14 access nodes connected to the edge routers.

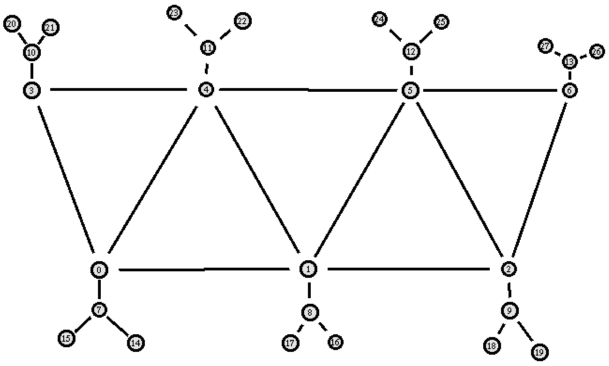


Figure 2 Basic Simulation Scenario.

The simulations have used both TCP and UDP traffic sources, where UDP traffic has been marked as high priority traffic and TCP as best-effort.

The scenario based on the Madrid Subway network, has also been recently simulated in [RODR07]. The scenario comprises 36 nodes that are connected with links that follow the metro lines.

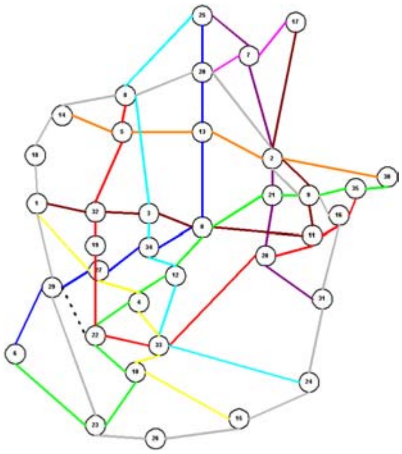


Figure 3 Madrid Metro Network.

The traffic pattern used to feed the topology is composed of two UDP (User Datagram Protocol) sources, one of each traffic class, between each pair of nodes. Both sources generate traffic at the same base rate, which is multiplied by a scaling factor to increase the load over the series of simulations. In order to investigate dynamic traffic patterns it is further assumed that trains travelling through the network generate traffic between themselves and to external networks through gateway nodes.

Simulations in the GEANT set-up use the traffic matrices published on the TOTEM Toolbox website [TOTEM].

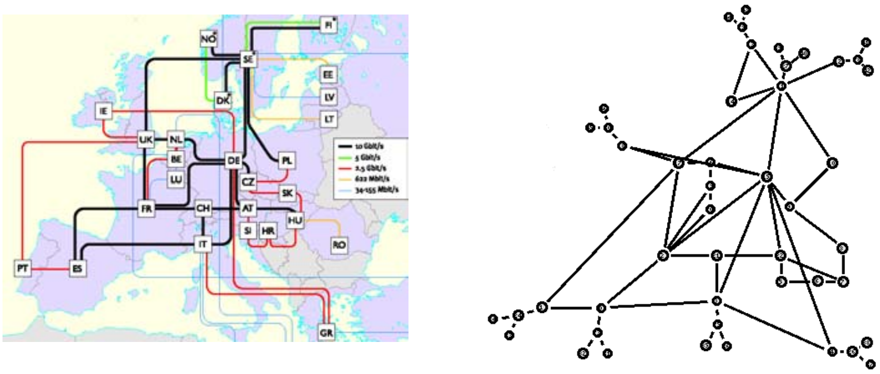


Figure 4 GEANT Synthetic view.

The traffic matrices from the GEANT set-up are going to be aligned with those used in MTR simulations in order to compare both mechanisms. In fact, those that are going to be used are the traffic matrices from 23/04/2005 to 26/04/2005. This way, traffic will be dynamically changing every 15 minutes.

2.2.3 MRDV Implementation on Quagga routers

The original version of MRDV [RAMO02] has never been implemented on Quagga routers. Therefore, this is the first step to take in order to implement the extended version, presented in AGAVE, for NP realization. This overall task is divided in three subtasks:

- Implementation of the original version of MRDV [RAMO02].
- Implementation of the extended version of MRDV.
- Implementation of LAP [CALL05].

In order to avoid loop appearance during the first two subtasks, the implementation is going to be carried out in a very basic scenario. This scenario has already been implemented and it is shown in the following figure:

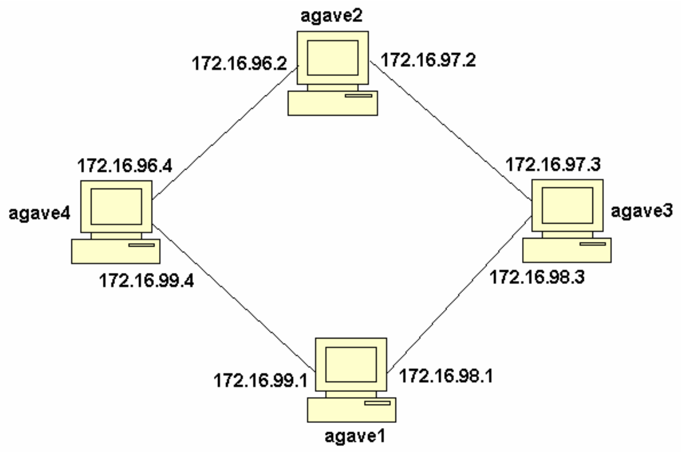


Figure 5 Initial Testbed.

A traffic source will be attached to 'agave4', being the destination 'agave3'. Link costs will be assigned in such a way that the path through 'agave2' is the optimum path and that at a certain link load level, traffic is also routed through 'agave1'. Therefore, it is necessary to be capable of measuring link load from each of the routers implementing MRDV. To do so, a tool called MAPI [MAPI] will be used and will be inserted in the Quagga software.

Once the original version of MRDV is implemented and tested, the extended version will be implemented. The first modification to be made is to differentiate among several QoS traffic classes, which has not been yet implemented on Quagga and then implement the mechanism to be capable to realize NPs. At first, the same testbed (Figure 5) will be used with two traffic classes. This way, it will be possible to observe how the different NPs can be created.

The last subtask is to implement LAP [CALL05]. This subtask is twofold: firstly, primary loop avoidance will be implemented and then, secondary loop avoidance. Primary loop avoidance does not need 'extra' information exchange (i.e. LAPM messages exchange) and, therefore, will be firstly implemented. In both tasks, more meshed topologies must be employed in order to guarantee that loops may appear and that they are successfully removed by LAP. For instance, the topology used in Figure 2 could be a first scenario to test the implementation.

Once the overall mechanism is implemented and tested, more complex topologies may be used.

2.2.4 New NS-2 MRDV Load Distribution Module

Until now, the extended version of MRDV has been implemented considering a traffic load distribution in which traffic from a given traffic class could be split among several paths towards the destination. A new way of distributing traffic in which the traffic from the same class, except for best-effort traffic class, is not split is proposed.

Each node running MRDV in the network should have a list of traffic classes carried by each of its links; this list must include what amount of the whole traffic in the link corresponds to each class. Also, each node should have a Routing Table with the possible links to the destination and the optimum proportions of traffic that should be carried by each link.

The highest class of traffic must be carried using the optimum path. If the proportion of this class is lower than the proportion assigned to the optimum path, spare capacity will be left in the optimum path link. If this spare capacity is greater than the traffic of the second highest level class traffic, this second traffic will be also carried by the optimum path. This way, each class tries to travel along the optimum path. If there is no spare capacity in the optimum path, or it is not enough to carry the whole amount of traffic assigned to one class, the node must look at each of its output interfaces until one is able to carry that traffic. If there were no link able to manage the traffic associated to a given traffic class, the link with greater spare capacity should carry it. This is done for each traffic class; finally, best effort traffic is carried by a link with enough capacity. If there is no link with enough capacity to carry all of the best effort traffic, this is divided between the links with spare capacity.

This new load distribution module will be implemented in NS-2 [NS]. The same simulations as in 2.2.2 will be carried out in order to compare both load distributions.

Some examples that will be useful to understand the proposed mechanism are introduced.

2.2.4.1 Example 1

Supposing there are three different classes of traffic in a link:

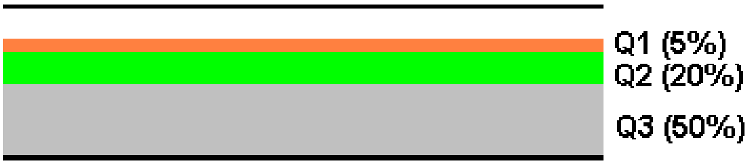


Figure 6 Link with different traffic classes for example 1.

Where Q1 refers to the traffic with higher priority, Q2 is the traffic with second higher priority and Q3 is best-effort traffic.

It could also be supposed that this link could distribute its traffic between other three different links. If the three existing classes were considered as one, the optimal distribution (based on costs) would be the following:

Links	Proportion
L1	40%
L2	15%
L3	10%
L4	10%

Table 1 Optimal distribution based on costs for example 1.

In this case, the Q1 traffic will be carried by L1 (optimum path). Thus, there will be still a 35% of spare capacity for other traffic classes. For this reason, the traffic of class Q2 will be carried also by L1. Still, the optimum path would have a 5% of capacity left. This 5% will be used for best-effort traffic. In the same way, the 15% of L2, the 10% of L3 and the 10% of L4 will be used for the remaining best-effort traffic.

2.2.4.2 Example 2

Supposing there are the same traffic classes as in Example 1 but with different distribution (Figure 7). Where again, Q1 refers to the traffic with higher priority, Q2 is the traffic with second higher priority and Q3 is best-effort traffic.

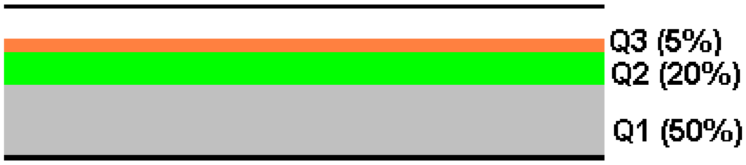


Figure 7 Link with different traffic classes for example 2.

It could also be supposed that this link could distribute its traffic between other three different links. If the three existing classes were considered as one, the optimal distribution (based on costs) would be the same as in the previous example:

Links	Proportion
L1	40%
L2	15%
L3	10%
L4	10%

Table 2 Distribution based on costs for example 2.

In this case, the Q1 traffic will be carried by L1 (optimum path). The traffic from Q1 is 10% bigger than the optimal proportion for L1. However, the Q1 class cannot be split and it must be carried by the optimum path. When traffic of class Q2 has to be assigned a path, all links' optimal proportions are lower than the one of traffic of class Q2. For this reason, Q2 is routed through L2, which is the link with higher traffic proportion (-10% for L1, 15% for L2, 10% for L3 and 10% for L4). The remaining traffic is best effort. Due to the impossibility to split Q1 and Q2 traffic, the proportions for L1 and L2 differ from the optimum in –10% and –5%, respectively. Thus, best effort traffic could be routed either by L3 or L4.

2.2.5 Control Parameters

Control Parameters		
Parameters controlled in MRDV	*Vmax*	The maximum variance permitted in each router.
	K	Controls the influence of load during the variance calculation.

2.2.6 Performance Metrics

Performance Metrics		
Performance Metrics for MRDV	*PACKET LOSS*	Packet loss ratio in the network.
	DELAY	Mean Delay between nodes in the network.
	JITTER	Mean Delay Variance between nodes in the network.

2.3 NP Emulation Platform

2.3.1 Objectives

The NP Emulation Platform (NPEP) provides a ‘snapshot’ of an INP-domain embodying the essential aspects of project work; clear separation of INP and SP roles in terms of CPAs and engineering of INP domains in terms of Network Planes (NPs) and Parallel Internets (PIs) according to business policies regarding service provisioning. It also provides means for generating traffic corresponding to the established CPAs and measuring the performance of the network in accommodating the generated traffic flows. The platform assumes IP networks with Diffserv/MPLS capabilities for realizing NPs. However, its design is modular and alternative IP network technologies/capabilities can be incorporated.

The platform is built with the purpose of validating the concepts and notions developed by the project, for exhibiting the business-driven (rule-based) engineering of INP-domains and for running ‘what-if’ scenarios and comparison tests to assist decision-making in business policies on service provisioning, network upgrades and technology choices, including traffic engineering.

Experimentation aims at validating and demonstrating the use of the NP Emulation Platform:

- By validating the platform, we validate the concepts and notions developed within the project, proving that they can lead to a working system with feasible network configurations that allow the honouring of the established agreements.
- By demonstrating the platform, we exhibit the technology-agnostic abstractions at the business and network layers for managing and engineering an IP network domain (INP perspective) to the end of provisioning and delivering services in the Internet. Furthermore, we verify the capability of the platform to support the execution of ‘what-if’ scenarios assisting decision-making processes.

2.3.2 Functional Validation Tests

- CPA models for specific business cases e.g. for VoIP service provisioning.
- Translation of CPAs to a common information model, based on which the underlying INP functions for fulfilling and assuring the established CPAs operate.
- Network configuration for fulfilling established CPAs according to respective business-driven guidelines based on defined NPs and PIs assuming particular IP network capabilities.
- Performance of the network given the resulting network configuration.
- Ability to run ‘what-if’ scenarios over different topologies, algorithms and CPA mixes – optional.
- Realization of NPs and PIs assuming different IP network capabilities –optional.

2.3.3 Demonstrations

- Technology-agnostic definition of NP engineering rules, NPs and PIs, driving the INP operation.
- Realization of NPs/PIs with specific IP network technology (ies).
- ‘What-if’ scenarios (correlated/comparison tests) for particular decision-making problems – optional.

2.3.4 ‘What-if’ Scenarios

‘What-if’ scenarios aim at assisting well-defined decision-making problems related to network growth, traffic evolution and technology employment. Presently, possibilities for ‘what-if’ scenarios include, but are not exhausted in:

- Impact of requesting new downstream NIAs.
- Impact of accepting requested upstream NIAs.
- Impact of downstream NIA failure(s).
- Impact of accepting CPAs.
- Compatibility of service requirements –feasibility of fulfilling diverse service requirements.
- Criteria/policies for instantiating NPs.
- Criteria/policies for allowing and handling dynamic modification of CPA characteristics e.g. invocations for bandwidth modification within admissible margin.
- Assessment of relative impact of different policy parameters, identification of dependencies.
- Physical resource upgrades so that to minimize cost and operations overhead –minimum required upgrades.
- Reengineering cost-benefit analysis.
- Comparison of alternative IP network for realizing NPs.

The exhaustive analysis and undertaking of 'what-if' scenarios to assist INP decision-making problems is neither the main subject of the project nor feasible within the time and resource limits of the project. The above indicative list of scenarios is put forward primarily for exhibiting the additional value of the NP Emulation Platform.

2.3.5 Experimentation Environment

Experimentation will be carried out in a computer-based environment.

Figure 8 presents an overall view of the NP Emulation Platform. As it can be seen, it consists of (a) components pertinent to project work –interfaces for CPAs, NP engineering guidelines, NPs, PIs, NP provisioning algorithms and (b) generic components of an emulation system –traffic generation, emulation engine, reporting facilities.

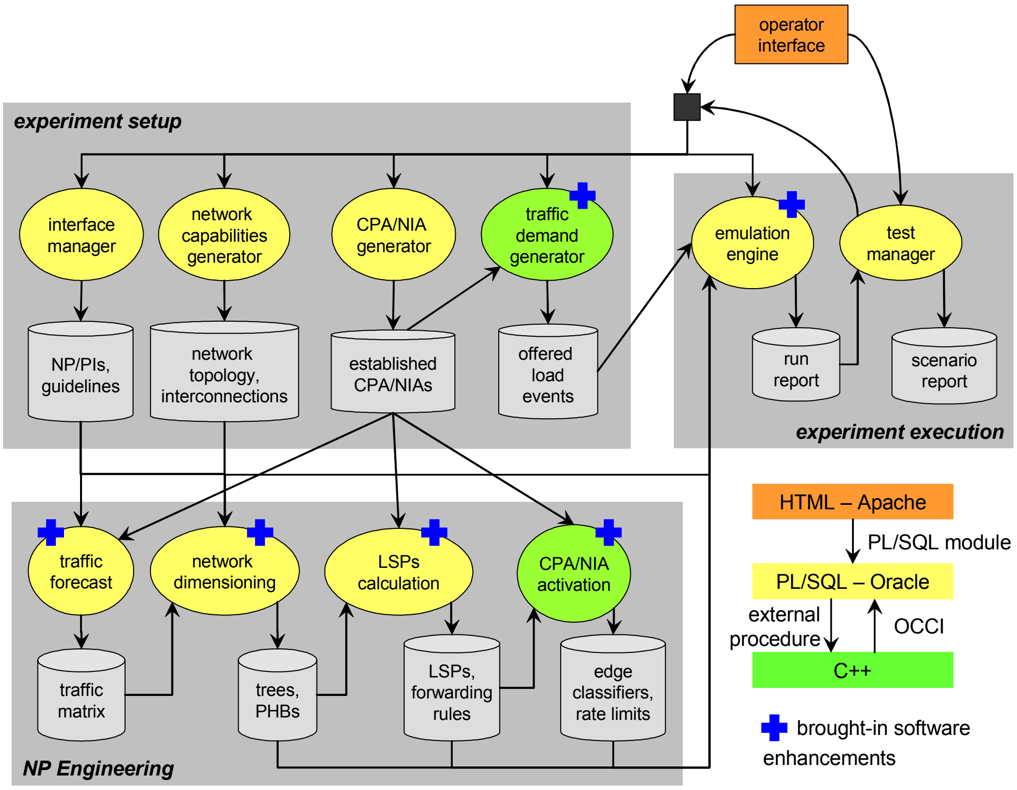


Figure 8 Overview of NP Emulation Platform.

Based on the particular test requirements in mind, the network topology and traffic generation parameters, including the population of the established CPA/NIA, are determined. The network is appropriately dimensioned so that it can cope gracefully with the anticipated demand, in accordance to the NP engineering guidelines. Traffic load events are generated in a chronological order (see next section) and the network performance is evaluated, based on the network configuration resulted from the dimensioning process, taking into account the defined NP engineering guidelines.

The network topologies that can be supported by the platform can either correspond to specific networks e.g. the GEANT network or can be randomly generated based on suitable generators found in the research community.

The traffic generation capabilities of the platform are built around the notions of CPAs/NIAs and are based on widely accepted source models and distributions. Traffic is generated at an aggregate flow level per CPA/NIA. More details are provided in the following section.

2.3.5.1 NP Emulation Platform Traffic Demand Generator Tool

The NP Emulation Platform Traffic Demand Generator Tool[2] (NPEP-TDG) (see Figure 9), integral part of the NP Emulation Platform, generates traffic load events based on a population of established CPA/NIAs, specified source profiles and parameters regarding the desired level of load to be injected in the network during an NPEP experiment execution.

The traffic load events are arranged into chronological order and present an aggregate of the traffic - over active traffic sources - to be injected in the network on behalf of a CPA/NIA from a particular access point.

[2] NPEP-TDG is based on the traffic generator function of the Traffic and Network Emulation Tool built in IST-TEQUILA project (see section 8.4.3 in TEQUILA deliverable http://www.ist-tequila.org/deliverables/D3-4a.pdf).

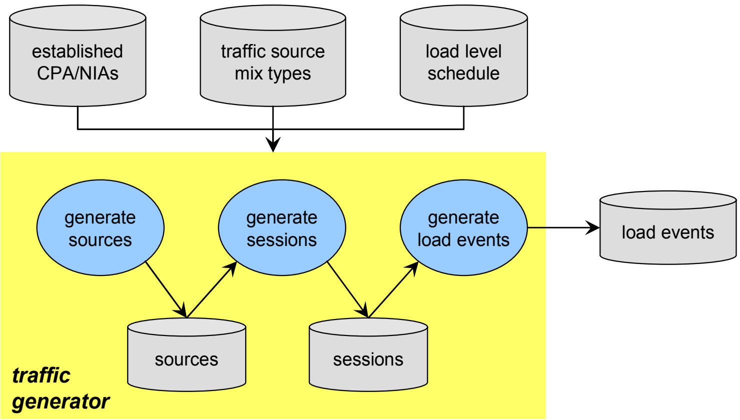


Figure 9 NP Emulation Platform Traffic Demand Generator Tool.

Specifically, NPEP-TDG takes as input:

- *Established CPA/NIAs*: the CPA/NIAs that have access to the network resources; each CPA/NIA comes with the access points, the connectivity requirements and the pools of source/destination IP addresses defined; established CPA/NIAs may be generated by the CPA/NIA generator tool or defined by the NPEP operator or a combination of the two;
- *Traffic source mix types*: specification of the types of traffic source mixes and association of each CPA/NIA with a type; a traffic source mix is composed by a population of sources with common traffic profiles, where a traffic profile (see Table 3 NPEP-TDG source traffic profile parameters) is defined in terms of session inter-arrival and holding times (distribution can be uniform, Pareto, exponential), bandwidth demand and source active/idle distribution;
- *Load level schedule parameters*: parameters that determine the load level evolution throughout an experiment. These parameters are set overall or per (type of) CPA/NIA. Load level is defined as the desired ratio of the total traffic to be generated over specific target values e.g. capacity or availability of network resources. Load levels may change in time. Based on these parameters and depending on the population of the established CPAs/NIAs, the number of traffic sources per CPA/NIA and their characteristics are determined; the more CPA/NIAs, the fewer the traffic sources to achieve a given load level.

The established CPAs/NIAs, in terms of their number and characteristics, and the load level schedule parameters are set manually or derived from a set of high-level parameters characterizing the particular traffic generation scenarios pertinent to a specific experiment.

Session	Distribution
	Inter-arrival mean time
	Holding mean time
	Variance
Load	Distribution
	Peak rate
	On mean time
	Off mean time
	Variance

Table 3 NPEP-TDG source traffic profile parameters

The generation of load events involves the following steps:

1. *Source generation*: For each CPA/NIA a number of traffic sources (customer applications) are generated following the load level schedule. The source type is determined following the settings of the associated traffic source mix, and the number of the sources is calculated so that the desired load level can be achieved. Each source is associated to an IP address from the source IP address pool associated to the CPA/NIA. The traffic source mix may change over time following the load level schedule parameters.
2. *Session generation*: For each existing source, sessions are generated following the source traffic profile (inter-arrival and holding times). Each session corresponds to an IP packet flow with source IP address the IP address assigned to its source and a destination IP address, randomly selected from the pool of destination IP address of the CPA/NIA.
3. *Load event generation*: For each generated session, load events (ON/OFF source model) are generated following the source traffic profile. Load events are aggregated over all active sources in a CPA/NIA to determine the total traffic load to be injected in the network from the particular edges that the CPA/NIA has been defined.

2.3.6 Control Parameters

Control Parameters		
Topology	*Network Size*	The number of network nodes, including ASBRs, and the connectivity degree.
	Link Capacity	The distribution of the capacity to be given to the links, including inter-domain and external links.
Traffic Generation	*CPA Types/Network Services*	The types of CPAs expected to be served by the network based on the supported network services, which are defined according to the IP network capabilities.
	Target Demand Level	The target value of the traffic demand, overall or in certain topological scopes, per supported network service/CPA type.
	Symmetry	Factors determining the symmetry in creating demand in the network. Based on the above parameters, the population of established CPAs/NIAs is produced and NPEP-DGT is configured appropriately to generate traffic.
Business Guidelines and NP Design	*NP Characteristics*	Packet transfer and availability characteristics of the NPs that can be realized based on the IP network capabilities; NPs to be created and their mapping to network services.
	Resource Distribution	The distribution of the resources allocated to NPs, overall or in certain topological scopes per (types of) CPAs.
NP Provisioning	*Demand Calculation Parameters*	Multiplexing and aggregation factors for calculating the total demand to the network over the established CPAs/NIAs.
	Route Alternatibility	Number of available routes for load-balancing or resilience purposes.

	Objective Function	The type of the objective function, e.g. minimize maximum link utilization to be used in the off-line network dimensioning process.

2.3.7 Performance Metrics

Performance Metrics		
Network Performance	*Link Utilization*	The distribution of the link load and utilization over time, per NP (QoS-class in the case of Diffserv/MPLS NP Provisioning).
	Throughput, Goodput	Traffic throughput is measured at INP edges. Goodput is the rate of the traffic delivered with the desired QoS levels.
	QoS Level	Duration of QoS deterioration.
Operations	*Configuration Load*	Complexity of configurations at nodal level, required for operating the network. It is measured based on the lines of configuration commands.

The above parameters are the ‘first-level’ network-wide measurements that can be provided by NPEP. Based on them, higher-level metrics such as, overhead traffic, processing load, could be derived e.g. through suitable models.

3 INTER-DOMAIN ROUTING EXPERIMENTS

3.1 Joint (intra and inter) Robust TE and Interactions

3.1.1 Objective

The objective of this simulation part is to study the empirical performance of the proposed algorithm that enhances the robustness of the existing NPs that use the IGP/BGP protocols for both intra- and inter- domain routing. More specifically, we will evaluate how the proposed scheme optimises intra- and inter-domain traffic engineering purposes such as minimizing the maximum link utilization in case of no failure and also any single intra- or inter-domain link failure. We refer to no failure state as Normal State (NS, i.e. no intra- or inter domain link failure) and to link failure state as Failure State (FS, i.e. single intra or single inter-domain link failure).

Note that as mentioned in D3.1, our proposal is not used to design a specific NP, but it can be "replicated" to individual NPs that apply the IGP/BGP routing protocol.

Performance comparison will be carried out between the proposed algorithm and an alternative approach proposed in [NUCC03] that only considers intra-domain robust TE ignoring inter-domain link failures. Other possible alternative approaches are: an approach with no TE optimisation (e.g. inversed bandwidth capacity) and an approach with TE optimisation without any failure consideration similar to [FORT00]. These two approaches may be implemented for completeness.

3.1.2 Environment

All the experiments will be performed on top of a standalone simulation platform. As mentioned in D3.1, in order to achieve our objective, the NP provisioning and maintenance functional block encompasses an offline joint (intra and inter) TE optimiser unit. The task of this unit is to compute a set of IGP link weights that by taking hot potato routing into account determines intra-domain paths and BGP egress point selection such that the intra and inter-domain TE objectives are optimised under both NS and all FSs. Since this problem is a multi-objective optimisation problem we place a constraint on the inter-domain TE objectives while optimising the intra-domain TE objectives. More specifically, we aim to optimise the intra-domain utilization under NS and the worst case among all FSs while not violating the worst case maximum inter-domain utilization constraint across all states.

To achieve this task, the joint offline intra- and inter- domain TE optimiser unit requires:

- The overall intra- and inter- domain network topology that contains information on intra-domain connectivity, ASBR connections and intra- and inter-domain link capacities;
- The overall traffic matrix coming from the NP mapping and NIA order handling functional blocks;
- Remote destination prefixes and their reachability information.

The outputs of this unit are:

- A robust set of egress points that determine the egress point both under NS and FSs.
- A robust set of IGP link weights that determine the intra-domain path both under NS and FSs.

We perform our simulation on Point-of-Presence (POP) level topologies randomly generated by BRITE. Moreover, according to [BROI04], inter-domains traffic volumes are top-heavy and can be approximated by Weibull distribution with shape parameter 0.2-0.3. We therefore generate the inter-domain Traffic Matrix (TM) with this distribution while setting the shape parameter to 0.3. Also according to [NUCC03] intra-domain traffic volumes of a POP topology follows the Gravity Model (GM). Following the suggestions in [BHAT01], we randomly classify 40% of POPs as "small", 40% as "medium" and 20% as "big". We therefore consider that the amount of incoming traffic at a POP is proportional to its size.

3.1.3 Control Parameters

Control Parameters		
The offline joint TE optimiser unit	*Network size*	The number of intra-domain nodes, links and number of ASBRs.
	Number of Destination Prefixes and their reachability	The number of considered destination prefixes and their reachability through each ASBR.
	Overall traffic demand	The volume of intra and inter-domain traffic demands in the traffic matrix.
	Inter-domain link utilization constraint	The maximum inter-domain link utilization that must be satisfied while optimising the intra-domain TE objectives.
	Weighting parameter	The parameter used to weight or balance conflicting objectives in multi-objective optimisation problems.
	Algorithm iteration	Algorithm iteration counter, part of the algorithm software.
	Initial link weight set	A set of IGP link weight used for algorithm initialisation.
	MAX_LINK_WEIGHT	The maximum IGP link weight to be set.

3.1.4 Performance Metrics

Performance Metrics		
The offline joint TE optimiser unit	*Maximum intra-domain utilization under NS*	The highest utilization among all intra-domain links under no failure.
	Maximum inter-domain utilization under NS	The highest utilization among all inter-domain links under no failure.
	Worst case maximum intra-domain utilization under all FSs	The highest among all maximum intra-domain links utilizations under *all FSs* consisting of intra and inter-domain single link failure.
	Worst case maximum inter-domain utilization under all FSs	The highest among all maximum inter-domain links utilizations under *all FSs* consisting of intra and inter-domain single link failure.

3.2 BGP planned maintenance

3.2.1 Introduction

3.2.1.1 Objective

The objective of this simulation part is to evaluate the performance of the proposed algorithm "BGP Planned Maintenance (BGP-PM)" and assess the convergence time of the BGP protocol in specific cases of a planned maintenance operation where this PM has an impact on the BGP protocol such as an ASBR shutdown or an eBGP session shutdown to another AS.

For main typical iBGP and eBGP topologies the objective is to measure the Loss of Connectivity (LoC) currently perceived by the customers and to measure the enhancement brought by a planned maintenance strategy.

From AGAVE perspective, the goal is to evaluate the impact on high availability-related parameters when a Network Plane uses BGP-PM.

3.2.1.2 Overview of simulation methodology

One goal of this simulation is to have results as close as the behaviour experienced by customers of a real network using exiting router technologies. As performances tests are very hardware and software specific, we will use the real hardware and software of a major router vendor currently used by service providers. To reduce the CAPEX and OPEX of this experiment, we will use virtual routers to emulate two networks topologies (ASs). In fact, we will run on one hardware router multiple times the routing process(es) to emulate the routing behaviour of a network composed of multiples routers. This has also the advantage of providing perfect time synchronization between the routers of the network, which is important to better understand the order and the timing of the BGP messages and events required during the convergence time.

The network topology emulated will be specific to this experiment (i.e. not shared with other AGAVE experiments) as on the one hand the virtual router technology has some limitations regarding performance and can only simulate a small number of routers (typically fifteen) and on the other hand, we want to simulate multiple BGP topologies using the same network topology at the IP layer. Given these two constraints, the network topology will be chosen and optimised for these tests. For example, the topology of a customer network dually connected to an INP using a hierarchical route reflector topology would be:

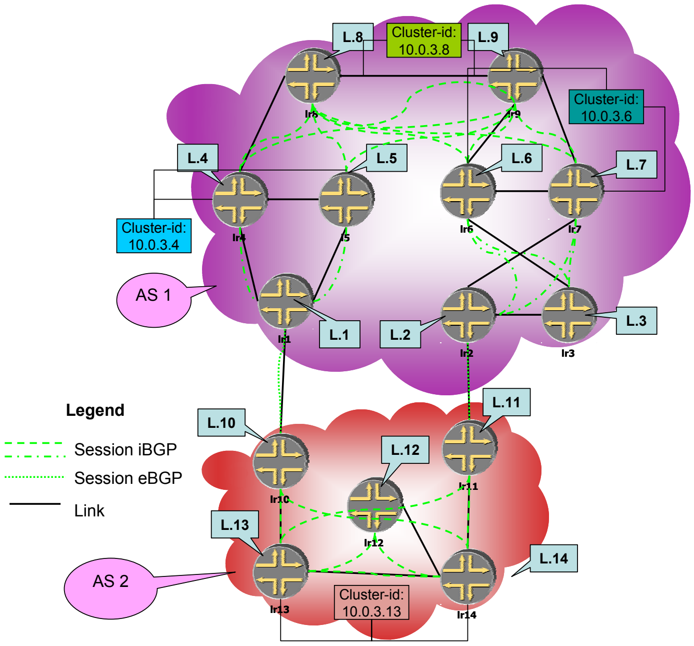


Figure 10 Example of INP using hierarchical route reflector topology.

At the BGP layer, the simulation will be performed using different BGP topologies: iBGP full mesh, iBGP RR, hierarchical RR, eBGP route selection based on local_pref and IGP cost. The simulation will also use different forwarding paradigm: IP (pervasive BGP), MPLS (BGP free core), BGP/MPLS VPNs. The loss of connectivity experienced by the customers will be measured for all theses topologies with and without the BGP planned maintenance enhancement; the goal being to investigate the gain of this enhancement.

These simulations do not need any traffic matrix; as for time accuracy only a limited set of flows will be used (note that, according to the Shannon theorem, the timing accuracy is linearly dependant of the polling frequency hence the frequency of sending IP packets between customers sites). The simulations do need to use BGP routing tables to load the BGP control plane. In order to reflect a real case, BGP routing tables from a real Internet network will be injected.

3.2.2 Terminology and definitions

This section provides a list of terms definitions as used within this document:

- Convergence: The point at which every router on the network has received and processed all routing information from its peer routers.
- Logical Router: The Juniper M7 router can be partitioned into several independent logical routers. Each of theses routers is a single entity with its own routing tables, its own interfaces and its own configuration set up.
- Traffic Interruption time: It is the time during which packets from or to the customer are lost because of the convergence of the network after a break.

3.2.3 Test Plan

3.2.3.1 Organization

The structure is hierarchical; the suite is composed of tests groups that are composed of elementary tests subgroups (or elementary tests). The structure of tests groups depends on the continuation itself. An elementary test has a unique reference: REF / Group reference (/ Subgroup reference)* / Elementary test reference.

3.2.3.2 Definitions of selection conditions

[ISO-9646-2] defines following values with respect to the variable "status":

Status	**Meaning**
C	Conditional
C*N*	N is an integer, for mutually exclusive or selectable conditions from a set
M	Mandatory
O	Optional
X	Prohibited
N/A or -	Not applicable

3.2.3.3 Structure by subgroups of elementary test

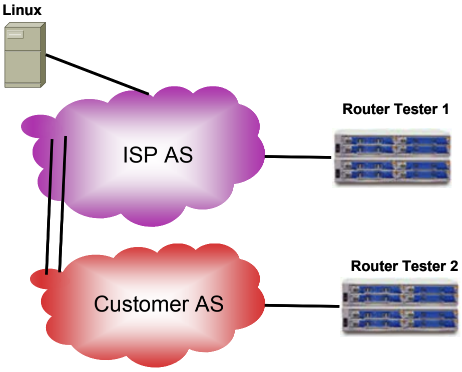


Figure 11 Elementary test topology.

Each elementary test of traffic interruption adopts the following method:

- The architecture described in the hierarchy is loaded in the Juniper M7. A capture of the configuration of the Juniper router is done, together with a capture of the content of the BGP routing table and the state of each BGP session of each logical router. Of course this will be done without the charge of routes added by the Linux host to improve the readability.
- The Linux host advertises the selected amount of routes in the ISP network to simulate a realistic network. For this test suite, 12000 routes are advertised.
- Traffic is injected in the network in both directions (5 streams from RT1 to RT2 and 5 streams from RT2 to RT1). This traffic flows using the link that will go up and down during the planned maintenance operation.
- One of the eBGP links between customer AS and ISP AS is shutdown.
- The traffic (packet) loss is monitored by the RT in both directions. The traffic interruption time can be calculated using the formula:

$$Traffic\_Interruption\_Time = \frac{Nb\_of\_packets\_lost * size\_of\_1\_packet}{Transmitted\_Throughput}$$

Caution: this formula works if the transmitted (injected) throughput is constant and if most packets are lost during the shutdown or the restart of the link and not when the network is converged due to various misconfiguration statements for instance. This will ensure that the traffic is lost only because of the shutdown or restart of the BGP link. This hypothesis can be easily checked looking at the reported graphs (packets losses and received packets per second) of the router tester.

An additional constraint must be added: the TTL of the packets must be low enough to forbid the reception of looped packets in the receiving ports of the Router Tester.

- During this process, update messages sent and received on all routers are captured.
- At the end, the final configuration, BGP routing table and BGP sessions status of each logical router, is saved after the routes advertised by the Linux host are withdrawn.
- These operations are performed again when the shutdown link is started again.

- The measure of the traffic interruption in case of a shutdown and a restart of the link is performed five times to produce a pool of measured times. The capture of the configuration, the content of the routing tables and status of the BGP session is only necessary once.

The elementary test for each subgroup defined further on will be referred to as:

- Down X-Y_Nb for the shutdown of the BGP session between the logical routers lrX and lrY on the logical router lrX. Nb designing the test number.
- Up X-Y_Nb for the restart of the BGP session between the logical routers lrX and lrY on logical router lrX. Nb designing the test number.

This method will be applied for the architectures described in the following hierarchy. The first level of the hierarchy is:

- First, the current behaviour without any tuned set up of BGP.
- Then, the behaviour with a manual planned maintenance strategy performed. That is to say, the operator will have to perform manually changes in the policies to induce a planned maintenance behaviour.

Ref. of subgroup	**Condition of selection**	**Object**
REF / CURRENT	M	Tests of traffic interruption time with actual BGP behaviour.
REF / MANUAL_PM	M	Tests of traffic interruption time with Planned Maintenance manually performed by the operator through the use of route-map.

3.2.3.3.1 Elementary tests of subgroup REF / CURRENT

This subgroup is split into three subgroups. In each subgroup, the type of path used to transport the traffic is changed:

- In the first subgroup, the traffic is transported normally using IP.
- In the second subgroup, MPLS paths are used in one of the AS.
- In the third subgroup, L3VPN tunnels are used in one of the AS.

Ref. of subgroup	**Condition of selection**	**Object**
REF / CURRENT / IP	M	Tests the architectures when IP forwarding is used with pervasive iBGP within the AS.
REF / CURRENT / MPLS	M	Tests the architectures when MPLS forwarding is used within the AS.
REF / CURRENT / VPN	M	Tests the architectures with BGP/MPLS VPNs.

3.2.3.3.2 Elementary tests of subgroup REF / CURRENT / IP

In this subgroup, the impacts of the different BGP topologies are evaluated. First several eBGP topologies will be evaluated then several iBGP topologies.

Ref. of subgroup	Condition of selection	Object
REF / CURRENT / IP/ eBGP	M	Test of the current behaviour with several eBGP topologies.
REF / CURRENT / IP/ iBGP	M	Test of the current behaviour with several iBGP topologies.

3.2.3.3.3 Elementary tests of subgroup REF / CURRENT / IP/ eBGP

This subgroup defines two different customers <> ISP architectures:

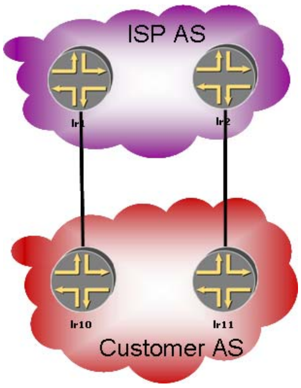


Figure 12 A single homed dual attached customer with two separated paths (2CE-2PE)

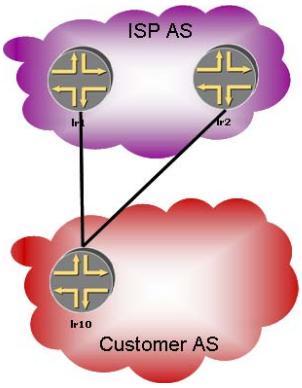


Figure 13 A single homed dual attached customer with a single router connecting to the ISP but redundant links (2PE-1CE)

The physical and ISIS architecture used is the following:

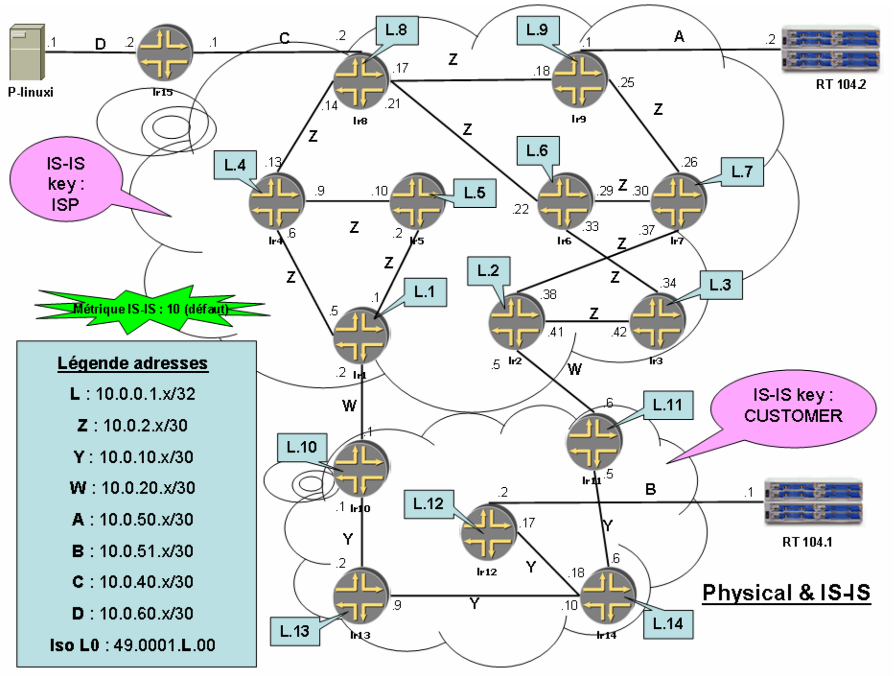


Figure 14 First eBGP architecture.

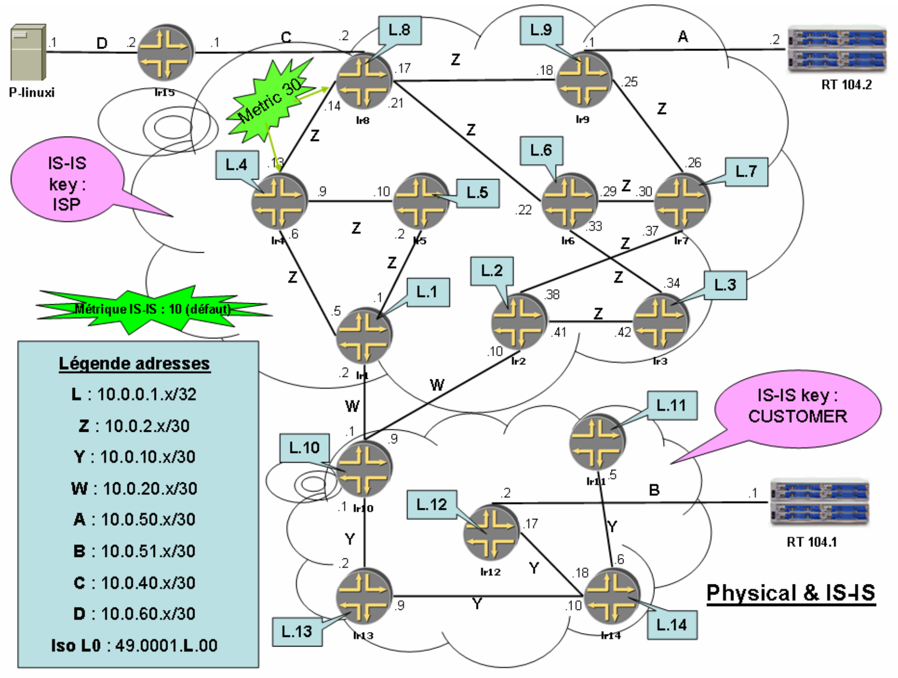


Figure 15 Second eBGP architecture.

The ISP network is in the upper part (logical routers lr1 to lr9) and the Customer network in the lower cloud (logical routers lr10 to lr14).

The equipment RT 104.x is the test equipment: an Agilent Router Tester (RT).

Finally, the logical router (lr15) and the Linux host (P-linuxi) in the upper left are used to inject routes in the ISP network.

The default ISIS metric is to be used unless specified otherwise. Wide-metrics are used as well as the simple ISIS authentication type with the keys indicated in the figure above.

BGP updates messages sent and received are logged in each logical router with a timestamp having a precision of one microsecond.

In the tests, the connection between P-linuxi and lr15 is done using an Ethernet 100Mbps RJ45 connection.

The connection between lr12 <> RT 104.1 and lr9 <> RT 104.2 is done using Gigabit Ethernet optical physical links.

The host P-linuxi will advertise in BGP a specified amount of routes extracted from real data of the RBCI. These routes will be advertised in the ISP network using the logical router lr15.

The iBGP topology chosen for the customer AS is a single level of Route Reflector with the same Cluster-ID. For the ISP network, the topology is a hierarchical Route Reflector topology with the same Cluster-ID for each pair of Route Reflectors.

Ref. of test	Condition of selection	Priority	Object
REF / CURRENT / IP / eBGP / 2PE-2CE	M	Normal	The customer uses two separated paths on two PE and two CE.
REF / CURRENT / IP / eBGP / 2PE-1CE	M	Normal	The customer uses two separated paths on two PE connected to a single CE.

3.2.3.3.4 Elementary tests of subgroup REF / CURRENT / IP / iBGP

Finally, several iBGP architectures will be tested in this configuration:

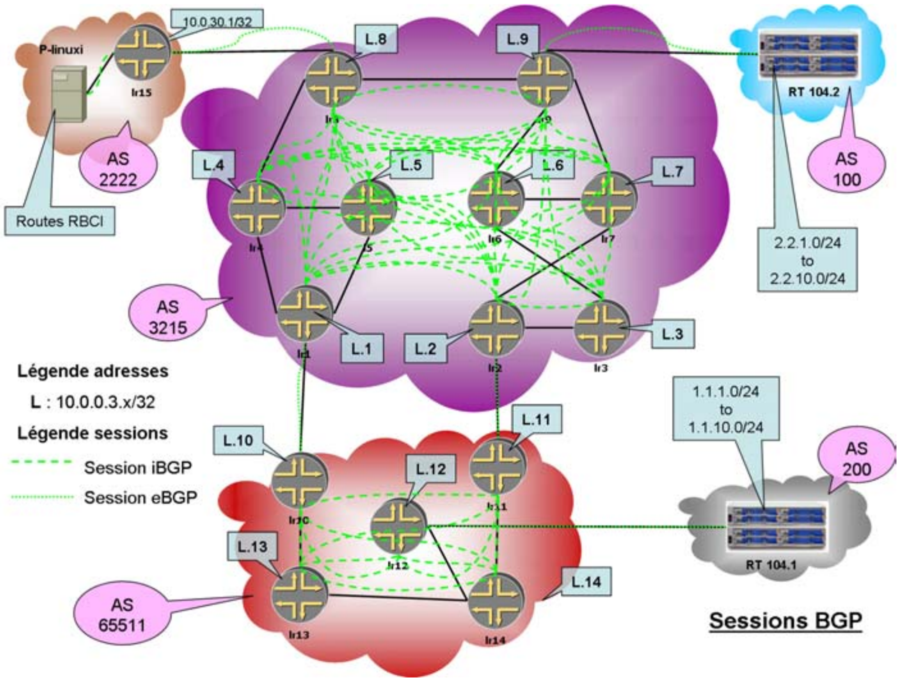


Figure 16 Full mesh topology.

In the full mesh topology the routers in each AS are fully meshed with the others using iBGP sessions.

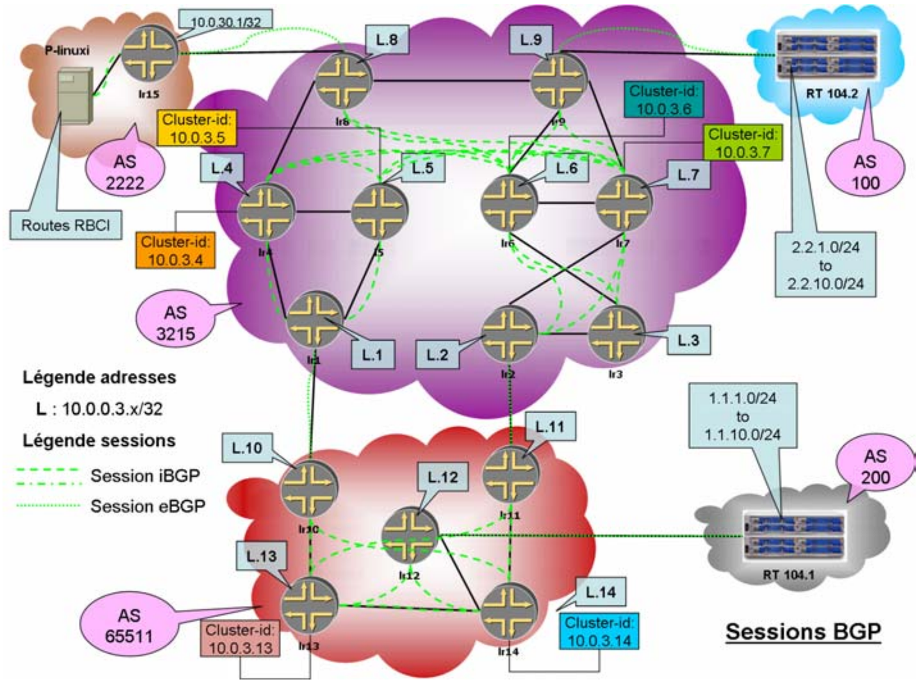


Figure 17 Route Reflector topology.

In this topology, lr4, lr5, lr6 and lr7 are RR fully meshed together. Each one has its own Cluster-Id. The Customer network has two RR: lr13 and lr14.

- Hierarchical Route Reflector topology with different Cluster-Id for each RR.

In this topology:

- lr4 and lr5 are redundant RR for lr1,
- lr6 and lr7 are redundant RR for lr2 and lr3,
- lr8 and lr9 are hierarchical redundant RR for lr4, lr5, lr6 and lr7.

Each RR has its own Cluster-Id.

The Customer BGP configuration is the same as the previous configuration since there are not enough routers to implement a hierarchical topology.

The link between lr8 and lr4 has a different ISIS metric (30 instead of default 10). This will impact the selection of BGP next hop for customer AS for router lr8. The logical router lr8 will select lr2 as a next-hop instead of lr1. This will result in the masking of the route via lr1 to router lr6 and lr7.

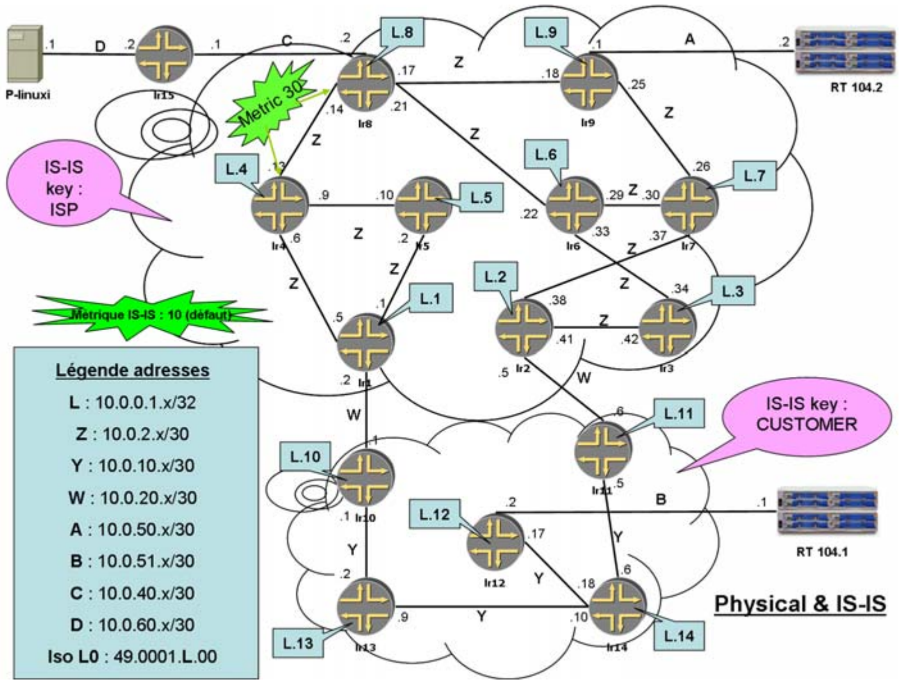


Figure 18 Topology with different metric between lr8 and lr4.

The BGP topology is the following one:

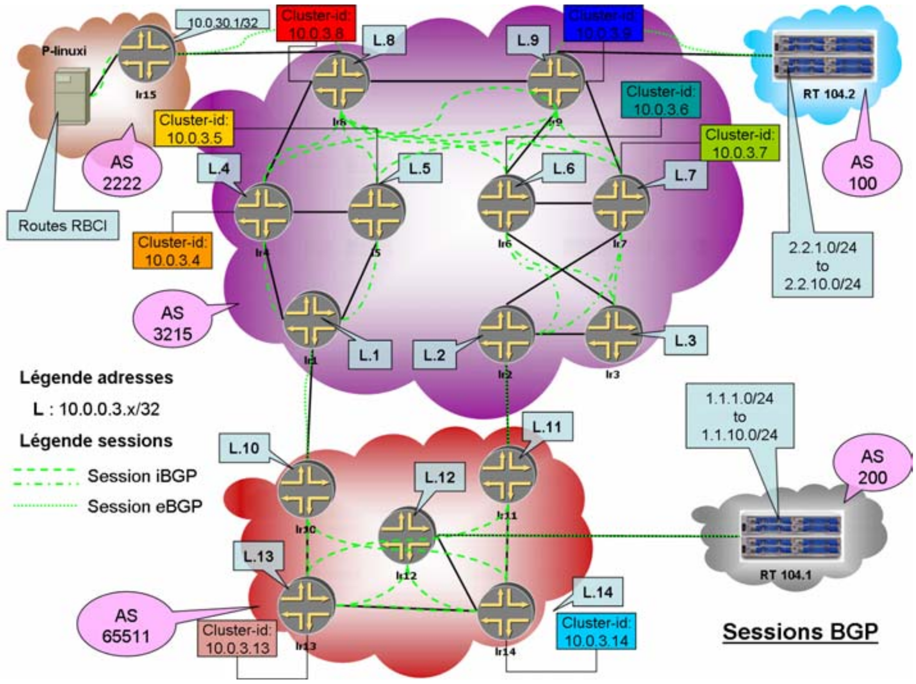


Figure 19 BGP topology with different Cluster-IDs.

- Hierarchical Route Reflector topology with the same Cluster-Id for each pair of redundant Cluster-Id.

This topology is the same as the previous one with the ISIS weighted link, but this time the cluster-Id of each pair of redundant RR is the same:

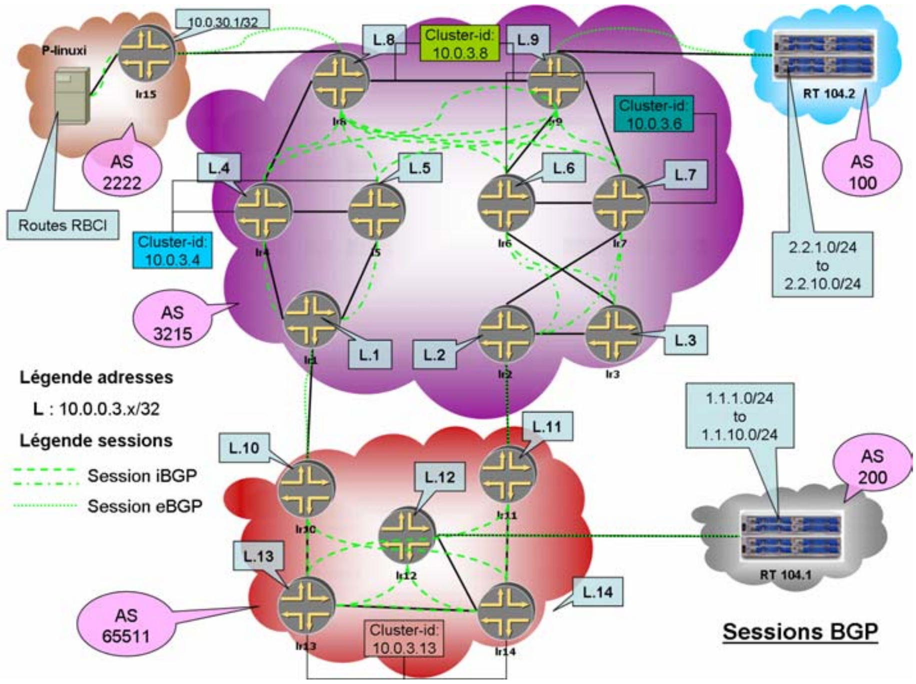


Figure 20 BGP topology with same Cluster-ID for each pair of redundant Cluster-IDs.

Then all the previous iBGP architectures are done again but with a local preference attribute set on logical routers lr1 and lr2. These LOCAL_PREF are set to force the route advertised by logical router 2 to be preferred.

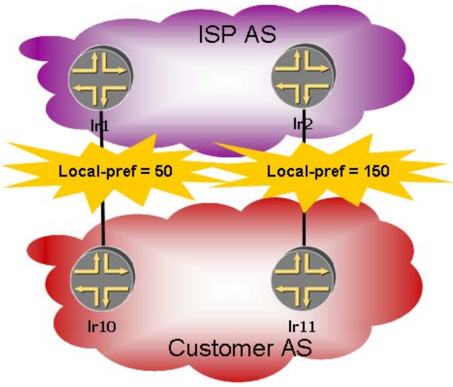


Figure 21 Topology with different local-pref attributes.

In the ISP AS, the routes advertised by logical router lr1 will have a local preference of 50 and the routes advertised by logical router lr2 will have a local preference of 150. This will force all traffic to Customer AS to go through lr2.

An additional test is performed with the topology REF / CURRENT / IP / iBGP IGP / HRR_same_Cluster_Id but instead of shutting down the eBGP session, the entire BGP process will be shutdown.

Ref. of test	Condition of selection	Priority	Object
REF / CURRENT / IP / iBGP IGP / Full-mesh	M	Normal	The iBGP topology used in the customer and in the ISP AS is a full mesh topology. The route selection process is done using the IGP selection.
REF / CURRENT / IP / iBGP IGP / RR	M	Normal	The iBGP topology used in the customer and in the ISP AS is a redundant Route Reflector topology (lr4-5-6-7 & lr13-14). Each Route Reflector has its own Cluster-ID. The route selection process is done using the IGP selection.
REF / CURRENT / IP / iBGP IGP / HRR_diff_Cluster_Id	M	Normal	The iBGP topology used in the customer AS is a redundant Route Reflector topology (lr13-14). In the ISP AS, a hierarchical redundant RR topology is used with lr4-5 RR for lr1, lr6-7 RR for lr2-3 & lr8-9 HRR for lr4-5-6-7. In addition the ISIS link between lr8 and lr4 is weighted (30 instead of 10). Each Route Reflector has its own Cluster-ID. The route selection process is done using the IGP selection.
REF / CURRENT / IP / iBGP IGP / HRR_same_Cluster_Id	M	Normal	The iBGP topology used in the customer AS is a redundant Route Reflector topology (lr13-14). In the ISP AS, a hierarchical redundant RR topology is used with lr4-5 RR for lr1, lr6-7 RR for lr2-3 & lr8-9 HRR for lr4-5-6-7. In addition the ISIS link between lr8 and lr4 is weighted (30 instead of 10) and each pair of redundant RR or HRR has a single Cluster-Id. The route selection process is done using the IGP selection.
REF / CURRENT / IP / iBGP LP / Full-mesh	M	Normal	The iBGP topology used in the customer and in the ISP AS is a full mesh topology. The route selection process is influenced by LOCAL_PREF set up (50 lr1, 150 lr2).
REF / CURRENT / IP / iBGP LP / RR	M	Normal	The iBGP topology used in the customer and in the ISP AS is a redundant Route Reflector topology (lr4-5-6-7 & lr13-14). Each Route Reflector has its own Cluster-ID. The route selection process is influenced by LOCAL_PREF set up (50 lr1, 150 lr2).
REF / CURRENT / IP / iBGP LP / HRR_diff_Cluster_Id	M	Normal	The iBGP topology used in the customer AS is a redundant Route Reflector topology (lr13-14). In the ISP AS, a hierarchical redundant RR topology is used with lr4-5 RR for lr1, lr6-7 RR for lr2-3 & lr8-9 HRR for lr4-5-6-7. In addition the ISIS link between lr8 and lr4 is weighted (30 instead of 10). Each Route Reflector has its own Cluster-ID. The route selection process is influenced by LOCAL_PREF set up (50 lr1, 150 lr2).
REF / CURRENT / IP / iBGP LP / HRR_same_Cluster_Id	M	Normal	The iBGP topology used in the customer AS is a redundant Route Reflector topology (lr13-14). In the ISP AS, a hierarchical redundant RR topology is used with lr4-5 RR for lr1, lr6-7 RR for lr2-3 & lr8-9 HRR for lr4-5-6-7. In addition the ISIS link between lr8 and lr4 is weighted (30 instead of 10) and each pair of redundant RR or HRR as a single Cluster-Id. The route selection process is influenced by LOCAL_PREF set up (50 lr1, 150 lr2).
REF / CURRENT / IP / iBGP IGP / HRR_same_Cluster_Id_down_BGP	M	Normal	Same as REF / CURRENT / IP / iBGP IGP / HRR_same_Cluster_Id but instead of shutting down the eBGP session, the BGP protocol is disabled.

3.2.3.3.5 Elementary tests of subgroup REF / CURRENT / MPLS

The tests of the subgroup REF / CURRENT / MPLS are the same than the tests of the subgroup REF / CURRENT / IP but in addition of the configuration set up explained above, MPLS is enabled in the ISP AS and in the customer AS on each router with the LDP protocol. MPLS and LDP must be set up on each internal interface of the ASs.

3.2.3.3.6 Elementary tests of subgroup REF / CURRENT / VPN

Some of the tests done within the subgroup REF / CURRENT / IP will be done with a L3VPN architecture implemented in the ISP network. These include:

- All the tests for the eBGP topology:

The iBGP topology chosen in this case is a single level of Route Reflector for the client AS and the ISP AS.

Ref. of test	Condition of selection	Priority	Object
REF / CURRENT / VPN / eBGP / 2PE-2CE	M	Normal	The customer uses two separated paths on two PE and two CE.
REF / CURRENT / VPN / eBGP / 2PE-1CE	M	Normal	The customer uses two separated paths on two PE connected to a single CE.

- Some of the tests for the iBGP topology.

All the tests do not need to be performed because several architectures are not really relevant in this context. The tests will be done for the RR topology, with and without the local preference attribute, with a single route distinguisher for both paths (via lr1 and via lr2) and with a route distinguisher for each path. The VPN is set between lr8, lr9, lr1 and lr2. All their eBGP sessions are inserted into the VRF.

Finally, a test will be performed with Inter-AS VPN option B between the two AS.

Ref. of test	Condition of selection	Priority	Object
REF / CURRENT / VPN / iBGP IGP 1RD / RR	M	Normal	The iBGP topology used in the customer and in the ISP AS is a redundant Route Reflector topology (lr4-5-6-7 & lr13-14). Each Route Reflector has its own Cluster-ID. The route selection process is done using the IGP selection. A common Route Distinguisher is used for the paths to the customer AS.
REF / CURRENT / VPN / iBGP LP 1RD / RR	M	Normal	The iBGP topology used in the customer and in the ISP AS is a redundant Route Reflector topology (lr4-5-6-7 & lr13-14). Each Route Reflector has its own Cluster-ID. The route selection process is influenced by LOCAL_PREF set up (50 l1, 150 lr2). A common Route Distinguisher is used for the paths to the customer AS.
REF / CURRENT / VPN / iBGP IGP 2RD/ RR	M	Normal	The iBGP topology used in the customer and in the ISP AS is a redundant Route Reflector topology (lr4-5-6-7 & lr13-14). Each Route Reflector has its own Cluster-ID. The route selection process is done using the IGP selection. Each path to the customer AS has its own Route Distinguisher.

REF / CURRENT / VPN / iBGP LP 2RD/ RR	M	Normal	The iBGP topology used in the customer and in the ISP AS is a redundant Route Reflector topology (lr4-5-6-7 & lr13-14). Each Route Reflector has its own Cluster-ID. The route selection process is influenced by LOCAL_PREF set up (50 l1, 150 lr2). Each path to the customer AS has its own Route Distinguisher.
REF / CURRENT / VPN / Inter-AS option B / IGP	Optional	Low	The Inter-AS VPNv4 exchange is implemented between the ISP AS and the client AS. The IGP metric is used to select the paths. The VPN is set between lr8, lr9 and lr12.
REF / CURRENT / VPN / Inter-AS option B / LP	Optional	Low	The Inter-AS VPNv4 exchange is implemented between the ISP AS and the client AS. A local-pref attribute is used on lr1 and lr2 like above.

3.2.3.3.7 Elementary tests of subgroup REF / MANUAL

The tests done in the subgroup REF / CURRENT will be done again but a planned maintenance manual operation will be simulated using communities and policies in logical router lr2 and lr11 to induce a Planned Maintenance behaviour between logical router lr11 and logical router lr2.

3.2.4 Testbed

The tests campaign will be performed on a single Juniper M7 using the logical routers feature to simulate a complex customer <> ISP network within a single router. This will induce an easier logging capability of the updates messages exchanged and a perfect time synchronization & cohesion between the logical routers. These features are essential to correctly understand the dynamic of the network tested.

Actual Internet Routes will be simulated and injected in the network to reproduce a realistic situation. The number of routes injected has to be chosen with caution to avoid any overload of the router, which could modify its comportment.

3.2.4.1 Overview of equipments used

Name	Purpose	Hardware	Software	Additional Information
JM7B	Simulation of the architecture under test.	Juniper M7i	JUNOS 7.1B2.2	RE-5.0, M7i midplane REV4, 2x G/E, 1000 BASE SFP-SX & 4x F/E, 100 BASE-TX
AgtN2X1	Router Tester, flow generation & measurement of interruption time.	Agilent Router Tester N2X	N2X version 6.5, Router Tester 900 6.5, build 4.10B	2x G/E, 1000 BASE SFP-SX
P-LinuxI	BGP route simulation.	PC	Red Hat Linux 3.2.3-47, Linux version 2.4.21-27.EL	

3.3 IP Tunnelling

The validation and evaluation campaign planned for the IP Tunnelling solution is made up of two different parts. The first part will be based on the implementation of prototype TSCs and their deployment in a testbed. This part mainly focuses on implementability assessment and is described in Section 3.3.1. The second part will be based on simulations and will focus on stability and scalability evaluation. Simulation-based experiments are described in Section 3.3.2.

3.3.1 Testbed-based Experiment

3.3.1.1 Objectives

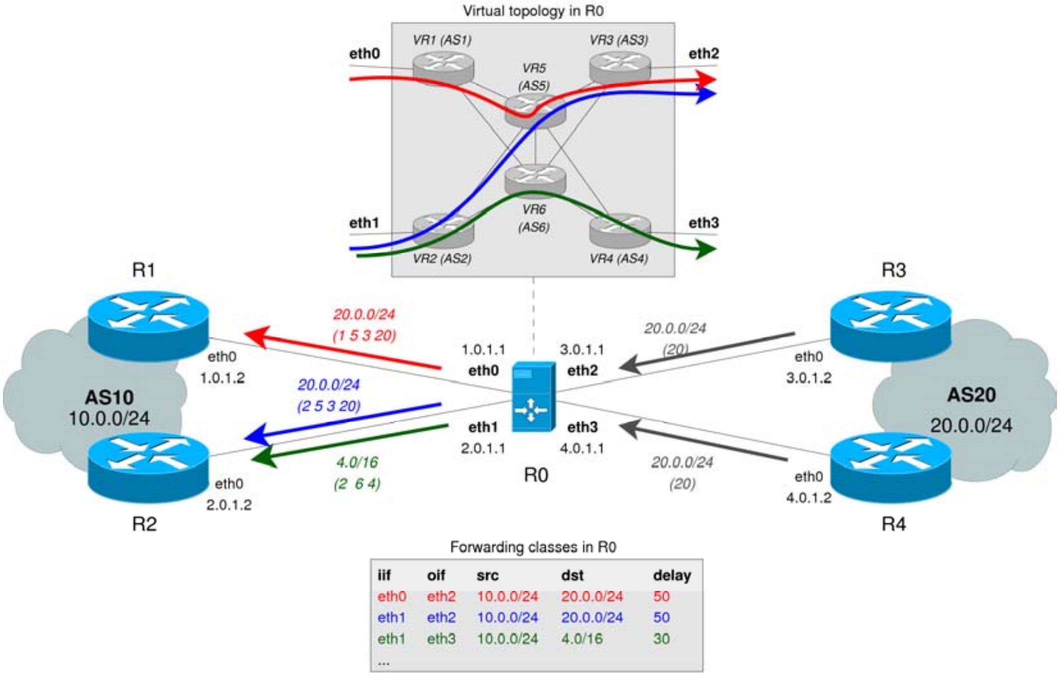
The objective of this experiment is to validate the initial specification of the IP Tunnelling framework described in D3.1. Specifically, we will evaluate how the proposed framework is able to achieve end-to-end performance enhancements and interdomain Traffic Engineering objectives. The D3.1 specification introduces the notion of a *Tunnelling Service Controller* (TSC), a software process running in the control plane of a router or separate workstation. The purposes of the TSC are to automate the discovery and selection of alternative interdomain paths, the establishment of IP tunnels and a continuous monitoring of the paths performances. Each piece of functionality was specified in D3.1 as a distinct system component. In addition, D3.1 specified the parameters, protocols and algorithms envisioned for implementing each component.

3.3.1.2 Overview of the methodology

The first step of this experiment consists in validating the initial specification and assessing the feasibility of the solution through the construction of prototype TSCs and their deployment in a small testbed. The TSCs will be implemented as a kernel module in FreeBSD OS. In particular, this solution will be aligned to the recent IETF/IRTF proposal called LISP (Locator ID Separation Protocol) [LISP00]. LISP allows to easily set-up tunnels (or even nested tunnels) between border routers and does not introduce constraints on the tunnel end-points selection criteria. We will validate the following aspects: (1) the mechanism used by LISP to advertise and discover the ingresses and parameters of a remote INP; (2) the construction of the list of candidate end-to-end paths; (3) the measurement of the performances of each candidate path; (4) the selection of the end-to-end paths that best fulfil the individual flow constraints and global network objectives; (5) the establishment in the network of the state necessary for directing the traffic flows in their assigned paths.

To perform this initial validation we rely on the following set-up. We need a testbed topology that allows us to emulate two different INPs. In a first time, we will emulate single-router INPs. In a second time, we could rely on software virtualisation [FERN04] for emulating more complex INP networks. The Network Planes in each INP would be implemented by using policy routing and traffic control since it is available under the Linux platform. Each emulated INP need to be at least dual-homed in order to offer realistic path diversity. For this purpose, each border router in the emulated INP will be connected to two different ISPs. Each ISP can be emulated with a single router since the topology behind the peering links is seen as a cloud by the TSCs.

In order to get realistic BGP routes in the lab, we rely on virtual topologies that are emulated on a single computer. We plan to use the AS-level topology inferred by Subramanian et al [SUBR02] from BGP routes collected on real routers. This topology contains one node per AS and the business relationships between each pair of ASs. We also plan to use a topology based on the GEANT (See Section 5) and Abilene topologies to emulate a research and academic environment. This second dataset, though it contains only two intermediate ASs, has the advantage of providing a real router-level topology. We use the C-BGP routing solver [QUOI05] to compute how the BGP routes between both INPs are propagated over the virtual topology. The routes are originated by the border routers of each stub INP and by their providers. Once the routes have been propagated by C-BGP, the resulting routes are injected in the border routers of each INP. The BGP routes exchanges that occur between the Internet emulator and the real border routers are done by relying on the SBGP tool [MRT].



iif	oif	src	dst	delay
eth0	eth2	10.0.0/24	20.0.0/24	50
eth1	eth2	10.0.0/24	20.0.0/24	50
eth1	eth3	10.0.0/24	4.0/16	30
...				

Figure 22 - Network emulation methodology.

This methodology is illustrated in Figure 22. In this example, the two dual-homed stub INPs under test are *AS10* and *AS20*. The first INP, *AS10*, is connected to the Internet through its border routers *R1* and *R2*, attached to *AS1 (VR1)* and *AS2 (VR2)* respectively. Similarly, *AS20* is connected to the Internet through its border routers *R3* and *R4*, attached to *AS3 (VR3)* and *AS4 (VR4)* respectively. *VR1*, *VR2*, *VR3* and *VR4* are virtual routers that are emulated by the software router *R0*.

In order to evaluate the IP tunnelling proposal, we also need to propagate the prefixes of the ASs providing their Internet connectivity to the INPs under test. The routes towards these ISPs are originated from inside the virtual topology. We show on Figure 22 an example of such route. This is a route originated by *VR4* for the prefix *4.0/16* owned by *AS4*. After the convergence, this route will be advertised to *R1* and *R2*. We only show the route learned by *R2*, which has AS-Path *(2 6 4)*.

In addition to the computation and advertisement of BGP routes propagated over the virtual topology, R0 can also emulate the forwarding properties of the paths. This emulation can be performed by relying on Linux policy routing and traffic control capabilities or by using NetPath [AGAR05]. NetPath is a wide area network properties emulator relying on Click [KOHL00], a modular router that can run on commodity hardware. NetPath is able to emulate variable delay, loss, duplication and re-ordering. It currently lacks IP routes computation that can be provided by C-BGP. Basically, the single path emulation performed by NetPath relies on Click elements, as described in [AGAR05]. This works as follows: read packets from a network interface, delay them for a fixed interval and send them to an egress network interface after making sure they are headed to their appropriate next-hop. In the current NetPath architecture, the packets are delayed before any forwarding decision is taken. We illustrate in Figure 23 how the NetPath Click elements are modified in order to forward packets based on the paths computed by C-BGP and then to apply the delay corresponding to the forwarding path. We show in Figure 22 an example of the forwarding classes that would be obtained from the C-BGP computation and enforced by the Click elements.

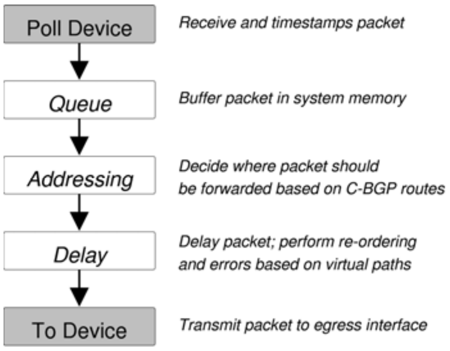

Poll Device
Receive and timestamps packet
Queue
Buffer packet in system memory
Addressing
Decide where packet should be forwarded based on C-BGP routes
Delay
Delay packet; perform re-ordering and errors based on virtual paths
To Device
Transmit packet to egress interface

Figure 23 - Click elements used in NetPath and modified to make use of C-BGP routes.

Using this methodology, we can perform a large number of experiments. First, we can evaluate how the system would react to changes in routing. By changing the virtual topology, and re-computing the BGP routes, BGP updates will be sent to the INP border routers, changing the available interdomain paths. Second, we can evaluate how the system would react to path performance changes (degradation/improvement). The path performance could be due to routing changes or by link performance changes. An example of performance degradation due to path change could be to switch from path A→B→C to A→D→C where the second path goes through higher latency links but the BGP quality of the route (AS-Path length) is unchanged. Third, we can evaluate the behaviour of the system when facing performance oscillation. We can simulate this behaviour by continuously degrading and improving the performance of a path. Finally, we can evaluate the reaction of the system to the inability for remote INP to set-up the selected remote path. This can be done by advertising good performances during the network capabilities discovery step, but by refusing the requests during the provisioning phase.

3.3.1.3 Control Parameters

Control Parameters		
Parameters controlled in the emulated topology	*Network size*	The number of emulated ASs.
	Link Capacity	Bandwidth capacity of emulated links.
	Link delay	The delay introduced by emulated links.

3.3.1.4 Performance Metrics

Performance Metrics		
	End-to-End Throughput	Improvements in terms of end-to-end throughput due to the selection of the path with the highest capacity.
	End-to-End Delay	Improvements in terms of end-to-end delay due to the selection of the path with the lowest delay.

3.3.2 Simulation-based experiment

3.3.2.1 Objectives

The objective of the simulation-based experiments is to evaluate the performance properties of the IP tunnelling proposal described in D3.1. Specifically, we will evaluate the convergence, stability and scalability properties. By convergence property, we mean the ability to update the paths selection in case of condition changes such as traffic volume increase, path performance degradation or access link load increase. By the stability property, we understand the ability of the algorithm to avoid oscillations that could show up due to the interaction between the paths selection and the paths performance measurement. Finally, by the scalability property, we mean the ability for the developed solution to work with a large number of participants and flow constraints. In particular, the number of IP tunnels to establish in order to reach a given optimisation objective must be low.

3.3.2.2 Overview of the methodology

We will rely on the C-BGP simulator [QUOI05] for performing our large-scale simulation experiments. We will use an Internet topology such as the one inferred by Subramanian et al. [SUBR02] with one router per AS and realistic routing policies. We will also rely on topologies generated by GHITLE since it allows to control the shape (width and depth) of the topology. Using C-BGP, we will mainly rely on static simulations performed as follows. We first change the routes by cutting a link in the topology, or by changing the router policies and we compute the new BGP routing outcome using C-BGP. In a second step, we run the TSC paths selection algorithm, which leads to a set of forwarding paths. Finally, we measure the impact of the TSC decisions. These steps are run multiple times in a loop. Each iteration of the loop simulates a unit of time of the simulation. This is a realistic approach since the operation of one TSC will typically be to collect the available paths and measure their performance at each time interval.

Using this methodology, we plan to perform the following experiments. First, count the number of IP tunnels to establish in order to reach various optimisation objectives. The number of IP tunnels established is a key parameter of the solution scalability. Two main objectives will be considered: reducing the latency of the end-to-end paths used for reaching a subset of networks and balancing the traffic load over the access link. The combination of both objectives could also be evaluated. It is interesting not only to count the number of IP tunnels required by a single TSC, but also the number of IP tunnels established by all the TSCs if we assume that all the Internet stubs are using the IP tunnelling proposal.

Second, we will evaluate the frequency of switching paths given changes in the interdomain paths conditions. We will evaluate how frequently the paths selection algorithm needs to switch paths when each edge in the currently used path is cut. In addition, we will also measure the distance between the latency obtained by the initial path and that obtained after each edge cut.

For evaluating the latency optimisation objective, we need a topology where the edges are labelled with a latency value. We could assign these values in a random manner with latency picked in a predefined range. We could also rely on synthetic topologies generated by the BRITE generator [MEDI01]. In addition, there is currently no Internet-wide model of the interdomain traffic. However, we know that the traffic distribution seen from stub domains is often skewed in the sense that a small fraction of the source/destination pairs is responsible for the majority of the traffic volume ([UHLI02], [FEAM03]). Another possibility is to rely on the empirical interdomain traffic matrices construction proposed in [CHAN05].

In addition to this, we will evaluate the benefits of using IP Tunnels to cross the Internet core. Using IP tunnels would allow to split the locator and identifier functions of IP addresses. That is, the Internet core and the leave domains would use different addresses spaces. The address space in the Internet core is composed of routable identifiers named locators while the address space for end-hosts is composed of locally routable identifiers. Locators must be globally advertised in the default-free zone while identifiers need not. This allows for reduction of the size of global forwarding and routing tables (FIB and RIB).

3.3.2.3 Control Parameters

Control Parameters		
Parameters controlled in the simulated topology	*Network size and shape*	The number of AS. The number of routers/AS. The number of business relationships between AS. The shape (width and depth) of the topology.
	Link delay	The delay associated with links in the topology.

3.3.2.4 Performance Metrics

Performance Metrics		
	FIB size	Size of FIB with different prefix allocation schemes.
	Number of prefixes	Number of prefixes allocated to each AS.
	End-to-End Delay	Improvements in terms of end-to-end delay due to the selection of the path with the lowest delay.
	Number of IP Tunnels	How many IP Tunnels must be established Internet-wide to reach a common objective.

3.2 q-BGP

3.2.1Objectives

The objective of this series of simulation experiments is to examine the effect of q-BGP policies and QoS attribute types and their calculation on the macroscopic behaviour of an inter-network of many ASs. The experiments will attempt to investigate the effect of various parameters, algorithms and configurations in a range of scenarios, including multiple network planes and adaptive policies based on network monitoring.

This series of tests aims to examine three major aspects of a QoS-enabled BGP environment:

- ***Scalability***, which aims to examine how the number of q-BGP messages depends on variables such as network size, topology, and traffic demand patterns.
- ***Stability***, which aims to consider the sensitivity of the q-BGP routing algorithms and protocol to changes in the inter-domain network and their ability to settle in a stable state. These changes could include inter-domain link failure or changes in demands.
- ***Performance***, which aims to consider the ability of q-BGP routing algorithms to find the optimal routes for a given demand matrix. Optimal is considered to be an inter-domain routing configuration that will accommodate demands with an acceptable level of QoS with minimal resource usage (e.g. inter-domain link usage).

Other aspects of inter-domain routing such as security and authentication are not considered in these tests.

3.2.2Simulation environment and methodology

The experimentation will be performed with an enhanced version of the “qBGPSim” q-BGP simulator, a preliminary version of which was developed as part of the IST MESCAL project. qBGPSim was chosen as it was created specifically to simulate qBGP and inter-domain links, while making approximations for intra-domain network behaviour and only models the significant parts of the system under test. Thanks to these simplifications qBGPSim scales to approaching a thousand or more ASs per simulation, depending on which statistics are to be collected and what operations are to be performed. qBGPSim also models traffic at an aggregate flow level and as such uses traffic models to approximate packet behaviour in the network. Alongside these models the simulator is also capable of modelling contention for capacity and can estimate the QoS characteristics of the delivered traffic. This therefore allows us to experiment with qBGP policies and QoS attributes and examine the delivered QoS in a scalable and repeatable way.

A series of scenarios will be simulated and various measurements captured, such as routing table dumps, internal variables and entire topologies, and then a post-simulation off-line analysis will be performed. The intention is to have a modular software approach which will allow us to use input files which are generated by other applications, such as topology generators, and generate output files, which can then be analysed further.

In every case, the simulation process is the same. The simulator is initialised with an input inter-AS connectivity topology and a NIA capacity matrix. The overall simulation procedure is then to apply demands to the network with a series of trigger events. Initially these events would be "add demand" and would entail the addition of demands sequentially by routing each one individually and allowing the network to settle before applying the next one. Having applied the demands a series of other events may be then triggered such as:

- the addition of further demands
- the removal of demands
- the destruction of an inter-domain link
- the creation of an inter-domain link
- an internal change within an AS

When such a trigger event occurs the simulator will then repeatedly perform the functions of each simulation element once in every simulation epoch. These epochs are repeated until either the network state settles, or the simulator goes through a fixed loop of states. For example, an AS would execute the incoming q-BGP message filtering and decision processes and then send out new q-BGP messages within an epoch; messages which will be acted on by the adjacent ASs in the next epoch. The state of the entire network is stored after each simulation epoch so it can then be analysed off-line to investigate issues such as the time (number of epochs) it took to settle in an alternative routing configuration following an inter-domain link failure (as part of the stability tests).

This approach gives us the flexibility to examine the behaviour in time of various simulated conditions within the same modular environment, as well as make it possible to easily add future events.

qBGPSim is the main program which performs the simulation of events which are specified in an input file. As part of the functional validation we use a series of smaller (4 to 10 node) manufactured topologies (such as fish) for which it would be easier to manually find the routing results. For the non-validation experiments, however, power-law compliant topologies generated by BRITE [BRITE] will be used. A series of other programs perform ancillary functions and can be used to generate the events and other required input data, specifically the IP prefix distribution (which subnets are assigned to which ASs), the network demands, and the values of available inter-domain capacity. The main programs are:

ASPrefixGenerator: This program generates a series of IP network address prefixes and subnet masks (in the form of VLSMs), which are then assigned to ASs for simulation. Ideally this generation shouldn't be random and should have as its input the AS inter-domain adjacency matrix so that it will be possible, at least in some cases, to perform IP network address aggregation on the NLRI fields in the qBGP UPDATE messages. Although qBGPSim does not currently support address aggregation, the IP prefix distribution should be non-random in the case where it may be implemented. The current implementation of this program first identifies the centre of the input topology and from there traverses the network outward toward the edges, splitting and allocating subnet addresses along the path to the edge. This results in a distribution which is perfectly aggregatable, but only along the same exact distribution tree that was used to create it; most other aggregation paths will either not work, or will be excessively general.

DemandFactory: This is a program which generates a series of network demands which are defined by two end-points (specifying the source and destination network) and a bandwidth. The bandwidth distribution is currently uniformly distributed, and the parameter to the program is the total bandwidth to assign to demands.

NIAFactory: To correctly simulate q-BGP routing policies it is required that the offline traffic engineering of inter-domain links has already happened. It is not enough to just randomly assign capacities to inter-domain links, as this capacity will not be in "useful" locations. It is essential that this is approximately correct as it is really this inter-domain capacity that is being optimised for. Rather than implementing an entire system and modelling business processes, this program takes as input a demand matrix and routes the demands across the inter-domain topology. The total capacity used on each inter-domain link is then found and this becomes the base-line NIA capacity, and the output of this program. To prevent the shortest path always being the best route the demands are allocated away from the shortest path. This is achieved by first routing the demands along the shortest path and then artificially increasing the link weights of the most loaded inter-domain links. After increasing the link weights the demands are then routed again, based on the new weights, and the output of this program is the total bandwidth used at each inter-domain link. In this arrangement the q-BGP process in qBGPSim must now actively seek this available capacity to perform well in terms of delivered QoS. Since it would be extremely difficult to discover the exact routing configuration that NIAFactory used to generate the inter-domain capacities qBGPSim scales all of the link capacities and therefore increases the number of alternative paths, and makes differentiation of qBGP policy efficacy easier.

QBGPSim: This is the main program and takes as input a NIA capacity matrix, which forms the main resource for which we are trying to optimise for, a list of IP network prefixes and the ASs to which they belong, for routing purposes, and most importantly a list of events which are to be simulated. The events in the simulations events file include, but are not limited to the adding and removal of demands, the breaking and making of inter-domain links, events that control AS policy and simulation control events, like the start and stop of the simulation.

3.2.3Control Parameters

Control Parameters		
	Network size	This is the number of ASs in the network being simulated.
	Number of inter-domain links per AS	The ratio of inter-domain links to the number of ASs.
	Overall traffic demand	The volume of inter-domain traffic demands in the traffic matrix.

	Intra-AS Delay Range	This is the upper and lower bounds of the random values assigned to delays across an AS.
	q-BGP Policies and strategies	The q-BGP policy set in use as well as the QoS-attributes used.
	Dampening strategy	The policy used to generate and receive q-BGP messages for the purpose of dampening routing oscillations. Variables include qBGP_Update_param_threshold, which would specify the threshold value of incoming QoS parameters (contained in q-BGP update messages) for the message to be considered.
	NIA over-provisioning factor	The multiplication factor applied to the standard NIA capacities. Standard NIA capacities are generated from demand matrix (controlled variable). Values above 1.0 are therefore over-provisioning, and values below are under-provisioning.

3.2.4Performance Metrics

Performance Metrics		
	Delivered QoS	This is the actual QoS that the demands receive, expressed as the average and standard deviation of QoS attributes, e.g. delay. The fraction of the offered bandwidth which was actually delivered, averaged across all demands is also considered. A delivered bandwidth fraction of 1.0 would imply that all the bandwidth of the demands was successfully delivered.
	Network Utilisation (Average and Standard Deviation)	This is a measure of the accuracy and load balancing of the routing solutions created by the routing algorithms. This is expressed as the mean average and standard-deviation of the utilisation of inter-domain links.
	Convergence time	The number of simulation cycles/epochs required before the network settles in a steady state and no more q-BGP messages are being sent, or internal AS variables are changing. This condition could however never be reached in the case of network oscillations, in which case the loop attractor size should be considered.
	Loop Attractor size	In the case of the network not finding a stable solution and oscillating through a series of states this is the number of states that it oscillates through. It can be a measure of the extent of oscillation, i.e. whether it affects the entire network, or just a single AS or inter-domain link.
	Number of q-BGP messages	This is the total number of q-BGP messages sent since network initialisation. The delta of this value is the size of the messaging avalanche caused by a change in the network.

	Routing Table Size	This is the number of entries contained in the q-FIBs and q-RIBs.
	Number of Demands Affected	The total number of demands whose route has changed during the settling of the AS network into its final state.
	Number of ASs Affected	The number of ASs that have received or sent a q-BGP message as a result of the trigger event.

4 INTEGRATED PI ENGINEERING EXPERIMENTS

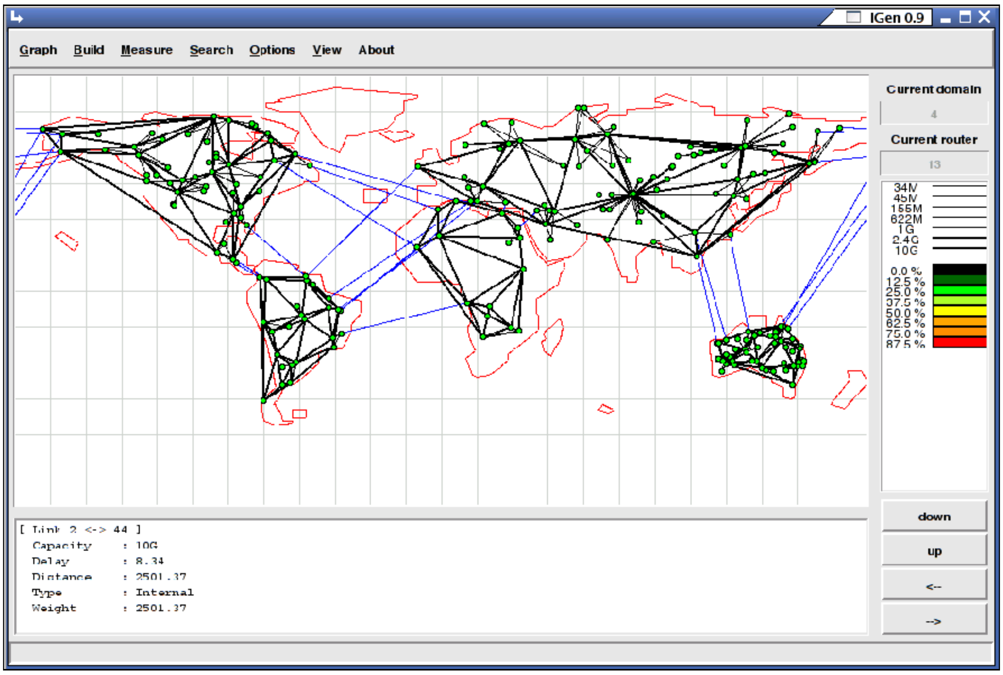
4.1 Overlay Routing

4.1.1 Objectives

The objective of this simulation part is to study the empirical performance of the proposed algorithms for end-to-end PI engineering using overlay routing schemes. Specifically, we will evaluate how the proposed schemes are able to achieve end-to-end QoS guarantees and also intra-/ inter-domain traffic optimisation purposes. Performance comparison will be carried out between the proposed schemes with non-overlay paradigms (e.g. MTR) as well as existing overlay approaches such as QRON [LI04]. In addition, we will also evaluate the resilience performance of the overlay topology in terms of edge-to-edge QoS availability in time of physical link failures (either inter-domain links or intra-domain links within any specific domains). A typical example in this case is to examine service assurance by the corresponding PI, e.g., whether the end-to-end delay between the source-destination pair can still be bounded as required, by following potential alternative overlay paths in time of the failure of any physical link.

4.1.2 Environments

Similar to the MTR scenario, there also exist two distinct components in overlay routing, namely overlay topology design and dynamic control of overlay routing. As it can be seen from D3.1, the input for the overlay topology design includes the physical network topology and the IGP/BGP routing configuration. We will use topology generators such as [IGen], [BRITE] and [GT-ITM] for creating multiple inter-connected domains, with each having a randomly generated intra-domain topology. Figure 24 gives a snapshot on a topology created by [IGen], which contains five autonomous domains (one for each continent). As for intra-domain routing configuration, we will use various sets of IGP link weights, including hop-counts, inversed bandwidth capacity and also the link weights that are randomly created. It should be noted that, as the design of the overlay topology does not assume the availability of the traffic matrix before hand, IGP link weights calculated through TM-aware optimisation schemes are not necessarily needed as the input. BGP routing attributes such as *local_pref* are also randomly created within each local domain representing independent inter-domain routing policies configured by each INP.



```
[ Link 2 <-> 44 ]
  Capacity     : 10G
  Delay        : 8.34
  Distance     : 2501.37
  Type         : Internal
  Weight       : 2501.37
```

Figure 24 Intra-/inter-domain topology created by IGen.

As far as overlay topology is concerned, the simulation metrics to be evaluated mainly include path diversity (i.e., the capability of detouring future traffic from default IP paths) and scalability related issues, such as the number of virtual overlay links needed compared to conventional approaches[3]. In order to evaluate the overlay routing control performance, the following information is to be obtained:

- The physical inter- and intra-domain topology;
- The IGP/BGP routing configuration;
- The designed overlay topology, including overlay nodes and overlay links;
- The traffic dynamics.

As at this moment we are not aware of any real inter-domain traffic matrix (involving multiple domains) available for public research, the traffic matrices and dynamics will be randomly generated for this test suite. The overview experiment methodology for INP-level overlay routing in NP provisioning and maintenance is illustrated in Figure 25.

[3] Detailed information will be released in a later stage.

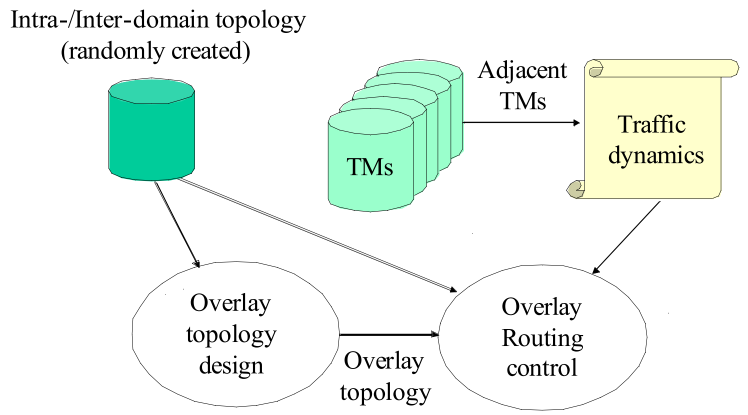


Figure 25 - Illustration of Overlay Routing simulation experiments.

4.1.3 Controlled Parameters

To be completed in D4.2.

4.1.4 Performance metrics

To be completed in D4.2.

4.2 Integrated multi-topology routing + qBGP

The experiment of integrated multi-topology routing + qBGP aims to evaluate the performance of end-to-end QoS service differentiation across multiple domains through offline intra-/inter-domain traffic engineering. Simulation based experiments will be performed after the fulfilment of the standalone multi-topology routing (section 2.1) and q-BGP (section 3.2) simulations. Detailed specification of this integrated experiment will be provided in D4.2 with the final results.

5 APPENDIX

5.1 The GEANT Topology

The GEANT set-up uses topology of the GEANT project. The GEANT network is a multi-gigabit pan-European data communications network, reserved specifically for research and education use. It is composed of 23 POP nodes interconnected using 38 (bi-directional) links. The GEANT topology (including actual bandwidth capacity and IGP link weights) is shown in Figure 26, while in Figure 4 there is a more synthetic view of the same network.

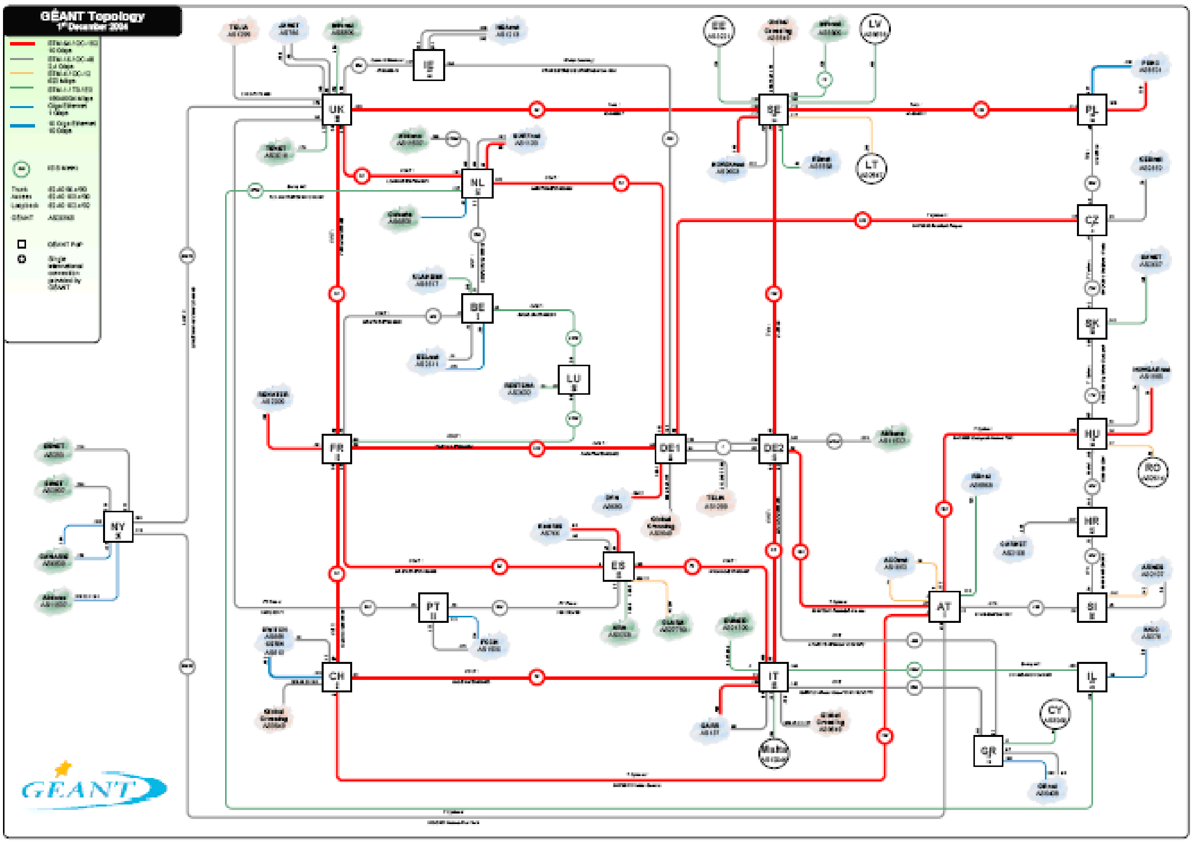


Figure 26 GEANT Topology (with actual IS-IS link weights) [GEANT].

6 REFERENCES

[AGAR05] S. Agarwal, J. Sommers and P. Barford. *Scalable Network Path Emulation*. In Proceedings of IEEE MASCOTS, September 2005.

[AN] Ambient Networks, http://www.ambient-networks.org

[BHAT01] S. Bhattacharyya, C. Diot, J. Jetcheva, N. Taft. POP-level and Access-Link-Level Traffic Dynamics in a Tier-1 POP. In Proceedings of ACM IWM 2001

[BRITE] The BRITE topology generator, http://www.cs.bu.edu/brite/

[BROI04] A. Broido, Y.Hyun, R. Gao, KC. Claffy. *Their Share: Diversity and Disparity in IP Traffic*. Proceedings of PAM Workshop 2004

[CALL05] M. A. Callejo-Rodríguez et al.: “A Decentralized Traffic Management Approach for Ambient Networks Environments", 16th IFIP/IEEE International Workshop on DSOM 2005, 145-156. Springer.

[CHAN05] H. Chang, S. Jamin, Z. Mao, and W. Willinger. *An Empirical Approach to Modeling Inter-AS Traffic Matrices*. Proceedings of ACM Internet Measurement Conference (IMC), 2005.

[D3.1] AGAVE deliverable D3.1, *Initial Specification of Mechanisms, Algorithms and Protocols for Engineering the Parallel Internets and Implementation Plan,* December 2006.

[FEAM03] N. Feamster, J. Borkenhagen and J. Rexford. Guidelines for interdomain traffic engineering. ACM SIGCOMM Computer Communications Review, October 2003.

[FERN04] D. Fernández, F. Galán, T. de Miguel. *Study and Emulation of IPv6 Internet Exchange (IX) based Addressing Models*. IEEE Communications Magazine, vol. 42(1), pages 105-112, January 2004.

[FORT00] B. Fortz and M. Thorup. *Internet Traffic Engineering by Optimizing OSPF Weights*. In Proceedings of IEEE INFOCOM, 2000.

[GEANT] The GEANT network, http://www.geant.net

[GT-ITM] The GT-ITM topology generator, http://www.cc.gatech.edu/projects/gtitm/

[KOHL00] E. Kohler, R. Morris, B. Chen, J. Jannotti and F. Kaashoek. *The Click modular router*. ACM Transactions on Computer Systems, Volume 18(3), pages 263-297, August 2000.

[IGen] http://www.info.ucl.ac.be/~bqu/igen/

[LI04] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," *IEEE Selected Areas in Communications,* vol. 22, pp. 29-40, 2004.

[LISP00] D. Farinacci, V. Fuller, and D. Oran, “Locator/id separation protocol (LISP),” *Internet Draft*, January 2007, available online at: www.ietf.org/internet-drafts/draft-farinacci-lisp-00.txt.

[MAPI] MAPI tool, http://mapi.uninett.no/

[MEDI01] A. Medina, A. Lakhina, I. Matta and J. Byers. *BRITE: An Approach to Universal Topology Generation*. In Proceedings of IEEE MASCOTS, 2001.

[MRT] MERIT Networks. The Multi-threaded Routing Toolkit. http://www.merit.net.

[NS] The Network Simulator –ns-2. http://www.isi.edu/nsnam/ns

[NUCC03] A. Nucci, B. Schroeder, S. Bhattacharyya, N. Taft, C. Diot, *IGP Link Weight Assignment for Transit Link Failures*, Proceedings of International Test Conference (ITC), August 2003

[QUAGGA] Quagga. http://www.quagga.net/

[QUOI05] B. Quoitin and S. Uhlig. *Modeling the routing of an Autonomous System with C-BGP*. IEEE Network, Volume 19(6), November 2005.

[RAMO02] F.J. Ramón-Salguero, J. Enríquez-Gabeiras, J. Andrés-Colás and A. Molíns-Jiménez. *Multipath Routing with Dynamic Variance*, COST 279 Technical Report TD02043, 2002.

[RODR07] J. Rodríguez Sánchez, M. L. García Osma, A. J. Elizondo Armengol, M. Boucadair. *A Lightweight Traffic Management Approach for Service Differentiation*. The Third International Conference on Networking and Services ICNS 07, 2007.

[SUBR02] L. Subramanian, S. Agarwal, J. Rexford and R. H. Katz. *Characterizing the Internet hierarchy from multiple vantage points*. In Proceedings of IEEE INFOCOM, 2002.

[TOTEM] The TOTEM Toolbox. http://totem.run.montefiore.ulg.ac.be/

[UHLI02] S. Uhlig and O. Bonaventure. Implications of interdomain Traffic Characteristics on Traffic Engineering. In European Transactions on Telecommunications, special issue on traffic engineering, 2002.

[UHLI06] S. Uhlig, B. Quoitin, S. Balon and J. Lepropre. *Providing Public Intra-domain Traffic Matrices to the Research Community.* ACM SIGCOMM Computer Communication Review, Vol. 36, No. 1, pp83-86, 2006.