AGAVE

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

IST-027609

D4.2 Experimental Results: Validation and Performance Evaluation of Parallel Internets and Connectivity Service Provisioning Components

Document Identifier: AGAVE/WP4/D4.2	
Deliverable Type: Report **Deliverable Nature:** Public	**Contractual Date:** M30 **Actual Date:** 18 June 2008

Editor:	Luigi Iannone, UCL.be	
Authors:	*TID:*	A. Elizondo, C. García, Ó. González, G. García, P. Montes, F. J. Ramón
	FTR&D:	Pierre Alain Coste, Bruno Decraene
	Algo:	Sevi Aggeletopoulou, Panos Georgatsos
	UCL.uk:	David Griffin, Suksant Sae Lor, Eleni Mykoniati
	UniS:	Ning Wang, K. H. Ho, M. Amin, M. Howarth
	UCL.be:	Luigi Iannone, Damien Saucez, Olivier Bonaventure

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

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

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 of the proposed solutions. It involves the setting-up of the required experimentation infrastructure, test-beds (including a pan-European integrated test-bed among the different partners) 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, agreement handling functions, intra- and inter-domain engineering techniques, algorithms and protocols.

Prototype experimentations are carried out in either a test-bed or simulated environment, as appropriate for the entity under test. Test-bed-oriented experimentations, besides validating, through proof-of-concept implementation and deployment, the specified functionality, proof that the AGAVE approach can be incrementally deployed on the existing Internet. The benefits introduced concerns lower-level functional aspects such as QoS routing/forwarding or resilience mechanisms. The purpose of simulation-based experiments is to allow testing the proposed solution in an extensive and large scale context, in terms of network topologies, traffic patterns, and operating variables, which would be impractical to be performed on a real test-bed.

In the above context, WP4 is concerned with the coordination of the different experimentation activities so that to ensure a consistent evaluation methodology; the setting-up and maintenance of the appropriate test-beds 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 test-bed and simulation platforms; the testing and validation of the functionality and the performance of the developed mechanisms, algorithms and protocols in test-bed and simulation platforms. This includes the integrated test-bed developed by the AGAVE consortium in a common effort.

Work in WP4 has been 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, test-bed 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 Test-bed-based Prototype Evaluation; it organizes and maintains test-bed platforms, integrating the prototypes developed within WP2 and WP3 activities. It undertakes the test-bed-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 is the project deliverable D4.2. It includes the description of the performed experimentation activities and their main outcomes.

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 the experiments performed for testing the solutions and mechanisms specified for realizing 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 test-beds (both local TID test-bed and AGAVE Integrated test-bed) and simulation environment: MRDV is a routing technique to enable intra-domain multi-path routing, used for realizing 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 realizing NPs with different QoS levels.

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

- q-BGP experimentation in a simulation environment: q-BGP is a means for conveying QoS information in BGP messages and provides enhanced QoS-based route selections, enhancing traditional BGP, that contributes to realizing PIs with different QoS levels.
- Resilience-aware BGP/IGP TE interactions in a simulation environment: an approach for maintaining optimized traffic distribution condition and QoS assurance on network link failures.
- BGP Planned Maintenance experimentation in test-bed: an approach for improving availability on disruptions due to maintenance by carrying out planned BGP session shutdowns.
- IP tunneling experimentation in test-beds (both local UCL.be test-bed and AGAVE integrated test-bed) and simulation environment: an approach for inter-domain routing and traffic control through IP tunneling mechanisms between cooperative remote domains; the work is aligned with the ongoing work at IETF/IRTF on locator/identifier separation.
- *Integrated PI engineering experiments:* It describes in details the AGAVE integrated test-bed. The specific experimental results are described in the section concerning the specific mechanism that has been tested. However, this section contains a description of some scenario that can be realized on the integrated test-bed thanks to the various mechanism developed in the project. These scenarios description aim at explaining how a selected subset of the techniques specified can be used to realize NPs and to horizontally bind these NPs, belonging to different INPs, to form Parallel Internets.

The deliverable concludes the work of activity AC4.2 and AC4.3.

2 NP ENGINEERING EXPERIMENTS

2.1 Multi-Topology Routing

2.1.1 Objectives

The objective is to evaluate the overall network performance and operational efficiency of the proposed multi-topology IGP routing that aims to tackle unexpected traffic dynamics and network failures. Comparison will be done between our proposed schemes with existing approaches and also the optimal values in order to find the gap to optimum.

2.1.2 Experimental Setup

In order to evaluate the performance of *AMPLE*, we use the real topologies and traffic matrices from the GEANT [GEANT] and Abilene [ABILENE] networks. The GEANT network topology contains 23 PoP nodes and 74 links, most of which are OC48 (2.5Gbps) and OC192 (10Gbps), but it also has a few links with low capacity of 155Mbps. The traffic matrices have been derived every 15 minutes for several months. We present results based on a 7-day long traffic matrices dataset obtained from [TOTEM]. The Abilene network topology contains 12 nodes and 30 links, most of which are OC192, but the link between Indianapolis and Atlanta has 2.5Gbps capacity according to the dataset provided by [ABILENE]. By studying the traffic matrices of these two operational networks, we realized that the patterns of their dynamics are very different. The GEANT traffic matrices follow a quite regular pattern on a daily basis, while the Abilene ones do not have this characteristic.

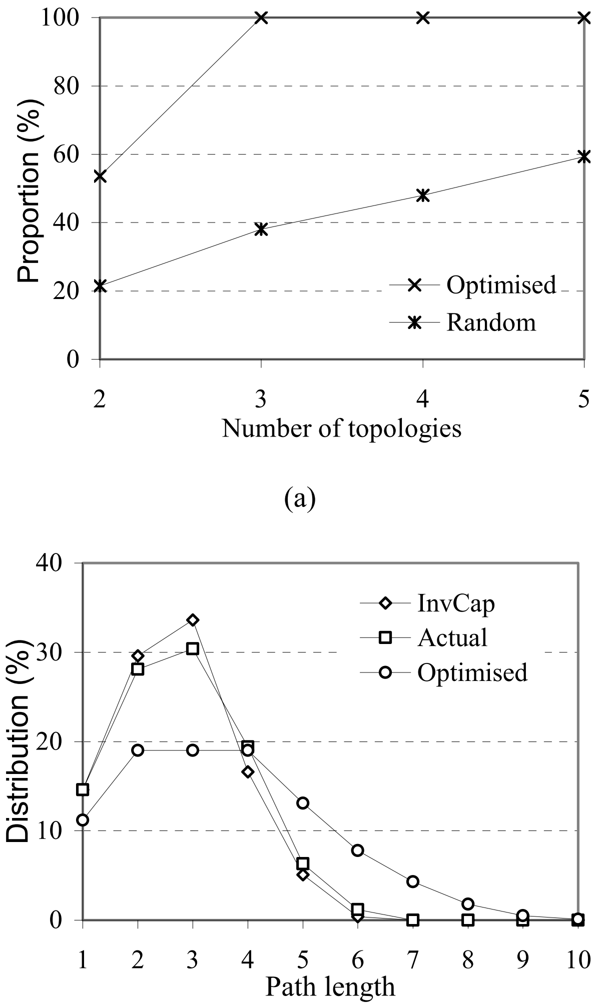
2.1.3 Test Results

2.1.3.1 Path Diversity Performance

In this section we present our simulation results for the offline MT-IGP link weight optimization. We compare the following approaches in our IGP path diversity evaluation:

- ***Actual*:** The actual link weight setting in the current operational networks;
- ***InvCap:*** Setting link weights in the inversed proportion to bandwidth capacity.

The performance metric we use to evaluate path diversity is the proportion of source-destination pairs that can successfully avoid any critical link with *FDoI* (i.e., shared by all RTs). Figure 1(a) shows the path diversity performance (i.e. the proportion of source/destination pairs that can fully avoid critical links) in the GEANT topology with: (1) optimized MT-IGP link weight setting for maximizing intra-domain path diversity (*Optimized*), and (2) random link weight setting in all RTs (*Random*). From the figure we can see that the optimized link weight setting substantially outperforms the random solution in terms of path diversity. More specifically, our algorithm is able to guarantee 100% avoidance of critical links shared by all topologies with only three RTs. In this case, in an event of network congestion, the associated sources are always able to remark their local traffic to enforce alternative IGP path selection to bypass the congested link. Figure 1(b) shows the path length distribution performance of individual schemes, including the actual link weight setting (*Actual*), proportional to inverse bandwidth capacity (*InvCap*) and our proposed GA-based scheme (*Optimized*), all with three RTs. We can see that our proposed algorithm leads to some longer paths due to the efforts for maximizing path diversity across multiple RTs, which accounts for some increment in the overall network cost (see section 2.1.3.2).



(a)

(b)

Figure 1 MT-IGP link weight setting performance (GEANT)

Figure 2 shows the corresponding performance in the Abilene network. Again, we can see that larger number of RTs lead to higher path diversity. It should also be noted that the overall path diversity performance is not as good as that of the GEANT topology. This is mainly due to the fact that one of the PoP nodes in Atlanta has node degree of one, thus it is not possible to provide path diversity for it. If we ignore this node, the corresponding path diversity performance can reach 100% with four RTs. The lower path diversity of Abilene is also due to its lower mean node degree (2.5) compared to that of GEANT (3.2).

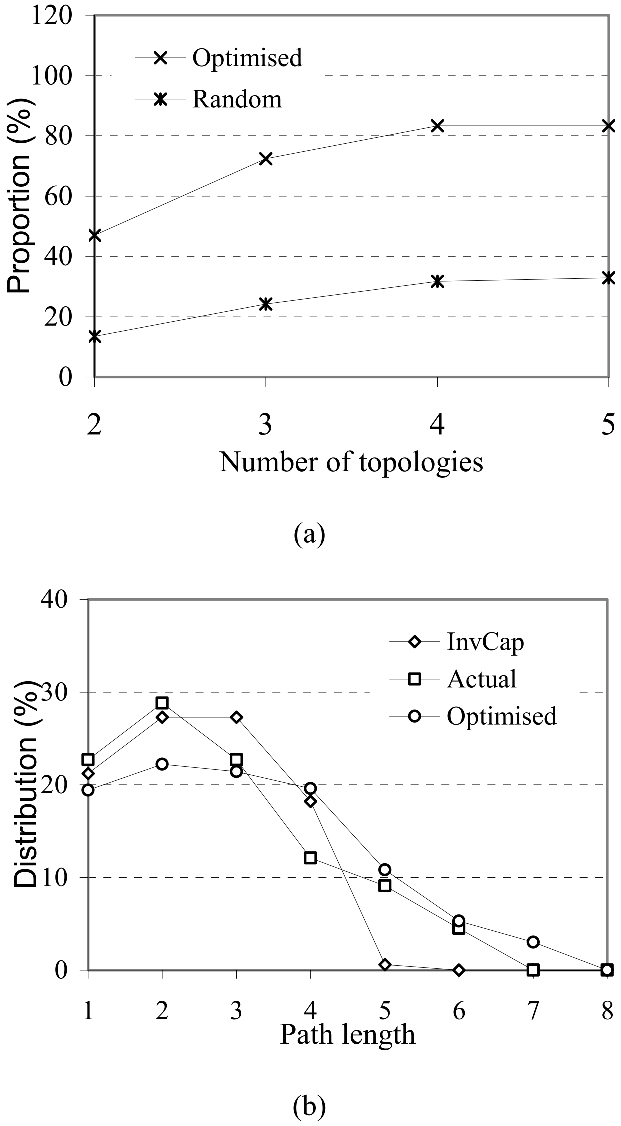


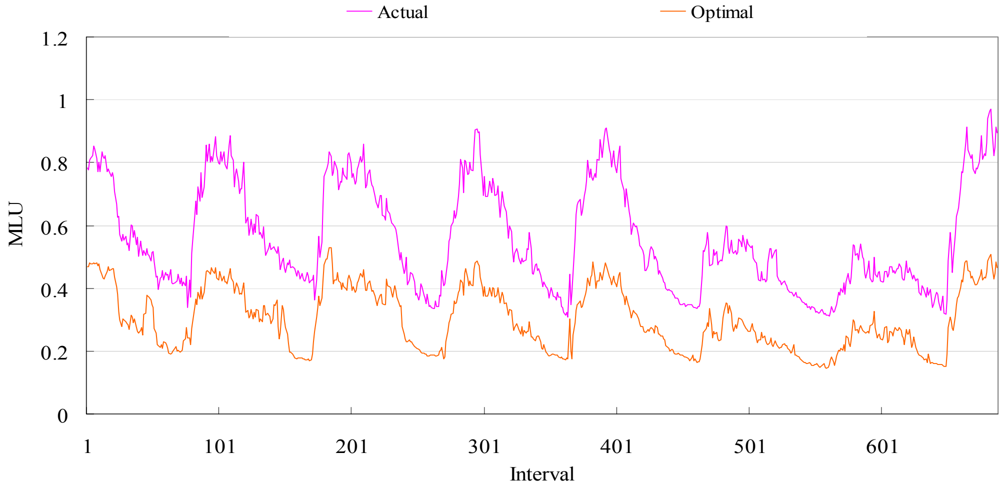
Figure 2 MT-IGP link weight setting performance (Abilene)

2.1.3.2 Adaptive Traffic Control Performance

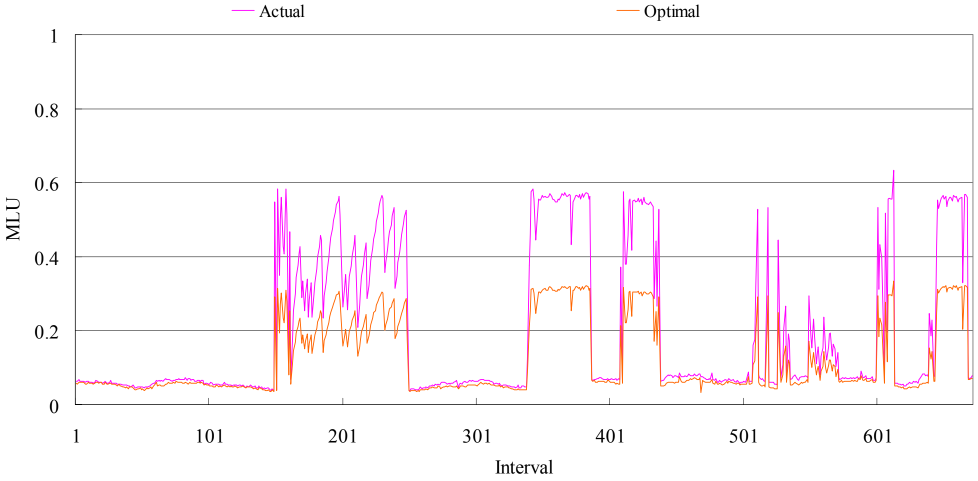
We compare the following approaches in our adaptive TE evaluation:

- ***Actual***: The actual link weight setting in the current operational networks.
- ***Multi-TM***: We use the TOTEM toolbox [TOTEM] to compute a set of link weights for multiple traffic matrices. The objective is to make the IGP TE robust to traffic demand uncertainty. Specifically, the link weights are computed at the beginning of each day based on the sampled traffic matrices (one per hour) on the same day of the previous week.
- ***AMPLE-n***: Our proposed adaptive TE algorithm that runs on top of *n* optimized MT-IGP routing topologies.
- ***Optimal***: As the baseline for our comparisons, we use the GLPK in the TOTEM toolbox to compute optimal MLU for the given topologies and traffic matrices.
- ***RandomWeight-n:*** Our proposed adaptive TE algorithm that runs based on *non-optimized* MT-IGP routing topologies, each using randomly generated link weights. The intention of showing this scenario is to evaluate the performance of *ATC* without the support from *OLWO*.

We first present in Figure 3 the MLU achieved by *Actual* and *Optimal* for all the TMs of the GEANT and Abilene network during a week. As shown in the figure, the performance gap between *Actual* and *Optimal* is very large. This reveals that network resources are far from being utilized at the maximum efficiency according to the current actual link weight configuration. In order to minimize or avoid if possible potential congestion caused by unexpected traffic spikes, it is desirable to maintain the network utilization as close to *Optimal* as possible.

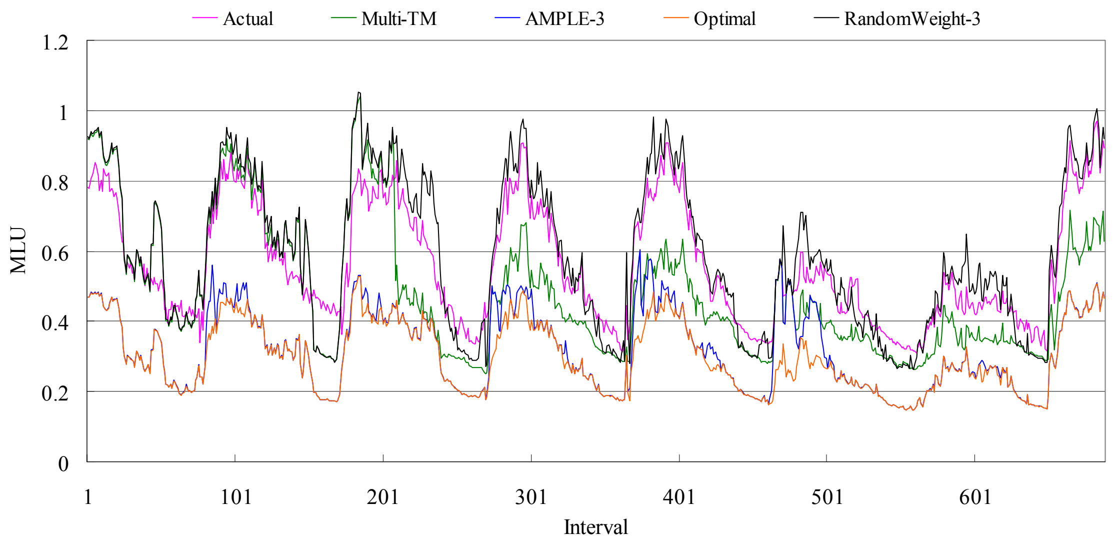


(a) GEANT

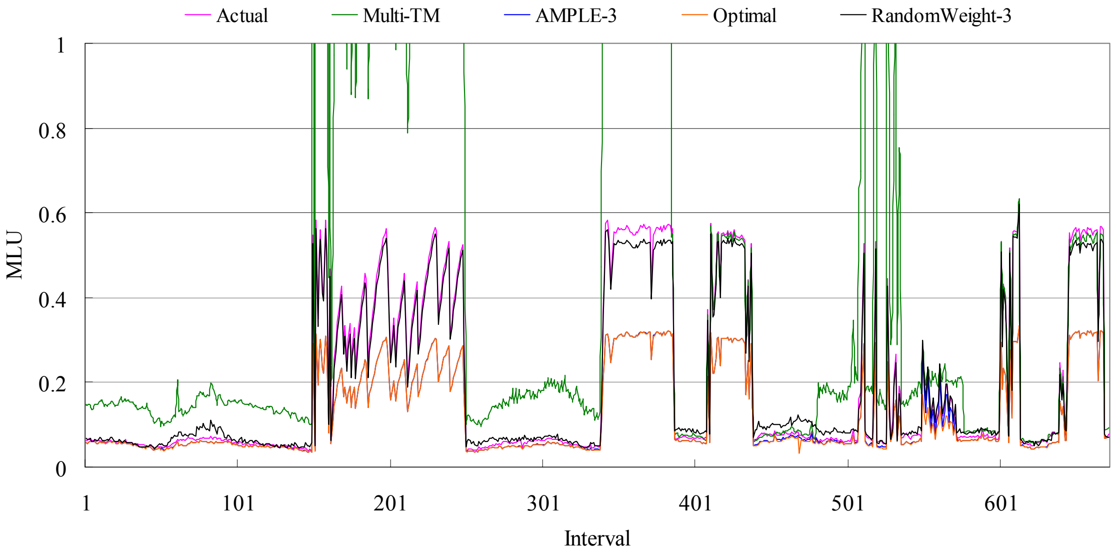


(b) Abilene

Figure 3 MLU achieved by actual and optimal link weights

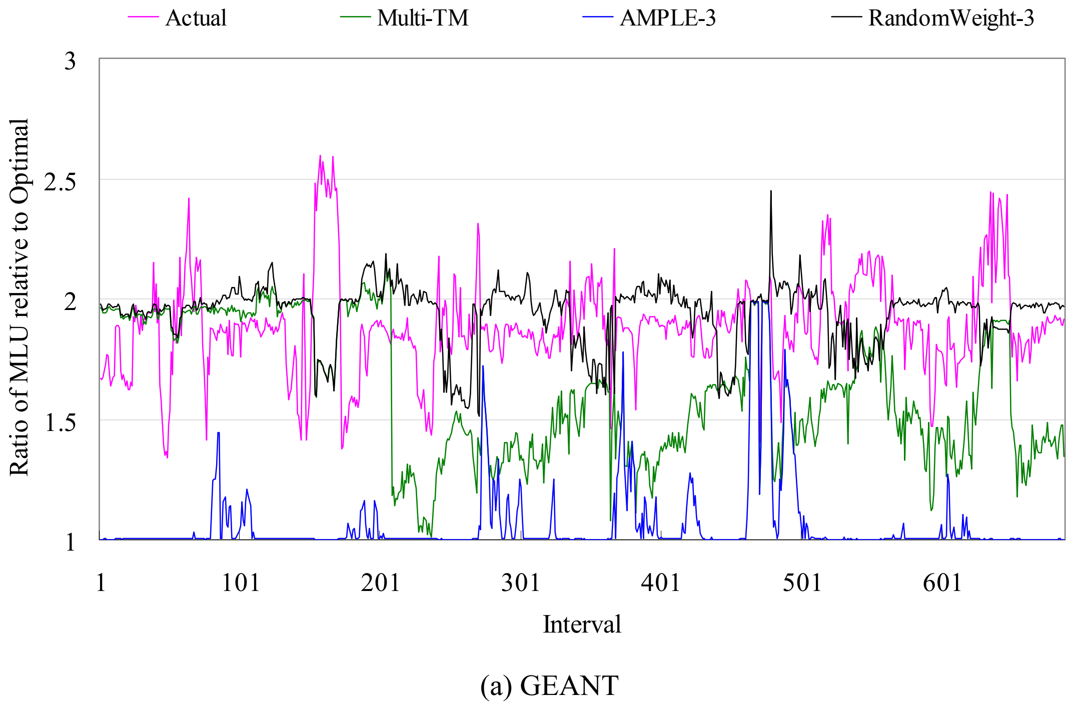


(a) GEANT

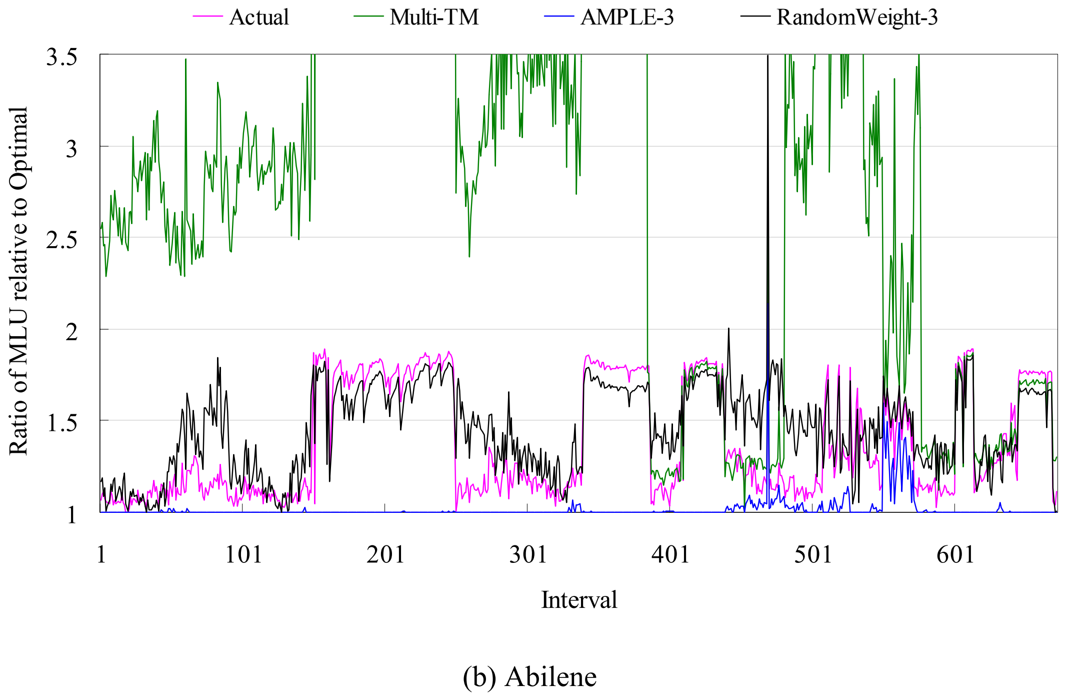


(b) Abilene

Figure 4 MLU over one week interval



(a) GEANT



(b) Abilene

Figure 5 Ratio of MLU relative to *Optimal* link weight

Figure 4 plots the actual MLU achieved by different schemes over one-week traffic traces. To see more clearly how each of these schemes performs compared to the optimal solution, we plot the corresponding ratio of MLU relative to *Optimal* in Figure 5. The ratio is calculated as the MLU of a specific method divided by that of *Optimal*. The closer the ratio to 1.0, the closer the achieved MLU is to the optimum. For illustration purposes, we only included with the scenario of three routing topologies (i.e. *AMPLE-3* and *RandomWeight-3*). Nevertheless, Table 1 shows additional statistics on both GEANT and Abilene networks with the number of topologies varying from 2 to 4. Specific MLU performance metrics in the table are defined as follows:

- **Average maximum link utilization (AMU) –** the average value of the MLU across all the traffic matrices during the seven-day period;
- **Highest maximum link utilization (HMU) –** the highest value of the MLU across all the traffic matrices during the period.
- **Proportion to near-optimal performance (PNO) –** the percentage over all the TMs in which *AMPLE* can achieve near-optimal performance. We define here the meaning of near-optimal to be the MLU that is within 3% gap to the optimality.

A common observation from the figures and table is that based on the optimized MT-IGP link weights, AMPLE is able achieve near-optimal MLU for most of the traffic matrices with a small number of routing topologies. On average, the MLU of Actual is nearly twice that of Optimal. This can be explained by the fact that the actual setting of link weights in the GEANT network mainly takes delay into account. Although Multi-TM perform better than Actual, their gaps from Optimal are still significant with the relative ratio being around 1.5. We further analyze the relevant statistics in Table 1. In the GEANT network, the Actual link weight approach produces AMU that is 86% higher than that of the optimal value. With AMPLE, the AMU varies between 0.1% and 43% depending on the number of routing topologies that are used. The larger the number of routing topologies, the closer to the optimal performance can be achieved. For the PNO metric in Table 1, if AMPLE is based on two routing topologies, the value is only 13.1% but it still outperforms significantly the Actual approaches. We can now start to see the practical usefulness of our approach in improving network utilizations: When the number of routing topologies increases to three, the PNO boosts up to 78.3%. With 99.6% of all the traffic matrices, AMPLE achieves near-optimal performance with four routing topologies. These results reveal that, for the GEANT network, AMPLE has very high chance in achieving near-optimal TE performance under any scenario of traffic matrix with four routing topologies. Our experiments based on the Abilene network also show similar results.

We also observed that the *Multi-TM* approach does not achieve good performance in minimizing the MLU according to Figure 5. There are two reasons for this. First of all, the ultimate objective of *Multi-TM* is to minimize the network cost represented by a piece-wise linear function [FORT00] rather than specifically minimizing MLU. Second, even if multiple traffic matrices with different pattern characteristics are considered in the link weight optimization, unexpected traffic spikes may still introduce poor TE performance. This is especially the case in the Abilene scenario (see HMU in table 1). Finally, we also noticed from Figure 4 and Figure 5 that OLWO plays an important role in minimizing MLU in the sense that optimized IGP link weight setting effectively provides diverse paths for ATC to perform dynamic traffic control. Non-optimized IGP link weights still lead to sub-optimal network performance despite the provisioning of multiple routing topologies.

Optimization Method	**GEANT (%)**			**Abilene (%)**		
	AMU	**HMU**	**PNO**	**AMU**	**HMU**	**PNO**
Optimal	30.05	52.82	-	12.2	33.42	-
Actual	55.74	96.91	0	19.59	63.24	1.19
Multi-TM	48.56	100.1	0.44	53.2	230	0.15
RandomWeight-3	59.01	105.17	0	19.43	62.01	2.23
AMPLE-2	42.9	94.61	13.08	18.61	60.96	64.14
AMPLE-3	31.95	60.36	78.34	12.36	33.44	88.69
AMPLE-4	30.08	52.88	99.56	12.4	49.6	97.77

Table 1 Probability of achieving near-optimality

Minimizing network cost is another important TE objective. To evaluate this, we adopt the commonly used piece-wise linear function [FORT00] to indicate the actual network cost. By using this cost function, the two objectives of minimizing network bandwidth consumption and improving load balancing are taken into account simultaneously. Figure 6 shows the corresponding performance in the GEANT and Abilene networks. In overall, *Multi-TM* is the best performer since it optimizes the network cost as the primary objective. Although *AMPLE* has higher network cost due to the trade-off to path diversity, the increase is small and acceptable.

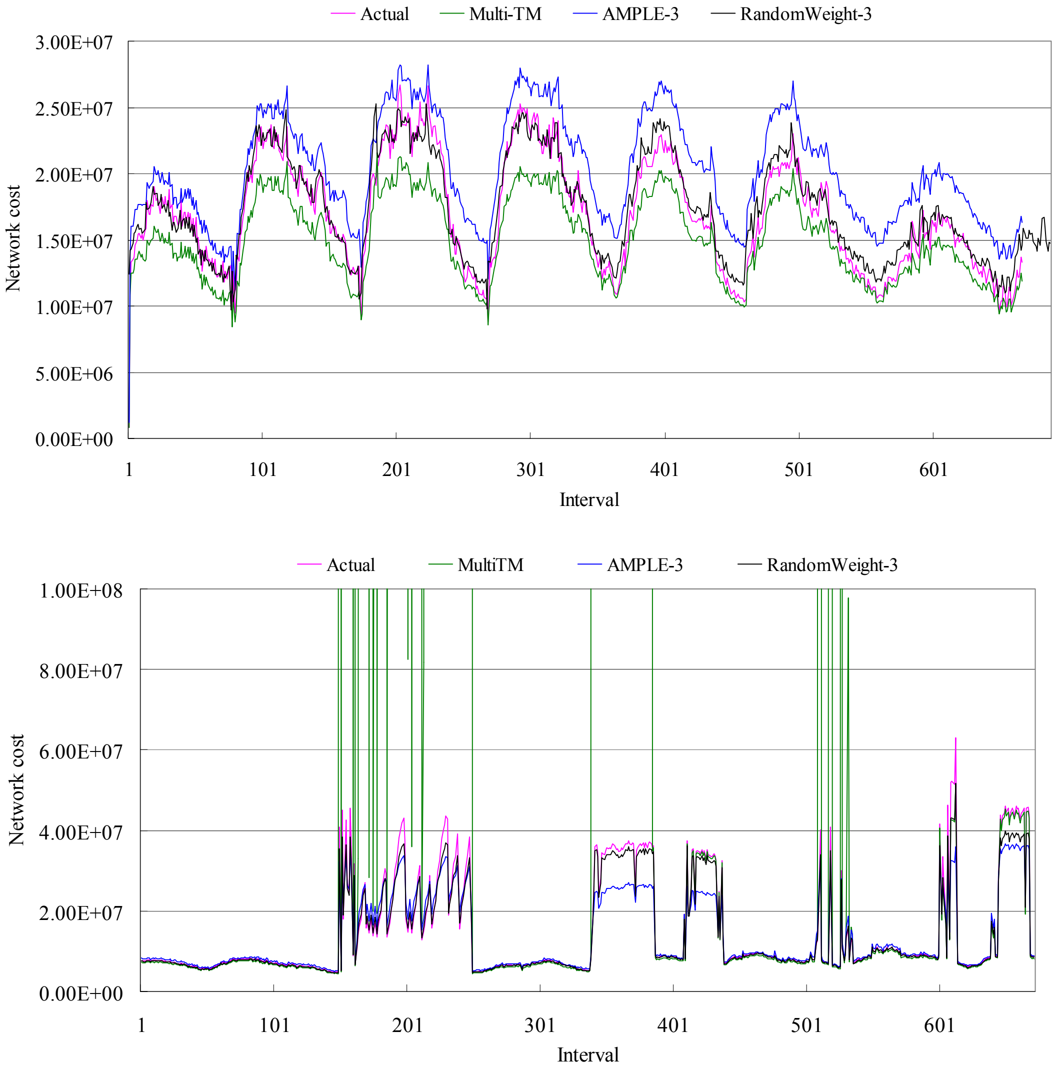
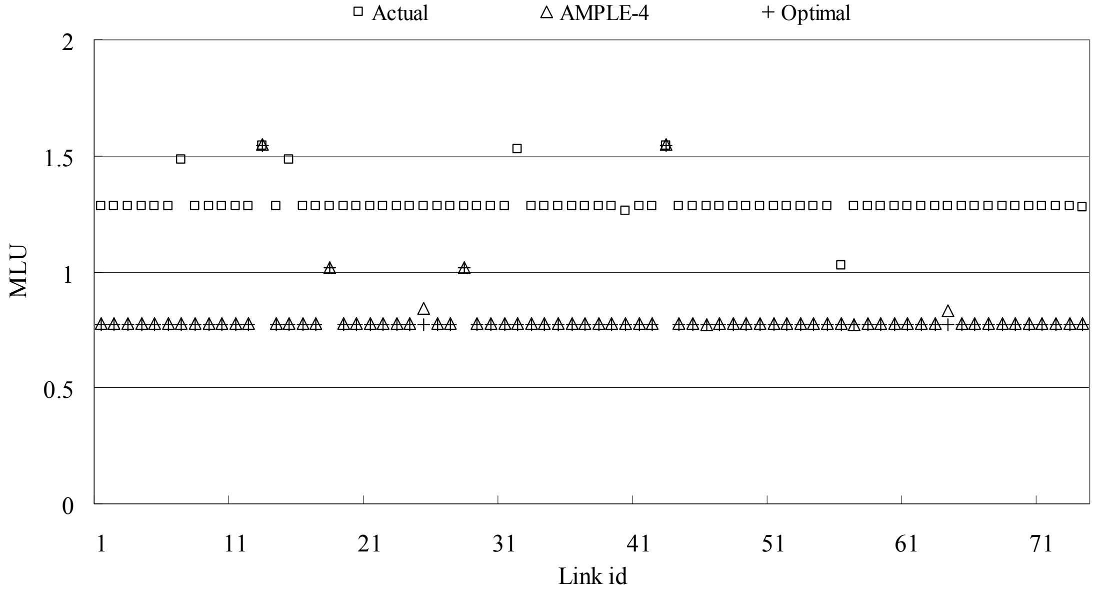
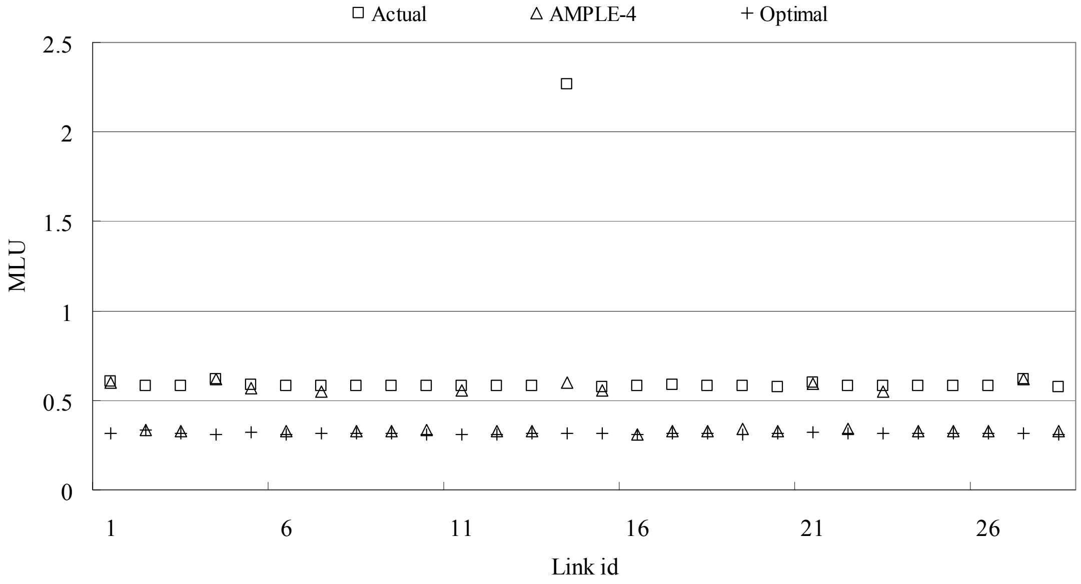


Figure 6 Network cost in GEANT

Link failure is a common cause of network congestion. Without making IGP TE robust to link failures, the network may experience post-failure congestion due to uncontrolled IGP rerouting. Figure 7 plots the MLU after the failure of each link (denoted by link id in the x-axis) under a single traffic matrix. The figure shows that, for most of the failures, *AMPLE* achieves near-optimal MLU (stays well below 100%), whereas the other approaches suffer from congestion. Once again, *AMPLE* is able to dynamically re-optimize the post-failure network utilization thanks to its capability of dynamic adjustment of splitting ratios across multiple routing topologies.



(a) GEANT



(b) Abilene

Figure 7 MLU after each link fails

On the other hand, we can also see in the figures that not all the post-failure MLU of individual links can be improved in both topologies. This phenomenon is mainly due to the topology characteristics of the networks. For example, in GEANT, the PoP node in Israel is connected to the Netherlands and Italy with two low capacity links of 155Mbps (link id 13 and 43). During the normal state, traffic destined to (or sent from) Israel can be appropriately distributed over the two distinct links in order to achieve load balancing. Even under this situation, these two links often become the bottleneck of the entire network in the normal state. In case of either link fails, the affected traffic has no alternative but to use the other one to carry the traffic from/to Israel, thus creating severe congestion. This phenomenon also appears in Abilene network where many nodes have low node degree of two.

Finally, we show the computational efficiency of *AMPLE* by examining the following two metrics associated with each distinct interval: (1) the actual number of iterations (bounded by *K*) based on the splitting ratio configurations for the previous interval and (2) the proportion of total number of traffic flows whose splitting ratios need to be adjusted at each interval. In our evaluations based on four routing topologies, on average, *AMPLE* needs 61 iterations to adjust only 2% of total traffic flows at each traffic interval in the GEANT network. For the Abilene scenario, it takes 21 iterations to adjust 7% of the traffic flows. We believe that the higher proportion of flows adjusted in Abilene is mainly due to its more irregular pattern of traffic dynamics with frequent steep upsurges between adjacent traffic matrices. In general, these results reveal that, within a few iterations, *AMPLE* only needs to adjust a small proportion of traffic flows in order to regain near-optimal performance.

2.1.4 Conclusions

Through our thorough simulation-based experiments with two real operational networks and traffic traces, we found that the proposed AMPLE scheme for handling unexpected traffic dynamics has a high chance of achieving near-optimal performance (within 3% gap from the optimal value) with only a small number of routing topologies. We believe this is a significant breakthrough in enabling dynamic traffic engineering on top of plain IP based routing protocols without the support of MPLS. In addition, our experimental results also indicates that the AMPLE scheme is also efficient in dealing with single link failures in the sense that optimized post-failure performance can be also achieved.

2.2 MRDV

2.2.1 Objectives

The main objectives of TID's work in WP4 are to evaluate the applicability of MRDV with the new extensions to build Network Planes and to deploy a test-bed with the implementation of MRDV with and without Classes of Service support in order to test the algorithm initially proposed, as well as testing a new load distribution module by means of simulation.

The first subsection describes the implementation of MRDV in Quagga software and the results of the first tests, as well as the validation of the results with the help of the simulator.

Next, follows the description of the new load distribution module behaviour and the simulation scenarios used to validate MRDV to support QoS in several different ways.

Finally, the last section shows performance tests of MRDV with CoS support in a scenario composed by seven nodes. It will be demonstrated through performance measurements (delay and loss rate) that the Network Plane differentiation provided by MRDV with CoS is able to support high load from Best Effort traffic while guaranteeing performance for Premium traffic.

2.2.2 Implementation of MRDV with CoS support on Quagga routers

2.2.2.1 Introduction

MRDV and its extended version to support multiple network planes have been implemented using routing software (Quagga) in a Linux system.

This section is structured as follows. First, the section 'Kernel aspects to consider when implementing multipath routing' introduces some aspects about multipath routing in Linux that are important to understand the software behaviour. Next, section 'Implementation of MRDV with CoS support' explains the internal mechanisms to implement MRDV in Quagga and the extensions to implement MRDV with CoS support. In section 'MRDV implementation tests', it is shown how the MRDV with CoS support behaves in a simple scenario. Finally, the behaviour of MRDV on this simple scenario has been validated by means of simulation.

2.2.2.2 Kernel aspects to consider when implementing multipath routing

Normally, kernel options are not set to work with multiple routes. To make this possible, it is necessary to configure some of these options:

1. IP_ROUTE_MULTIPATH
   To store the routes the routing tables specify a single action to be taken in a deterministic manner for a given packet. However, if the Linux kernel option IP_ROUTE_MULTIPATH is activated it becomes possible to attach several actions to a packet pattern, therefore specifying several alternative paths to route the packets matching the pattern.
2. CONFIG_IP_ROUTE_MULTIPATH_CACHED
   It is necessary to configure this option because by default multipath routing is not supported by the routing cache. If this kernel option is activated, alternative routes are supposed to be cached and on cache lookup a route is chosen according to internal Linux mechanisms. But because of a bug in Linux kernel, multipath routes with caching is broken for forwarded packets, and that makes it to break multipath routes for such packets merely by being enabled. It is necessary to deactivate the CONFIG_IP_ROUTE_MULTIPATH_CACHED kernel option to solve this problem. When this is done it will be possible to use multipath routing.

After deactivating the multipath cache, the traditional routing cache complements the routing tables functionality by saving which of the multiples paths to reach a destination (saved in the routing tables) should be used to route a package according to its characteristics. There are two different ways to flush this cache that will affect the MRDV implementation in Quagga behaviour:

1. The routing cache is flushed regularly in a time determined by the Linux system variable 'secret_interval'. In case there are more than one route with the same cost to reach a destination, this Linux variable allows us to specify if we want the kernel to choose the route every time it sends a packet, by setting secret_interval to 0, or if we want to keep always the same route, by setting secret interval to a very high period.
2. When a new route is added (and we are using multipath routing) or an old one is deleted, the routing cache needs to introduce major changes, so it is flushed automatically. In these cases, as the cache is deleted, it is possible that the kernel chooses a different path from the one that it was using to reach the same destination.

2.2.2.3 Implementation details

As it was explained in D3.1 MRDV allows the use of alternative paths to route traffic towards a destination when minimum cost paths are congested.

In order to explain the necessary components of the MRDV with CoS implementation, we will first introduce the implementation of the basic MRDV and then the necessary modifications to obtain the QoS support.

2.2.2.3.1 Basic MRDV

MRDV adjusts the load in each path according to the average load that the router detects the next hop of the optimal path towards the destination. MAPI software is used in the MRDV implementation in Quagga to measure the load in the optimal path interface, so the router software can calculate the variance associated to that load and, therefore, obtain the number of non-optimal paths that need to be included to or deleted from the routing table. Once the different paths have been calculated the routing table is updated adding the new paths and deleting the obsolete ones.

The internal variable 'MRDV_VAR_CALC' defines the periodical calculation of the variance parameters in the router output interfaces that will be used to determine all alternate paths to a destination. The MRDV software includes an internal structure to track if the routes have been included or not in the routing table, so it is able modify the routing table only if a modification needs to be done. This behaviour allows Linux kernel to keep the routing cache intact so the traffic flows can continue using the paths that were established before the timer was triggered. When traffic changes enough to make MRDV activate a new route or to delete one that was previously in the routing table, Linux kernel flushes the routing cache, so paths for all flows are reassigned. The MRDV software cannot avoid this path reallocation and it is a Linux kernel issue.

2.2.2.3.2 MRDV with QoS

When QoS is introduced in MRDV major modifications need to be done:

- Measure the traffic load by ToS. To be able to calculate the goodness of a route with a certain traffic load in the priority path, but only using the traffic that belong to a certain ToS and the one of higher priority classes it is necessary to measure the load independently for every ToS and in every interface.
- Modify variance calculation. When MRDV with QoS is implemented it is necessary to calculate variances for every ToS and according to the specifications in D4.1 this calculation is done by taking into account the traffic load of each ToS and the load from the ToS's that have a higher priority than this one.
- Use of multiple routing tables. It is necessary to store in different routing tables the routes for the different Classes of Service so every CoS has its own one. In Linux it is possible to use up to 256 different routing tables with the use of 'netlink' from the Quagga software that will assign the routes for every CoS to the right table. Therefore, the Linux box is able to use the right routing table for each ToS and it is possible to route through different paths every class of traffic.

- Mechanisms to classify the packets by ToS. In Linux it is possible to specify rules to check a certain routing table before checking the default routing table when a packet with a certain QoS is about to be sent, by means of the command 'ip'. This way, the system can be configured so every QoS will have its own routing table that will be checked automatically every time a packet needs to be sent.

Although other configurations are possible, the current implementation has been programmed to make the Linux kernel save the routes for the different classes according to Table 2, in which 'class 1' has the highest priority and 'class 5' the lowest one.

Class	ToS	Routing Table
1	0x10	51
2	0x8	52
3	0x4	53
4	0x2	54
5	Normal service	55

Table 2 Routing tables for current MRDV version

The 'Class' column shows the names that will be used in the next section, instead of specifying the ToS.

2.2.2.4 MRDV implementation tests

In order to test the MRDV implementation the following scenario has been configured. It has four nodes connected as can be seen in Figure 8, each one running the Quagga software. The only node running MRDV with QoS is agave4, the rest of nodes are running just OSPF. Link costs in the scenario have been set in a way that the optimum path to go from agave4 to agave2 is the path through node agave1.

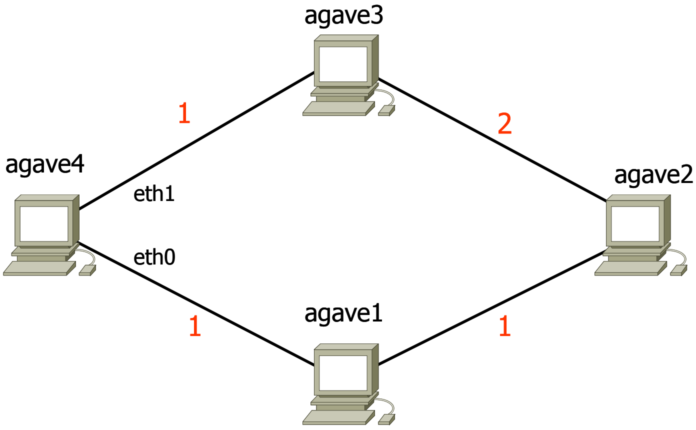


Figure 8 Test-bed used for validation of the MRDV implementation

Traffic is generated in 'agave4' using mgen, with destination 'agave2'. When agave4 starts sending traffic this goes through the optimum path (through 'agave1') and at a certain link load level, traffic is high enough to activate the alternative path and start routing through both paths.

The input traffic for 'class 1' (ToS 0x10) through the time is represented in Figure 9. This stair shape has been chosen in order to easily show how the alternative path is activated/deactivated for the different TOS. Input traffic for 'class 2' (ToS 0x8) is the same as traffic for 'class 3' (ToS 0x4) so both of them are represented in Figure 10.

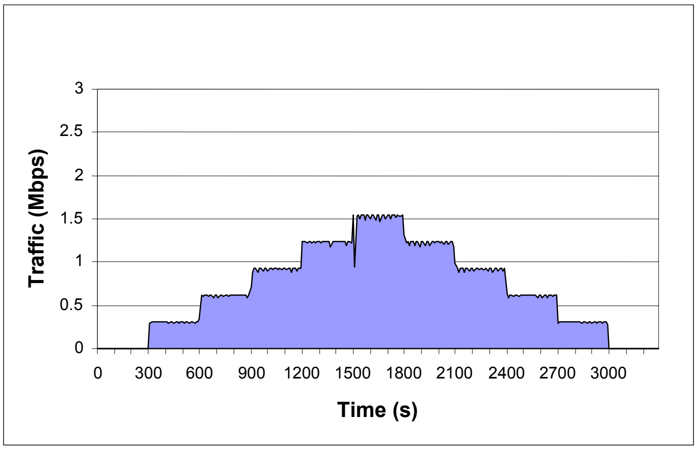


Figure 9 'Class 1' traffic

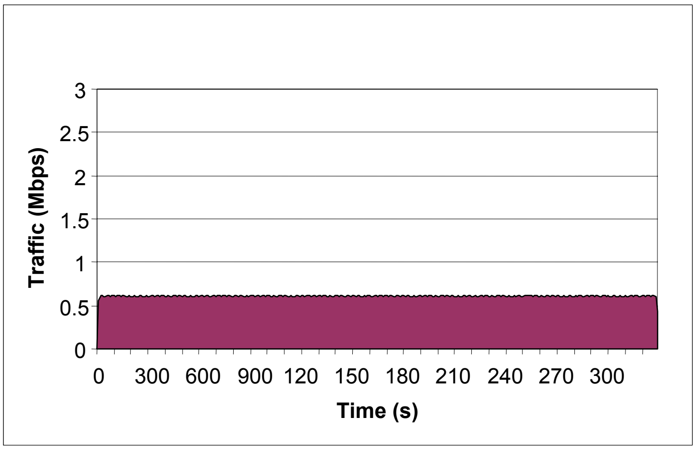


Figure 10 'Class 2' traffic /'Class 3' traffic

The expected behaviour of MRDV in "agave4" is summarized in Table 3. The table specifies the thresholds when the secondary path to reach agave2 is activated and deactivated. Both thresholds, 1.48 Mbps and 0.4 Mbps, have been obtained according to the procedure introduced in D3.1 to calculate the variance from the traffic load and taking into account the link costs depicted in Figure 8. Because of the MRDV hysteresis cycle, these two thresholds have not the same value.

Traffic measured in eth0 (high priority path)	Event	MRDV behaviour
Class 1	> 1.48 Mbps	Secondary path is activated for class 1
	< 0.4 Mbps	Secondary path is deactivated for class 1
Class1 + Class 2	> 1.48 Mbps	Secondary path is activated for class 2
	< 0.4 Mbps	Secondary path is deactivated for class 2
Class 1 + Class2 + Class 3	> 1.48 Mbps	Secondary path is activated for class 3
	< 0.4 Mbps	Secondary path is deactivated for class 3

Table 3 Expected MRDV behaviour

Next, the test results are shown for the traffic belonging respectively to classes 1, 2 and 3.

2.2.2.4.1 Test results for 'Class 1'

Class 1 is the one with the highest priority, so the traffic load for this class is the only data needed to decide the number of paths that must be used to route traffic, as can be seen in Table 3.

Figure 11 shows a purple line depicting the threshold to activate multiple routes (1.48 Mbps) and an orange line represents the threshold to return to a single route (0.4 Mbps). Because of the MRDV hysteresis cycle these two thresholds have not the same value.

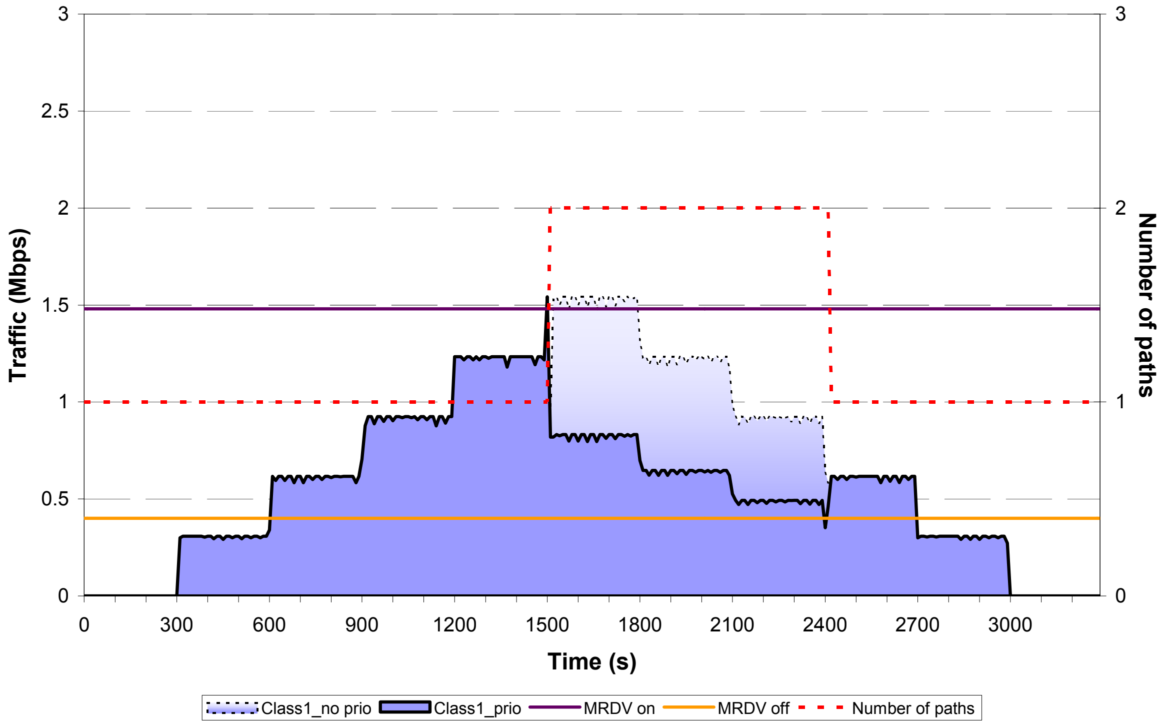


Figure 11 MRDV behaviour for Class 1

At time 1500 seconds 'class 1' traffic in the optimum path becomes higher than the threshold to activate multipath routing ('MRDV on' in the graph), a second path is activated to route the traffic. This can be seen because the number of paths (red line in the graph) changes from one to two and because part of the traffic goes through the non-priority path (degraded purple in the graph).

At time 2400 seconds, traffic in the interface of the priority path becomes smaller than the threshold to deactivate multipath routing ('MRDV off' in the graph), so the second path is deleted from the routing table. This can be seen because the number of paths (red line in the graph) changes from two to one and because no traffic goes now through the non-priority path (degraded purple area disappears in the graph).

2.2.2.4.2 Test results for 'Class 2'

Traffic belonging to class 2 has less priority than the one from class 1, so the number of paths that must be used to route traffic is decided depending on the joint load for class 1 and class 2.

Figure 12 represents the joint 'Class 1' and 'Class 2' traffic load in interface eth0 of agave4. Again, thresholds to activate multiple routes and to return to a single route are depicted respectively in purple and orange.

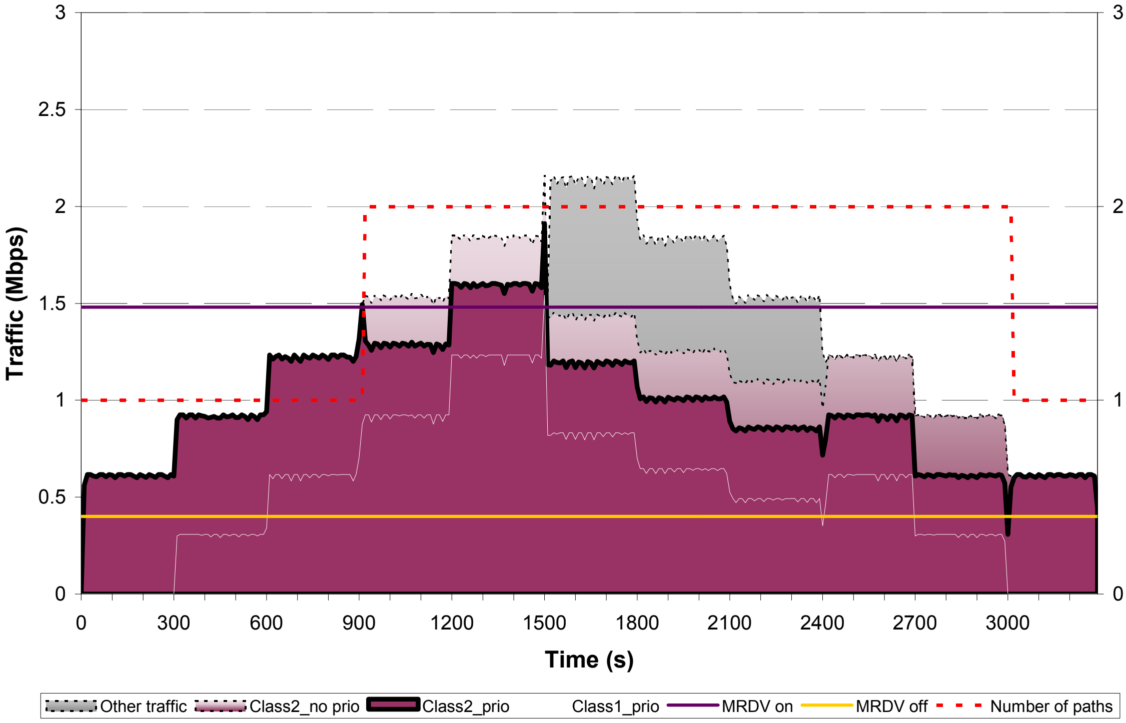


Figure 12 MRDV behaviour for Class 2

At time 900 seconds the sum of the class 1 and class 2 traffics (represented in red colour in the graph) becomes higher than the threshold to activate multipath routing ('MRDV on' in the graph), so a second path is activated to route the class 2 traffic which is represented in a degraded red in the graph and by the red line in the graph that shows the active number of paths for class 2.

As was explained before, at time 1500 seconds class 1 traffic (represented by the white line in the graph) becomes higher than 1.48 Mbps. The traffic belonging to class 1 goes through the secondary path until time is 2400 seconds when the secondary path for class 1 is deactivated. The traffic belonging to class 1 during that time is depicted in grey in the figure since the traffic does not travel through the primary path, and consequently through eth0.

Finally, at time 3000 seconds traffic in the interface of the priority path for the sum of class 1 and class 2 becomes smaller than the threshold to deactivate multipath routing ('MRDV off' in the graph), so the second path is deleted from the routing table. This can be seen because the number of paths (red line in the graph) changes from two to one and because no traffic goes now through the non-priority path (degraded red area disappears in the graph).

2.2.2.4.3 Test results for 'Class 3'

Class 3 has the lowest priority, so to decide the number of paths that must used to route traffic it needs to know the load for all classes.

Figure 13 represents the joint traffic load belonging to classes 1, 2 and 3 in the interface eth0 of agave4. Again, thresholds to activate multiple routes and to return to a single route are depicted respectively in purple and orange.

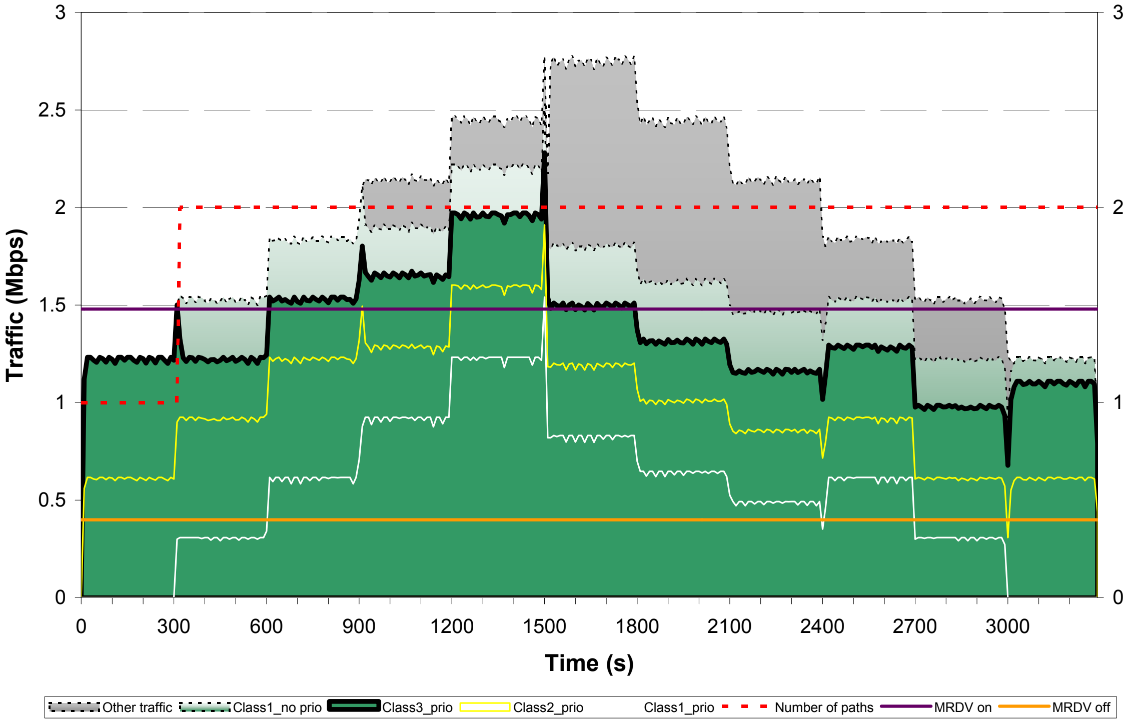


Figure 13 MRDV behaviour for Class 3

At time 300 seconds the sum of all classes (represented in green in the graph) becomes higher than the threshold to activate multipath routing ('MRDV on' in the graph), so a second path is activated to route the class 3 traffic which is represented in a degraded green in the graph and by the red line in the graph that shows the active number of paths for class 3.

As it was seen in the results for 'Class 2', at time 900 seconds the joint traffic from class 1 class 2 (represented by the white line in the graph) becomes higher than 1.48 Mbps. From that time, traffic belonging to class 2 starts going through the secondary path and not through eth0. That traffic load going through the secondary path is illustrated in grey in the figure.

As was seen in the results for 'Class 1', at 1500 seconds Class 1 traffic (yellow line in the graph) becomes bigger than 1.48 Mbps so multipath routing is activated for this class. This secondary path is deactivated at time 2400 seconds for class 1, and at time 3000 seconds for class 2 as was explained previously. This is why the grey area disappears in the graph after this instant.

As the experiment ends before the sum of the traffic for all classes in the priority path becomes smaller than the threshold to deactivate multipath routing ('MRDV off' in the graph), the second path remains in the class 3 routing table, so part of class 3 traffic keeps going through the non-priority path (degraded green in the graph).

2.2.2.5 Validation of the experiment

Follows the results of the simulations performed in order to show the implementation in the Quagga software is correct.

The results are described below:

- Figure 14 shows the time evolution of traffic level for the Class 1 traffic, showing the amount of traffic that is routed through the primary path and the amount of traffic being routed through the secondary path.
- Figure 15 depicts the time evolution for the Class 2 traffic, showing that the secondary path is activated when the Class 1 traffic reaches 900 kbps and therefore Class 1 + Class 2 reaches 1500 kbps, which is greater than the 1480 kbps threshold.
- Figure 16 represents the time evolution for the Class 3 traffic, for which the secondary path is activated when the global traffic level reaches 1480 kbps, at time 300 seconds, when Class 1 traffic level is 300 kbps and the sum is 300 + 600 + 600 = 1500 kbps.

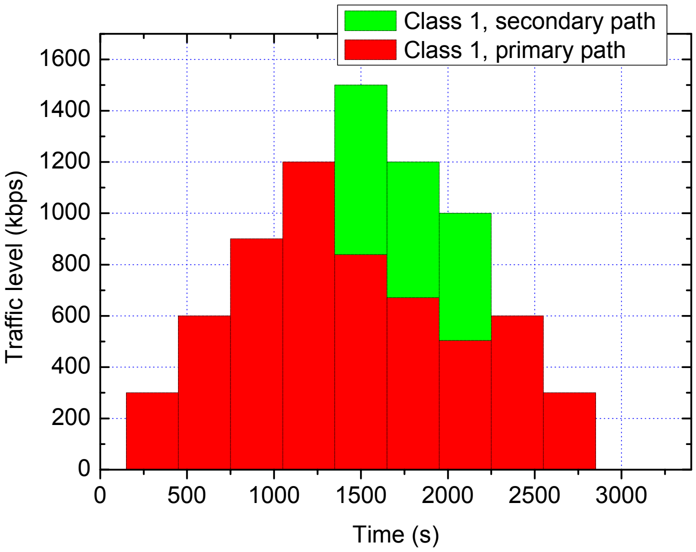


Figure 14 Class 1 traffic evolution

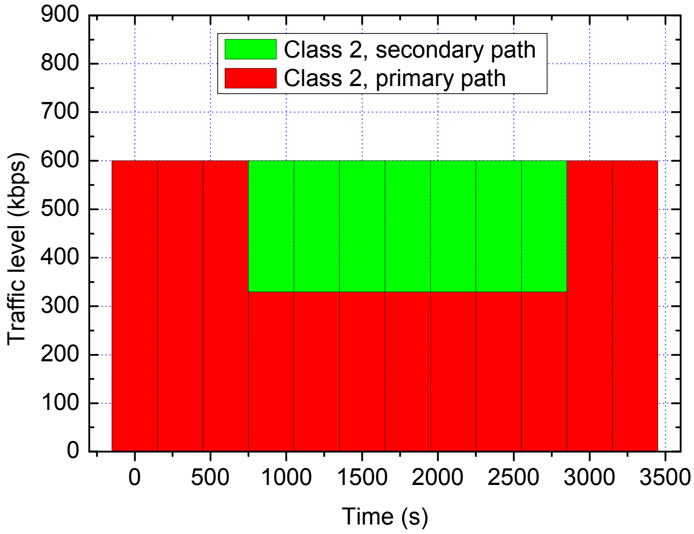


Figure 15 Class 2 traffic evolution

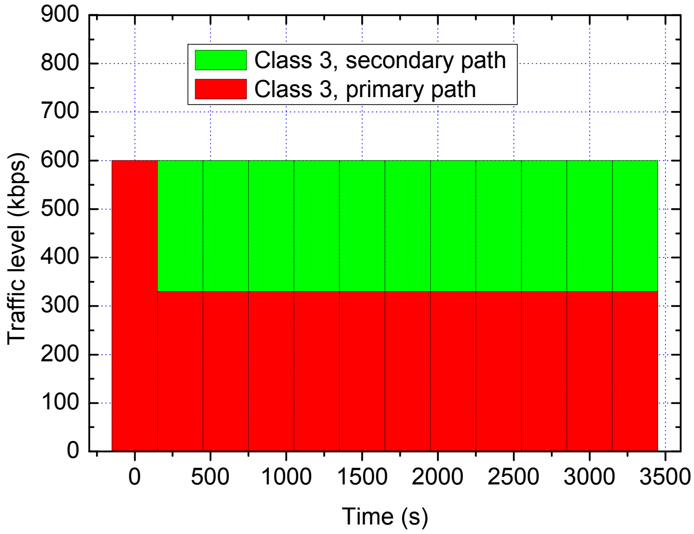


Figure 16 Class 3 traffic evolution

These results show the validation of the implementation of MRDV in the Quagga routers.

2.2.3 Study of MRDV with CoS support and new load distribution module by means of simulation

2.2.3.1 Objectives

In order to validate the new MRDV algorithm supporting Quality of Service, some simulations have been carried out with the NS-2 simulator using the MRDV extensions. The new load distribution module proposed in D4.1 [D4.1] has been slightly modified and implemented in the NS-2 simulator and some results have also been obtained in order to compare all options.

The modified MRDV for QoS support defines that highest priority traffic classes must be carried by the optimum path, if possible. This means that only traffic with lower priority will be initially divided among multiple (not optimum) paths if there is spare capacity for higher priority classes. Highest priority classes will only be divided among multiple paths if they go over the threshold considering the way MRDV works.

The objective of this test run is to show the capabilities of the new MRDV with QoS to differentiate service classes by selectively performing adaptive multipath, as well as the new load distribution module that should improve the results for “Premium class” traffic.

2.2.3.2 New MRDV Load Distribution Module

The new load distribution module initially proposed in D4.1 [D4.1] has been modified so it always sends “Premium class” traffic through the optimum path, i.e., the lowest cost path, while the other classes are routed through the remaining capacity in all available paths.

This means that, if some capacity is left in the optimum path, some lower class traffic will be sent by this path, but the traffic that cannot be accommodated in the optimum path will be sent by the alternative, with higher cost, paths.

The distribution proposed in D4.1 was too complex to be implemented and is not a realistic strategy, as it requires prior information on every class of service traffic load in order to distribute traffic among the available output links. With the alternative proposed in this document, only Premium class, with high delay and jitter requirements, is forced to be transmitted through the optimum path, while the other classes are routed through the spare capacity in all available paths.

2.2.3.3 Scenarios definition for simulations

Two scenarios have been defined and simulated to compare the performance of the different routing strategies, in terms of packet loss, average delay and jitter.

The first scenario is the 28-nodes NOBEL reference network, while the other is a 7-nodes basic core network that has also been deployed for the AGAVE demo. In both scenarios, MRDV is running on all nodes.

Both scenarios have been chosen to test the algorithms and compare their performance in two different situations: a big network that would be the 28-nodes NOBEL network, and a small network that is the 7-nodes core network.

Besides, the big network has been flooded with traffic between all pair of nodes available, to test how the presence of the two traffic classes interact with each other in different paths sharing links. On the other side, the small network has been simulated with a more controlled traffic distribution between two nodes, allowing us to show the way all routing strategies work separately and how the proposed algorithms can lead to an improve in the overall network performance.

Four routing strategies have been compared: ECMP, MRDV without QoS differentiation and MRDV with QoS and the latter with the new load distribution module.

We must point out that the only CoS distinction in the network is done in the MRDV nodes by means of DiffRouting, as no DiffServ queues have been used.

2.2.3.3.1 NOBEL reference network

For the NOBEL reference network, shown in Figure 17, the dimensioning and traffic matrices are taken from the 2005 data for the unprotected case [NOBEL-DIM].

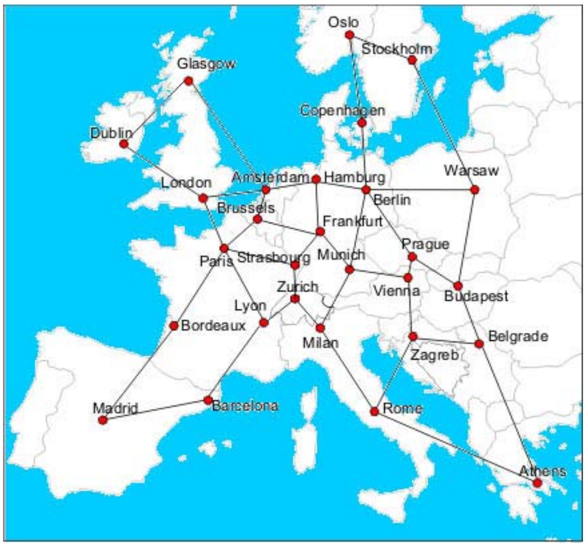


Figure 17 NOBEL network topology used for simulations

The simulations have used UDP traffic sources with 2 service classes. All traffic is CBR traffic sent from all nodes to all other nodes. The traffic matrix is used for the global traffic, divided 10% for the Premium class and 90% for the second priority class.

To compare the different routing strategies in the case of higher traffic conditions, a progressive increase was introduced. Therefore, the results are shown for an initial 0% increase with respect to the default traffic matrix and successive increases of 10, and up to 150% with respect to the initial value. Both classes of service are increased in the same relative amount.

The parameters used for MRDV have been adapted to the topology and the traffic distribution among links with the initial costs in the ECMP case, so there is an appropriate traffic loading of all links in the network. Some nodes will have a higher Maximum Variance value than others, while some others will have a Maximum Variance value of 1, meaning MRDV is "not running" on them.

2.2.3.3.2 Basic 7-node scenario

The network topology used for the simulations is depicted in Figure 18. This topology could correspond to a real network scenario in which there are alternative paths crossing the Atlantic Ocean to interconnect European countries. We have considered very long distances in order to show the differences in average delay and jitter that they introduce in the case of multiple paths to destination.

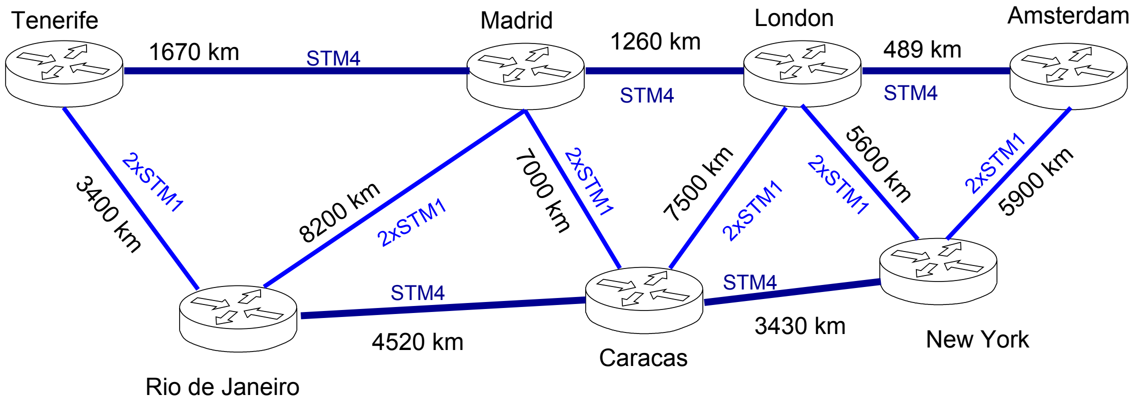


Figure 18 Basic network topology used for simulations

Distances depicted are approximate. Links costs are inversely proportional to their capacity, meaning the STM4 links have half the cost of 2xSTM1 links. Besides, their queues sizes are proportional to the links bandwidth.

The simulations have used UDP traffic sources with 2 service classes. All traffic is CBR traffic sent from Tenerife to Amsterdam and two cases have been studied:

1. Fixed 200 Mbps bandwidth for the high priority class and increasing low priority class from 20 Mbps by a factor varying from 1 to 30.
2. Fixed 200 Mbps bandwidth for the low priority class and increasing high priority class from 20 Mbps by a factor varying from 1 to 30.

In this scenario with traffic going from Tenerife to Amsterdam, ECMP behaves as Shortest Path First, because STM4 links have lower cost than 2xSTM1 links.

MRDV parameters used for the simulations were:

- Maximum variance: 2
- K = 1.5

2.2.3.4 Results for the big network scenario

The results obtained from the simulations previously defined are shown next for both classes of service. The parameters analyzed are the packet loss rate, average delay and jitter for each of the two classes of service.

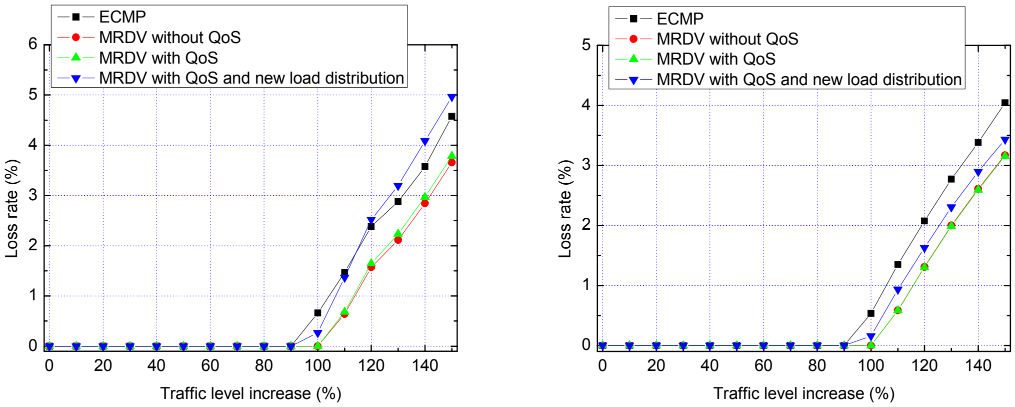


Figure 19 Loss rate for the (left) high priority and (right) low priority traffic classes with increasing low priority class

Figure 19 show the loss rates for both traffic classes for all the routing cases previously described. For the Premium class, MRDV with the new load distribution module behaves a little worse than ECMP, because there is no traffic bypass for the Premium class, while all MRDV implementations behave slightly better than ECMP for the second class.

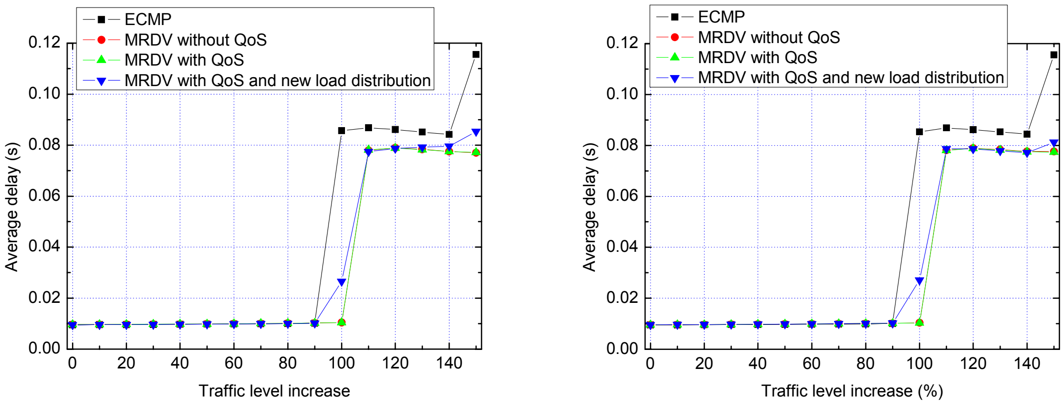


Figure 20 Average delay for the (left) high priority and (right) low priority traffic classes

Figure 20 shows average delay for both traffic classes using the three routing strategies. It is shown that MRDV with CoS distinction performs very similar to traditional MRDV in the context of delay as traffic increases. All MRDV implementations outperform ECMP in delay terms, for both traffic classes, and postpone service degradation.

Last, Figure 21 shows the mean delay variance for each of the traffic classes. It is shown that MRDV with the new load distribution module performs better than ECMP for the Premium class, as no highest priority traffic is divided among multiple paths, while MRDV without QoS treats each traffic class equally and so the highest priority class is divided and sent by longer paths that introduce a higher mean delay variance. It is also shown that all MRDV versions outperform ECMP in terms of delay variance.

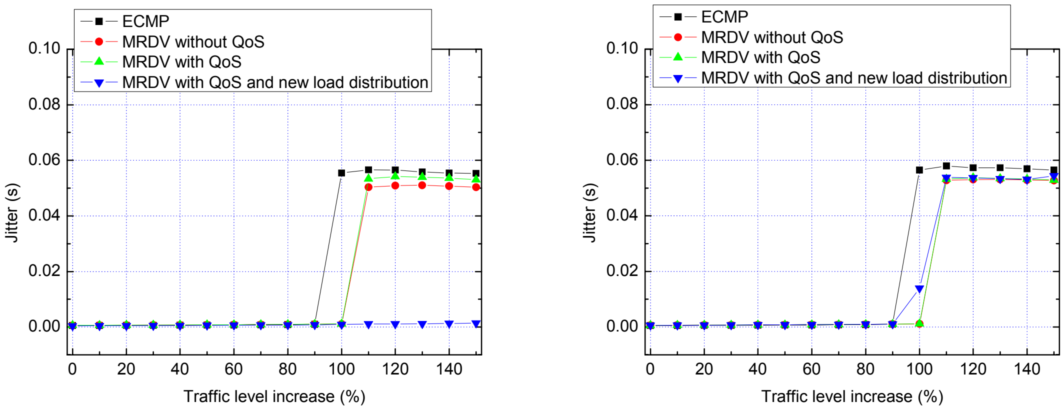


Figure 21 Jitter for the (left) high priority and (right) low priority traffic classes

It must be pointed out, however, that these results can be misleading regarding the performance of the various routing strategies, as traffic is flooded through the entire network with all nodes being source and destination. Because of this, some high priority flows share paths and links with low priority traffic in some parts of the network, and as the nodes queues lack of DiffServ, all classes are treated equally, something that should not happen in a real network supporting QoS.

2.2.3.5 Test results for the small network scenario

As with the big network scenario, the results obtained from the simulations of the small 7-nodes network are shown next for both classes and the two traffic profiles previously defined. The parameters analyzed are, again, the packet loss rate, average delay and delay variance for each of the two classes of service.

2.2.3.5.1 Results for increasing low priority class

In case of increasing the lower priority class traffic, the network is initially not overloaded. STM4 links with 622 Mbps are carrying 220 Mbps in the first traffic level, rising to 620 Mbps when the first packets losses appear. This is shown in Figure 22, where the loss rates for both traffic classes are depicted. For the ECMP case, the loss rate increase is linear as there is no multipath involved in the network. Besides, for the MRDV case, congestion is postponed and the high priority class has better performance in the case of the MRDV algorithm with QoS. This performance is even better in the case of the new load distribution module, as the left figure shows. For the lower priority class, the new load distribution causes a slightly higher loss rate, but still lower than the ECMP case, as some traffic is routed through unloaded paths.

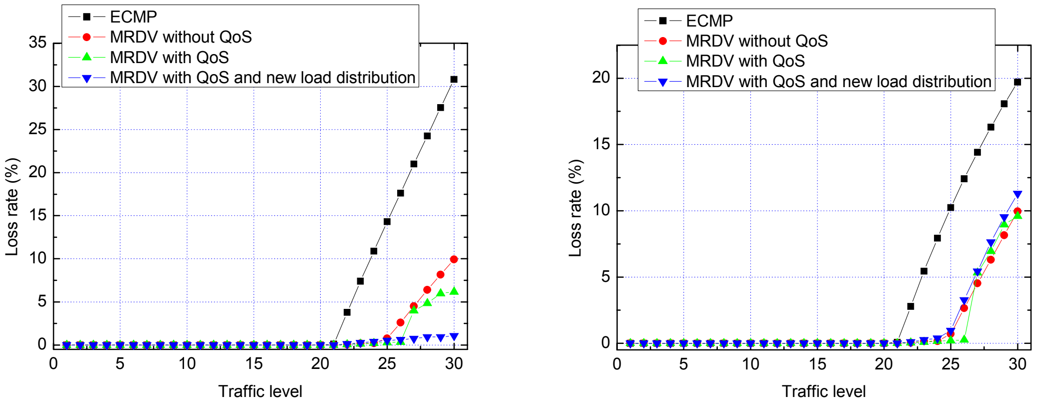


Figure 22 Loss rate for the (left) high priority and (right) low priority traffic classes with increasing low priority class

Figure 23 shows average delay for both traffic classes using the three routing strategies. It is shown that MRDV with CoS distinction outperforms traditional MRDV in the context of delay as traffic increases, because it postpones the highest priority traffic division. The collateral effect of a lower delay for the lowest priority class is due to less overall traffic divided among longer paths.

For the new load distribution module, the delay suffered by the Premium class is even lower than in the ECMP case, as some lower priority traffic is routed through other paths and thus the queues are freed.

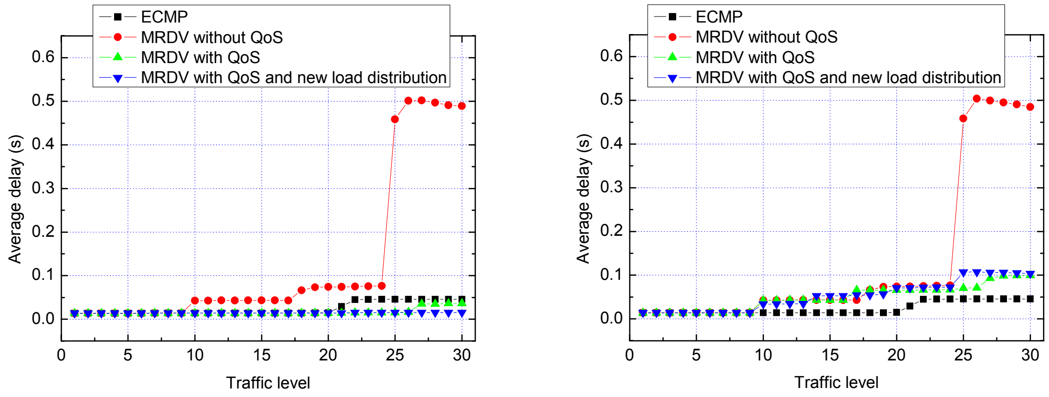


Figure 23 Average delay for the (left) high priority and (right) low priority traffic classes

Last, Figure 24 shows the mean delay variance for each of the traffic classes. It is shown that both MRDV with QoS distinction and MRDV with the new load distribution module perform like ECMP, as no traffic is divided among multiple paths, while MRDV without QoS treats each traffic class equally and so the highest priority class is divided and sent by longer paths that introduce a higher mean delay variance. For the low priority traffic, both MRDV with QoS and MRDV with the new load distribution module perform worse than MRDV without QoS, as expected.

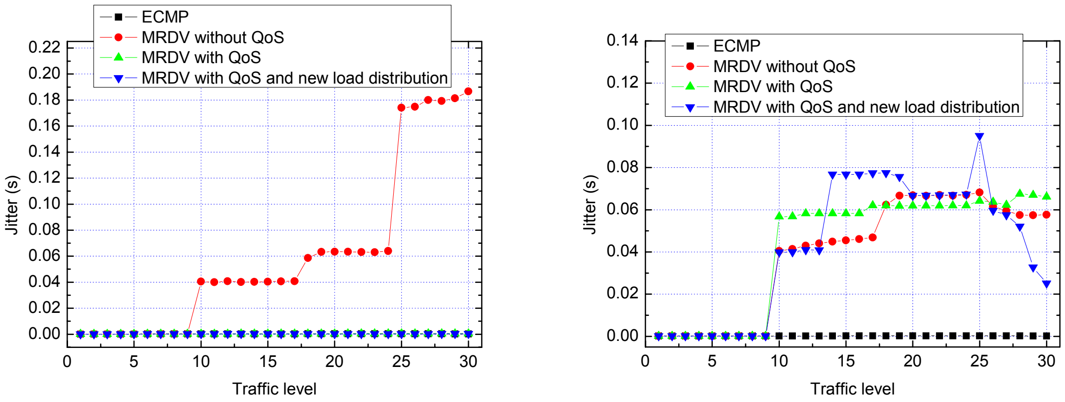


Figure 24 Jitter for the (left) high priority and (right) low priority traffic classes

2.2.3.5.2 Results for increasing high priority class

Follow the results of the simulations for the case of increasing the higher priority class traffic. As in the previous case, the network is not initially overloaded and when global traffic load reaches 620 Mbps, the first packets get lost, as shown in Figure 25. It is shown that the case of using MRDV with the new load distribution module gets worse results in terms of packet losses because Premium class traffic saturates the main path and cannot be split.

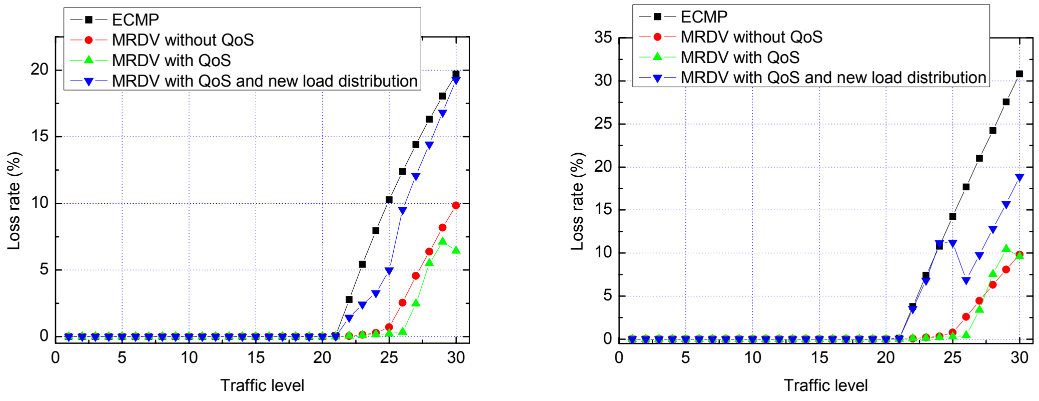


Figure 25 Loss rate for the (left) high priority and (right) low priority traffic classes with increasing high priority class

Figure 26 represents the average delay for both traffic classes. For both classes, the increase in average delay when MRDV with QoS differentiation is used is postponed with respect to the case of MRDV without QoS. The case of MRDV with the new load distribution scheme postpones delay increase in both classes of service. However, delay for the high priority traffic cannot be avoided as it reaches link congestion level and traffic must be divided among several links all over the network in the case of MRDV with QoS and the queues are overloaded in the case of MRDV with the new load distribution.

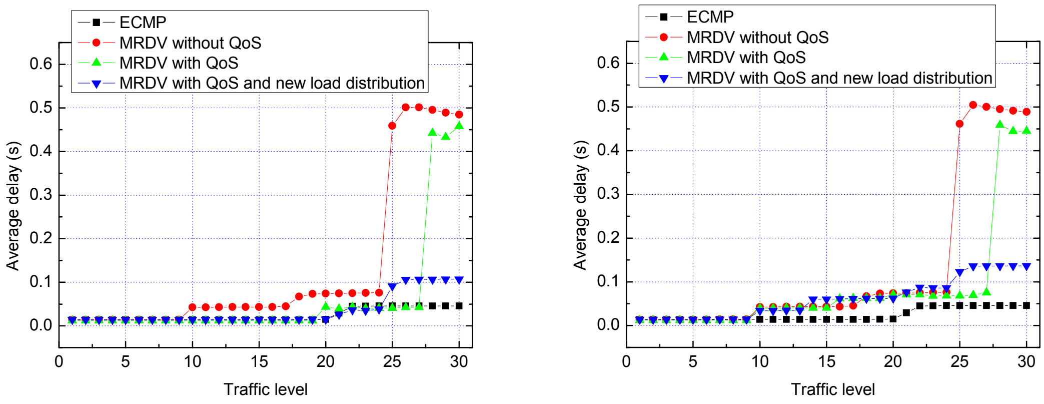


Figure 26 Average delay for the (left) high priority and (right) low priority traffic classes

Last, Figure 27 depicts jitter for both classes. It is shown that higher priority traffic suffers less from mean delay variance than lower priority traffic, both in MRDV with and without QoS distinction. However, as the mean delay for higher priority traffic increases when congestion appears, some variance is introduced, although the traffic level at which it starts to rise in MRDV with QoS is double the level it appears for the case of MRDV without QoS.

The new load distribution module outperforms the previous MRDV cases, showing no jitter at all traffic levels for the Premium class.

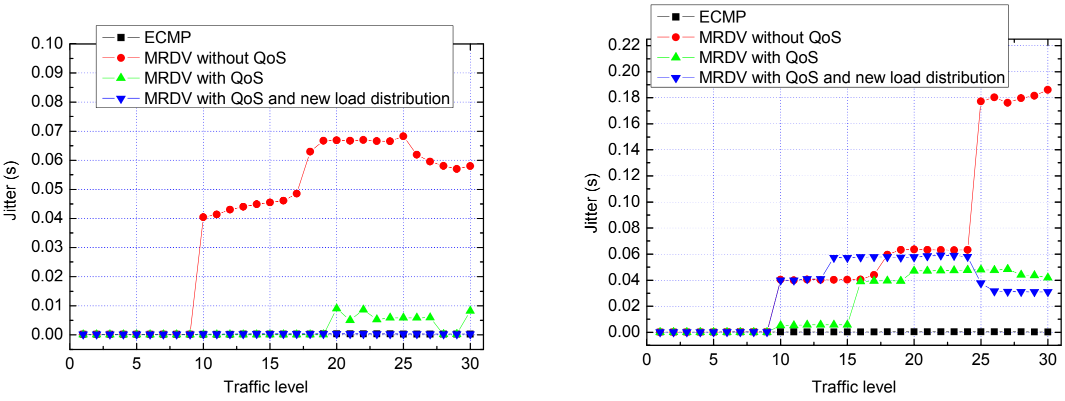


Figure 27 Jitter for the (left) high priority and (right) low priority traffic classes

2.2.4 Performance tests of MRDV with CoS support

2.2.4.1 Introduction

The implementation of MRDV with CoS support has been tested in a more complex scenario than the one presented in section 2.2.2.4. The goal of these new tests is the demonstration through performance measurements (delay and loss rate) that the Network Plane differentiation provided by MRDV with QoS is able to support high load from Best Effort traffic while guaranteeing performance for Premium traffic.

2.2.4.2 Scenario

Next figure shows the scenario used to perform these tests.

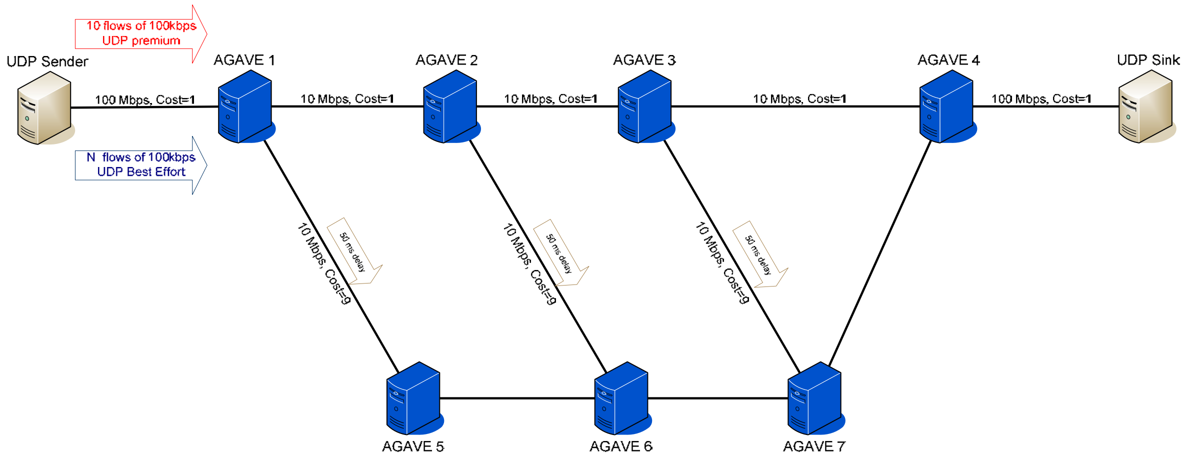


Figure 28 Test scenario

The scenario is composed by seven nodes (the ones depicted in blue) that are Linux routers, and two end nodes (the ones depicted in white) that are the traffic sender and traffic sink. The Linux routers in the top line (AGAVE1, AGAVE2, AGAVE3 and AGAVE4) are running MRDV with QoS, while the Linux routers in the bottom line (AGAVE5, AGAVE6 and AGAVE7) are running just OSPF. Links connecting the different routers have a bandwidth of 10 Mbps. The links connecting the top line with the bottom one have a propagation delay of 50 milliseconds.

Link costs in the scenario have been established in such a way that the optimum path between the sender and the sink goes through the top line of routers. If the load in the interface of the optimum path becomes high, MRDV nodes (the ones in the top line) will distribute the load between its interfaces so that the bottom line will be used and the packets, although delayed 50 milliseconds, will arrive to its destination.

Two classes of service (Premium and Best Effort) have been configured in the routers, so that the Premium traffic is prioritized in the routers queues. The MRDV nodes (the ones in the top line) will work with two Network Planes, one for each class of service, and their decisions about load distribution will be taken as specified in section 2.2.2.3:

- The decisions about distributing Premium traffic will be taken just by considering the Premium traffic load.
- The decisions about distributing Best Effort traffic will be taken by considering both the Premium traffic load and the Best Effort traffic load.

2.2.4.3 Test definition

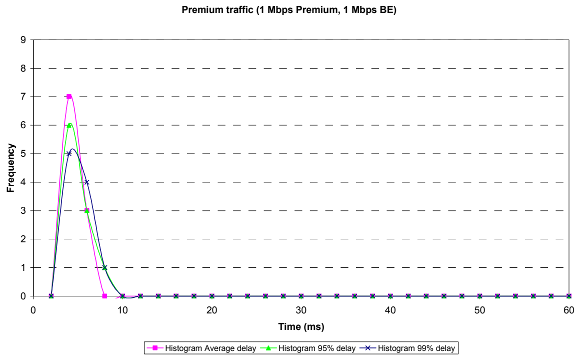
Different tests have been performed with this scenario. In all these tests, two types of traffic flows were injected from “UDP Sender” to “UDP Sink”:

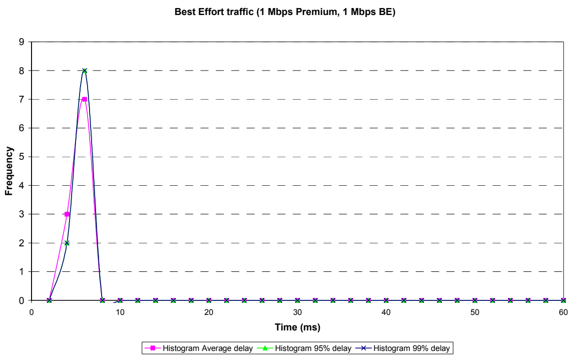
- Premium traffic: UDP traffic flows with constant bit rate, whose packets are marked with TOS 1, so that MRDV nodes can identify them as Premium traffic. In all tests, 10 flows of 100 kbps were sent, so that the total traffic load of Premium traffic is 1 Mbps, lower than the threshold in MRDV nodes to perform load distribution. In these conditions, Premium traffic will always flow through the top line.
- Best Effort traffic: UDP traffic flows with constant bit rate, whose packets are marked with TOS 0, so that MRDV nodes can identify them as Best Effort traffic. In each test, a different number of flows of 100 kbps were sent. Four tests were performed: 10, 40, 70 and 100 flows, so that the Best Effort traffic load is 1, 4, 7 and 10 Mbps respectively. For 10 and 40 flows, the sum of Premium and Best Effort traffic loads will be lower than the threshold in MRDV nodes to perform load distribution. However, for 70 and 100 flows, the sum of Premium and Best Effort traffic loads will be higher than the threshold in MRDV nodes to perform load distribution, so that some Best Effort traffic flows will travel through the bottom line.

All flows start sending traffic at the same time (excepting some randomness in the specific starting time) and keep active during 600 seconds.

2.2.4.4 Results

Next figures show the histograms of packet delay for Premium traffic flows and Best Effort traffic flows in the scenarios with 1 and 4 Mbps of Best Effort traffic.





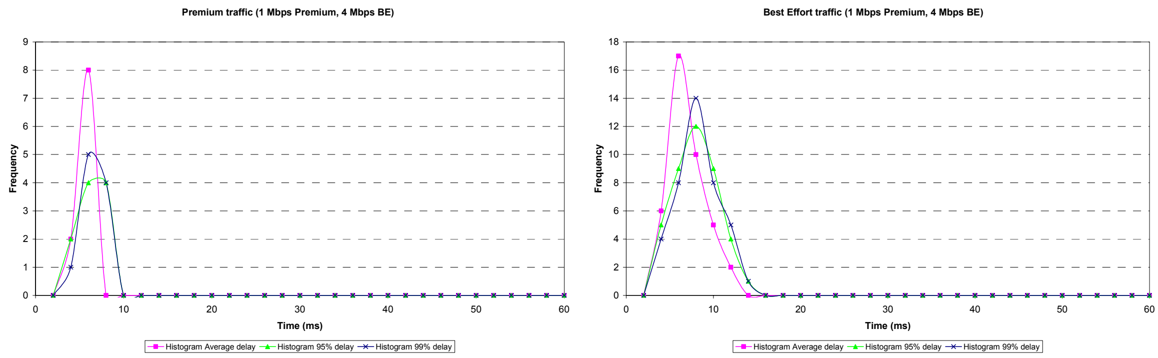


Figure 29 Histograms of packet delay for Premium traffic and Best Effort traffic flows in the scenarios with 1 and 4 Mbps of Best Effort traffic

As expected, the packet delays of both Premium traffic and Best Effort traffic flows are low. The delay in Best Effort packets stays low since the load is not high enough to make the Best Effort traffic slower. Nevertheless, certain impact can be seen in the histogram of packet delay for the 4Mbps Best Effort flows, where the histogram is wider than for the 1 Mbps Best Effort flows.

Next figures show the histograms of packet delay for Premium traffic flows and Best Effort traffic flows in the scenarios with 7 and 10 Mbps of Best Effort traffic.

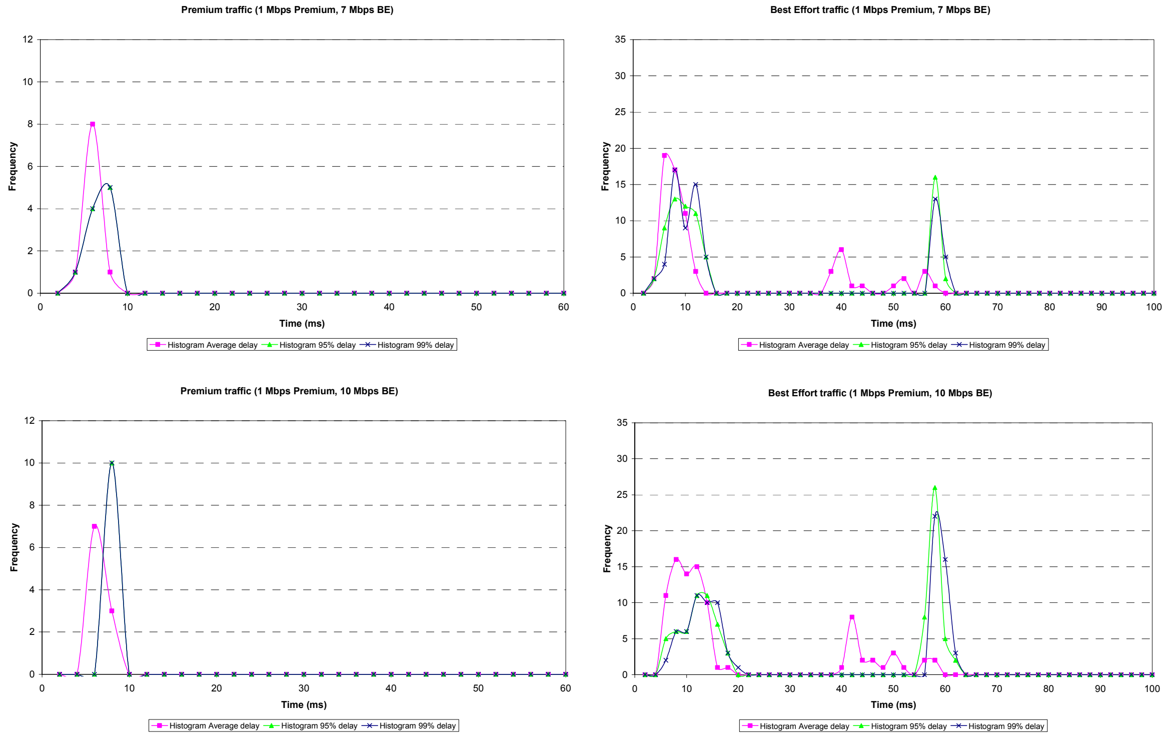


Figure 30 Histograms of packet delay for Premium traffic and Best Effort traffic flows in the scenarios with 7 and 10 Mbps of Best Effort traffic

As expected, the packet delays of Premium traffic flows stay low since packets of Premium traffic are prioritized over Best Effort ones. On the other hand, the packet delay in Best Effort traffic flows grows and the histograms become bi-modal since, in these load conditions, some of the Best Effort flows travel through the bottom line (and in its way, they get a propagation delay of 50 milliseconds). In the figure, some Best Effort flows have an average delay of 30-40 ms., which is explained by the fact that some flows use both the optimum path through the top line and the secondary path through the bottom line.

It must be noticed that the network has been able to deal with a total load of 11 Mbps (1 Mbps of Premium traffic and 10 Mbps of Best Effort traffic), higher than the admissible load in the optimum path (10 Mbps). Besides, although a propagation delay has been added when the flows have traveled through the secondary path, the packet delay becomes controlled since the Best Effort queues in the routers have not become full.

In order to illustrate the advantages of this approach over a network administrated with OSPF and with service class differentiation, the scenario of 1 Mbps Premium traffic and 10 Mbps Best Effort traffic has been run again, but with the MRDV nodes (the ones in the top line: AGAVE1, AGAVE2, AGAVE3 and AGAVE4) running OSPF this time.

Next figures show the histograms of packet delay for Premium traffic flows and Best Effort traffic flows in this scenario with OSPF and service class differentiation.

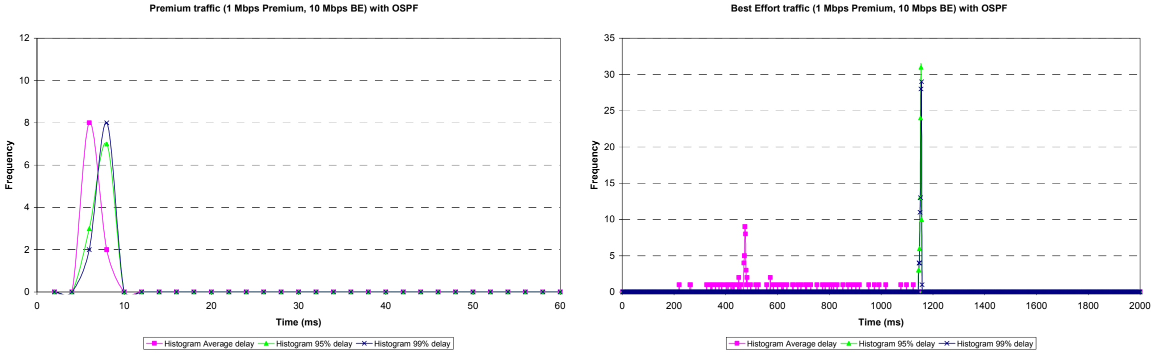


Figure 31 Histograms of packet delay for Premium traffic and Best Effort traffic flows in the scenario with 1 Mbps Premium traffic and 10 Mbps Best Effort traffic with OSPF and service class differentiation

As expected, the packet delays of Premium traffic flows stay low since packets of Premium traffic are prioritized over Best Effort ones. However, the packet delays in Best Effort traffic flows grow and become really high since the Best Effort queues in the routers in the optimum path become full.

Besides, next figure shows the packet loss rate distribution function for Premium traffic flows and Best Effort traffic flows in both scenarios with MRDV with QoS and OSPF with service class differentiation.

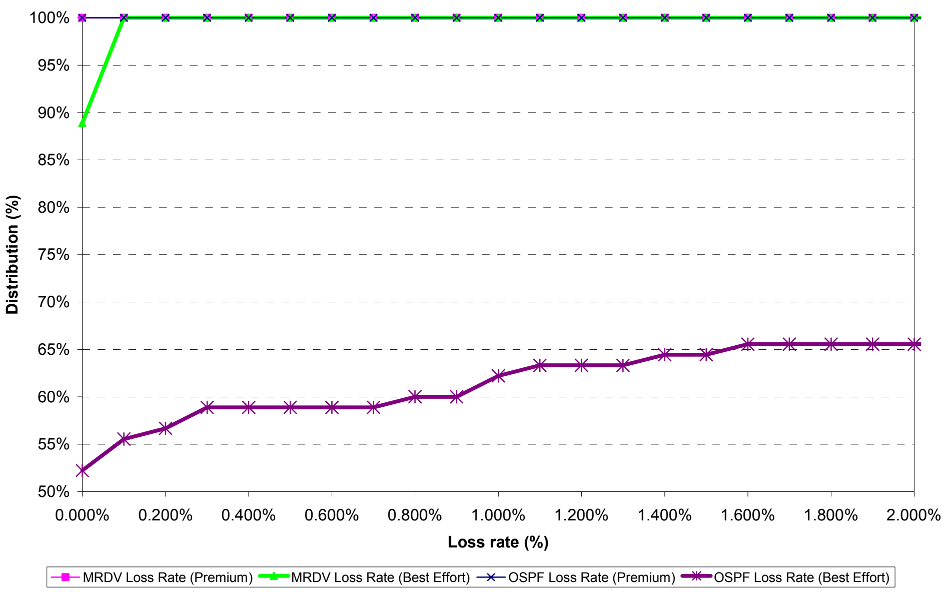


Figure 32 Distribution function of packet loss rate for Premium traffic and Best Effort traffic flows in the scenario with 1 Mbps Premium traffic and 10 Mbps Best Effort traffic with either MRDV with QoS or OSPF with service class differentiation

Packet loss rates are 0% for all Premium traffic flows in both scenarios, which is normal since the Premium packets are prioritized over the Best Effort ones. Regarding the Best Effort flows, all the flows in the MRDV scenario have a packet loss rate lower than 0.1%, whereas only a 56% of flows in the OSPF scenario have a packet loss rate lower than 0.1%. Besides, in the OSPF scenario, only a 66% of Best Effort flows have a packet loss rate lower than 2%.

These results demonstrate that the Network Plane differentiation provided by MRDV with QoS is able to support high load from Best Effort traffic while guaranteeing performance for Premium traffic.

2.2.5 Conclusions

The extension of MRDV to support multiple traffic classes was presented in D3.1 [D3.1], allowing the original MRDV algorithm to generate different Network Planes. For the work in this document, several studies have been made.

First, the MRDV with QoS support implementation in Quagga software has been presented, including some tests in a small scenario, validated by simulation, showing how different classes of service are treated from the routing point of view.

Then, we have shown the simulation results arising from the application of this MRDV with QoS support in bigger scenarios, one with a more controlled traffic profile and the other one flooding the entire network with traffic. The results demonstrate the capability of the new algorithm to differentiate traffic and prioritize one class over others in order to allow for a better performance of the higher priority classes.

Besides, the new load distribution module proposed in D4.1 [D4.1] has been modified and implemented in the NS-2 MRDV simulator in order to compare its performance against the other proposals. It has been shown in the results that it postpones degradation in Premium class traffic performance for the simulated case. Although the high priority class traffic loss rate is worse when this class' traffic increases, we must point out that Premium class traffic is not dominant in real networks.

Besides, from the results obtained in the big network scenario, it would be advisable to include DiffServ queues in the network in order to prioritize traffic further than the mere routing strategies.

Finally, some performance tests of MRDV with QoS support were done. These tests show through performance measurements (delay and loss rate) that the Network Plane differentiation provided by MRDV with QoS is able to support high load from Best Effort traffic while guaranteeing performance for Premium traffic.

2.3 INP-layer Overlay Routing

2.3.1 Objectives

The objective is to evaluate the overall network performance the proposed tunnelling based overlay routing that aims to provide IP based fast reroute with post-failure traffic optimization. Comparison will be done between our proposed schemes with existing TE approaches.

2.3.2 Experimental Setup

The experimental setup is the same as described in Section 2.1.2.

2.3.3 Test Results

2.3.3.1 Performance metrics

The following performance metrics are considered:

1) Fast Reroute Coverage: It indicates the coverage of failure protection by the proposed tunnel-based IP FRR mechanism. We use the following metrics:

- **(FC-1) The percentage of links that can be fully protected for all destinations**: a link is said to be fully protected for all destinations only if every affected destination has at least one feasible tunnel endpoint.
- **(FC-2) The percentage of destinations which can be fully protected for all single link failure scenarios:** a destination is said to be protected for all link failures if there exist at least one feasible tunnel endpoint for every link failure that affects the destination.
- **(FC-3) For all destinations and for all link failures, the percentage of the total potential failure cases which are protected.** This examines the overall “best effort” protection.

2) Post-Failure Maximum Link Utilization: For simplicity we assume each network link has equal chance to fail, but with no simultaneous failures of multiple links. We then consider the worst-case (i.e. highest) post-failure MLU among all the single link failure scenarios. Note that if a link cannot be fully protected for all destinations, we assume IP re-convergence will handle this type of link failure instead. Therefore, the worst-case post-failure MLU could be the result of our tunnel-based IP FRR mechanism or IGP re-convergence.

2.3.3.2 Algorithm comparison

We consider the following approaches in our evaluation of post-failure network utilization:

- **IGP-RCVG:** This approach relies on IGP re-convergence to recover routing failures. It is thus a basic and reactive approach that handles link failures without considering IP FRR.
- **FRR-G:** This approach adopts the tunnelling mechanism for IP FRR in conjunction with our proposed tunnel endpoint selection algorithm. It is thus the approach that considers both fast routing failure recovery and post-failure load balancing.
- **FRR-R:** This approach is similar to the FRR-G except that the tunnel endpoint selection is purely done *randomly*. It can be regarded as the approach that aims to achieve fast recovery from routing failure only, without considering post-failure load balancing. Note that, as random selection may produce results with different worst-case MLU, we take an average of 10 independent trials.

2.3.3.3 Failure coverage performance

For the TE-optimized link weight scenario, we further tested three traffic matrices with different levels of traffic intensity under normal network conditions (i.e. low, medium and high). Therefore, the link weights used for the three traffic scenarios may be different.

Link weight	**FC-1**	**FC-2**	**FC-3**
Actual	94.6%	69.57%	98.61%
Uniform	100%	100%	100%
InvCap	97.3%	60.87%	98.22%
TE-Optimized (low)	98.65%	95.65%	99.8%
TE-Optimized (med)	98.65%	39.13%	97.23%
TE-Optimized (high)	100%	100%	100%

Table 4 Fast reroute coverage – GEANT

Link weight	***FC-1***	***FC-2***	***FC-3***
Actual	78.57%	27.27%	84.55%
Uniform	100%	100%	100%
InvCap	100%	100%	100%
TE-Optimized (low)	92.86%	63.64%	96.36%
TE-Optimized (med)	92.86%	72.73%	97.27%
TE-Optimized (high)	89.29%	54.55%	94.55%

Table 5 Fast reroute coverage – Abilene

From the results in for the GEANT network, we see that IGP link weight configuration plays an important role in influencing the degree of FRR coverage. This suggests that *the FRR coverage could be improved by optimizing IGP link weights*. As far as *FC-1* is concerned, our tunnel-based IP FRR mechanism can protect all affected destinations from most of the link failures. The percentage of destinations that can be protected by all link failures (*FC-2*) as well as *FC-3* is also high in general. These results show that the proposed tunnel-based IP FRR mechanism is effective.

For the TE-optimized link weight scenario, although the same algorithm is used, the results based on different traffic matrices can lead to different degree of FRR coverage. This implies that *there may be a trade-off between traffic performance and effectiveness of IP FRR in which achieving near-optimal network performance comes at the expense of low degree of failure protection by IP FRR*.

For the Abilene network, we found that the results are similar to those of the GEANT network except that the degree of FRR coverage is in general lower, mainly due to small average node degree of the topology which reduces the number of feasible tunnel endpoints.

We also observed an interesting result from both tables that homogeneous link weight can always achieve full FRR coverage in general. This suggests that *there exist at least a set of link weights that can always achieve full FRR coverage by using the proposed tunnelling mechanism*. An analytical proof for this is given in the Appendix.

For the results in Table 4 and Table 5, we see that full FRR coverage cannot be achieved under some network configuration scenarios. We believe that there are two primary reasons that can influence the FRR coverage: IGP link weights and topology connectivity. For IGP link weights, we have already demonstrated its effects in Table 4 and Table 5. On the topology connectivity side, a straightforward approach is to add more links in order to enrich the overall network connectivity. A natural hypothesis is that adding links to the network increases the chance in finding feasible tunnel endpoints, thereby improving the FRR coverage.

By using BRITE [BRITE], we generate a large-scale network topology with 50 routers and 200 unidirectional links. For the purpose of demonstration, we assume that link capacity is randomly generated and InvCap link weight setting is used. Links are added to the base network following the Waxman's model, i.e. links are added between routers that are closest to each other if there were not any link exist.

Network Topology	**FC-1**	**FC-2**	**FC-3**	**Worst-case MLU**
Base	99.5%	98%	99.96%	172%
+ 6 links	99.03%⬇	86%⬇	99.71%⬇	134% ⬆
+ 12 links	99.06%⬇	80%⬇	99.59%⬇	93% ⬆
+ 16 links	99.07%⬇	84%⬇	99.67%⬇	93% ▬

Table 6 Network topology expansion

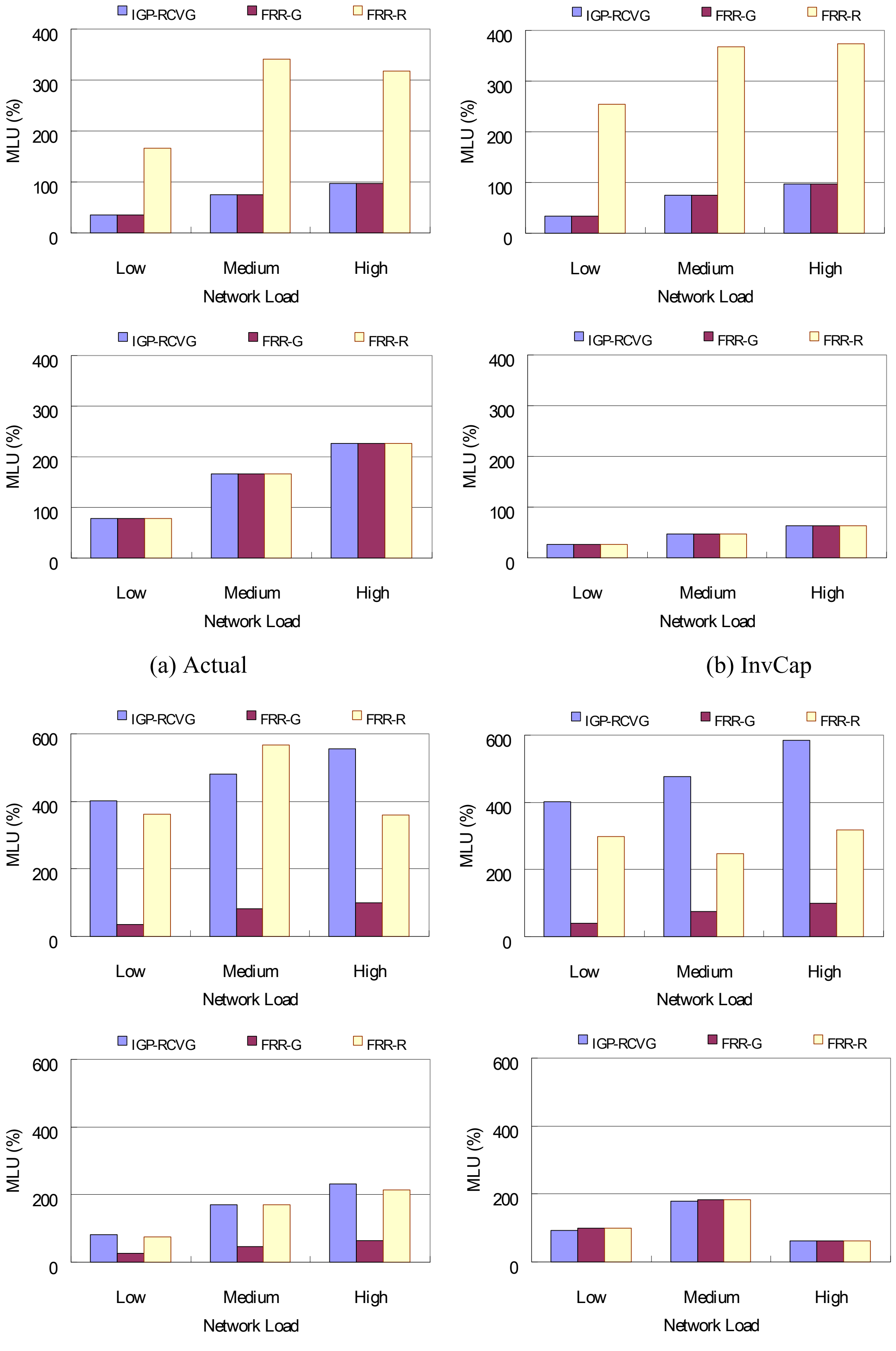
Table 6 shows the FC-1, FC-2, FC-3 and worst-case MLU with different number of links added onto the base network topology. An interesting result is that when adding 6, 12 or 16 links to the network, all the values of FC-1, FC-2 and FC-3 decrease compared to those for the base topology. This reveals that *adding more links to the network does not guarantee the improvement of FRR failure protection coverage. Instead, this could make the performance even worse*. These results refute the original hypothesis. Nevertheless, the worst-case MLU is improved, as more capacity is available in the network.

2.3.3.4 Load balancing performance

Having evaluated the FRR coverage achieved by the proposed tunnel-based IP FRR mechanism, we proceed to investigate how different approaches in Section 2.3.3.2 perform in terms of post-failure network utilization. Figure 33 shows the worst-case MLU among all the link failures.

An overall picture of Fig. 6 indicates that the FRR-G approach performs much better than the IGP-RCVG approach. This means that using IP FRR via tunnelling together with judicious tunnel endpoint selection can achieve significant performance improvement over the traditional IGP re-convergence. The gain includes not only very high FRR coverage for failure protection but also minimized possibility of experiencing post-failure network congestion. On the other hand, imprudent tunnel endpoint selection can cause severe congestion after the affected traffic is diverted onto the randomly selected tunnel endpoint. As a result, packets may still be discarded even though routing failures can be recovered rapidly, thereby not able to guarantee comprehensive QoS assurance.

However, under some scenarios in Figure 33, the FRR-G approach has the same performance to the IGP-RCVG or even the FRR-R approach. This is partly due to the reason that, under failure of some links, there does not exist any feasible tunnel endpoint for some destinations. As a result, FRR is not used when *any* of these link failures occurs and the traditional IGP re-convergence takes place instead. However, we observed that one of these link failures has caused the highest utilization. This explains why the worst-case MLU of both approaches is the same as they both account for the same highest utilization based on the traditional IP re-convergence.



(a) Actual

(b) InvCap

(c) Uniform

(d) TE-Optimized

Figure 33 Post-failure network utilization under various IGP link weight (top 4: GEANT, bottom 4: Abilene)

Another interesting result is shown in Figure 33 Abilene network. For the TE-optimized link weight, the worst-case MLU based on the high-loaded traffic matrix is surprisingly lower than those based on the low- and medium-loaded ones. In general, the higher the traffic load is, the highest the link utilization. However, this abnormal phenomenon implies that the link weight that is well optimized only for the normal network condition could perform poorly under link failures with IP FRR.

Based on the evaluation results in Table 4, Table 5 and Figure 33, we conclude with the following observations. (i) The proposed tunnel-based IP FRR mechanism with judicious tunnel endpoint selection can achieve both fast routing failure recovery and post-failure load balancing; (ii) Imprudent tunnel endpoint selection can easily cause severe congestion after failures and therefore reduces the effectiveness of IP FRR. Therefore, it is not advisable to deploy IP FRR only without considering how to control it to achieve post-failure load balancing; (iii) Link weights that are optimized for conventional traffic engineering under failure-free conditions may lead to lower FRR coverage and poor post-failure network utilization after affected traffic is re-routed onto the repair paths.

2.3.4 Conclusions

Our evaluation results based on real operational networks reveal that the tunnelling mechanism with judicious selection of tunnel endpoint can achieve high fast reroute coverage and improve post-failure load balancing. We also found that IGP link weight plays an important role in influencing the overall failure coverage, which facilitates network operators to intelligently configure the IP routing logic (e.g. setting appropriate IGP link weights) to achieve maximum fast reroute coverage.

2.4 NP Emulation Platform

2.4.1 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 and exhibiting the concepts and notions developed by the project. Due to its emulated nature, the platform can be used for running ‘what-if’ scenarios for alternative business and NP/PI set up cases on different network topologies and traffic generation patterns.

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.
- 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.

2.4.2 Experimental Setup

NPEP is a computer-based system that runs at the local test-bed provided by Algonet. It consists of a high-end server hosting a database and a Web server. Clients with standard browsers can access the server to operate NPEP.

In addition to access from the public Internet, access to the NPEP system is provided remotely from the AGAVE integrated test-bed.

Figure 34 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.

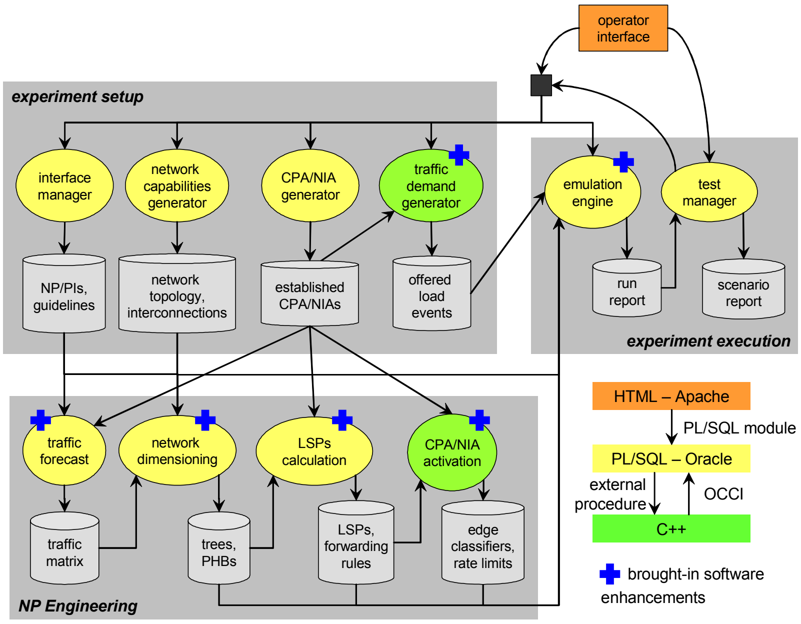


Figure 34 Overview of NP Emulation Platform

2.4.3 Test Results

NPEP experimentation focused on validation aspects. It was verified that the operations/functionality offered by NPEP operate correctly, producing the anticipated results. Namely with NPEP one can:

- Define the exact information model of the entities of concern: NIAs, NSs, NPs and PIs. Note that NPEP uses the respective AGAVE specifications as meta-models. We consider that this approach enhances the applicability of NPEP and its porting to different INP domains. Note that the attributes -as meaning, syntax and default/possible values- representing the offered services and the dimensions along which services can be provisioned may change from INP to INP. A universal model, able to capture all possible cases, is hard to specify.
- Enter/define, view, update and delete NIAs and NSs and collections of them capsulated in the context of a so-called Business Environment.
- Enter/define, view, update and delete multiple sets of PIs for a particular collection of NSs and NIAs (Business Environment). The set of PIs, which corresponds on a one-to-one basis to the NSs in mind, can be generated automatically.
- Enter/define, view, update and delete multiple sets of NPs based on which the PIs of a particular set will be instantiated.
- Define, view, update and delete associations between a selected set of NPs and the NIAs of the Business Environment under consideration for instantiating the PIs of a particular set.
- Enter/define, view, update and delete multiple physical network topologies.
- Define an experiment by selecting a Business Environment that is, collections of NSs and NIAs, and one of the virtual configurations, sets of PIs and NPs, defined for this environment.
- Drive the underlying Traffic Engineering system to produce a valid network configuration corresponding to a particular set of PIs and NPs and see the results. The current release assumes the TEQUILA TE system for Diffserv/MPLS IP networks; the scheduling characteristics of the appropriate PHBs and the LSPs defined by the TE system for provisioning the defined NPs are seen.
- Run an experiment and see the results; goodput, link and PHB usage per NP, totals or over time.

2.4.4 Conclusions

Overall, NPEP shows the validity of the AGAVE specifications in that they (a) can be workable and (b) can lead to working network configurations.

3 INTER-DOMAIN ROUTING EXPERIMENTS

3.1 Joint (intra and inter) Robust TE and Interactions

3.1.1 Objectives

The objective is to evaluate the overall network performance of the proposed joint robust TE algorithm. This algorithm proposes an IGP link weight optimization that aims at achieving intra- and inter-AS load balancing under both normal network condition and any single intra- or inter-AS link failure conditions while taking their interactions into account. We also compare our proposed approach with four IGP link weight optimization approaches in which two of them did not consider any link failure while the other two considered only intra-AS link failures. Nevertheless, all of them neglected the impact of both intra- and inter-AS link failures on the overall performance.

3.1.2 Experimental Setup

3.1.2.1 Network Topology and Destination Prefixes

Our experiments were performed on two Point-of-Presence (PoP) level topologies generated by BRITE [BRITE]. The two PoP level topologies have 50 nodes with 100 links and 100 nodes with 200 links. In each topology, all PoP nodes are ingress points while only some of them, namely border PoPs, are connected to adjacent provider ASes through inter-AS links and hence they can be both ingress and egress points. A similar network setup is also found in some ISP PoP topologies provided by Rocketfuel [SPRIN04]. We notice that the number of border PoPs in these topologies is about half of the total PoP nodes. Therefore, without loss of generality, we randomly select half of the PoP nodes as border PoP nodes each with only one inter-AS link. We also assume a homogenous environment in which the capacity of all the intra- and inter-AS links are OC-192 (9.6 Gbps) and OC-48 (2.5 Gbps) respectively.

For scalability and stability reasons, the joint robust TE can focus only on a small fraction of Internet destination prefixes, which are responsible for a large fraction of the Internet traffic [FEAM03]. In line with [BRES03] and [UHLI05], we consider 1000 popular destination prefixes. In fact, each of them may not merely represent an individual prefix but also an aggregate of multiple destination prefixes that have the same set of candidate egress points. This simplifies the problem by significantly reducing the number of prefixes to be considered. Nevertheless, the number of prefixes we consider could actually represent an even larger value of actual prefixes.

We assume that each border PoP has reachability to all the considered destination prefixes. Therefore, during NS, the inter-AS traffic received at a border PoP towards any destination prefix will exit the network through the same border PoP without traversing the network. However, if the inter-AS link attached to this border PoP fails, the inter-AS traffic will have to be routed within the network and then exit from another border PoP.

3.1.2.2 Traffic Matrices

We generate synthetic traffic matrices for our experiments. According to [BROI04], inter-AS traffic volumes are top-heavy and follow the Weibull distribution with shape parameter 0.2-0.3. We therefore generate the inter-AS TM with this distribution using the shape parameter of 0.3. In addition, following the methodologies in [NUCC07], we generate local intra-AS TM using the Gravity Model (GM). In this model, the amount of incoming traffic at a PoP is proportional to its size. Following the suggestions in [BHAT01], we randomly classify 40% of PoPs as “small”, 40% as “medium” and 20% as “big”.

3.1.2.3 Weighting Parameter

From AGAVE Deliverable D3.2, recall that the optimization problem of the joint robust TE can be formulated as follows:

$$\underset{W}{Minimize}(U^{intra}_{max\_NS}, U^{intra}_{worst\_AllFSs}) = \underset{W}{Minimize}((1-\alpha)U^{intra}_{max\_NS} + \alpha U^{intra}_{worst\_AllFSs}) \tag{3.1}$$

where $0 \leq \alpha \leq 1$, subject to the inter-AS utilization constraint:

$$U^{inter}_{worst\_AllStates} \leq \varepsilon \tag{3.2}$$

where $0 < \varepsilon \leq 1$.

By varying the weight parameters α in objective function (3.1) and re-solving it, one can generate a trade-off curve between the two objectives of each function using the method of multi-objective programming [CHAN83]. If we solve the problem with $\alpha=0$, the problem is simply reduced to the intra-AS TE optimization for only NS. If $\alpha=1$, the problem completely ignores the performance under NS and only optimizes the worst-case intra-AS TE performance across all FSs. While a specific value of α allows us to achieve a balance between the two objectives, the most suitable value depends on the combination of network topology and traffic matrix.

3.1.2.4 Constraint Value and our Heuristic Parameters

For the local search algorithm, we start with $\varepsilon=0.1$ for the inter-AS utilization constraint in (3.2) (i.e. the load on each inter-AS link should not exceed 10% of its capacity). However, if no solution that satisfies the constraint can be found, we step up the value by $c=0.1$ to relax the constraint. In this case, it becomes $\varepsilon_{new} = \varepsilon + n \times c = 0.1 + 1 \times 0.1 = 0.2$. If the algorithm remains unable to find a feasible solution, this value is then gradually increased by $n \times c$ until such a solution is found. ISPs can set the constraint and step values based on their desired operational objectives.

According to our experiments we realized that by setting our heuristic parameters to the following values we can achieve sufficiently good results: In the local search, the constraint value is increased if the utilization improvement is less than 2% after 20 iterations. For tabu search, the size of tabu list is set to 20, the threshold of utilization improvement for diversification is set to 5% of the best visited solution after 20 iterations. The stopping criterion is satisfied if either the search procedure reaches 5 times the total number of considered destination prefixes or the utilization improvement is less than 5% of the best visited solution after 10 consecutive diversifications.

3.1.3 Test Results

3.1.3.1 Alternative Approaches

We compare our joint robust TE with four alternative IGP link weight optimization approaches. The characteristics of these approaches are illustrated in Table 7.

1) **INVCAP**: as often used by vendors, the IGP link weights are set inversely proportional to the link capacity.

2) **INTRA-AS-TE**: the IGP link weights are optimized to achieve intra-AS load balancing only under NS. A notable work in this area is [FORT00]. However, it aims to minimize a piece-wise linear cost function which is not easily comparable with our objective function (3.1). For ease of comparison, we consider the objective of this approach also to be minimizing the intra-AS MLU under NS:

$$\underset{W}{Minimise}\, U^{intra}_{max\_NS} \tag{3.3}$$

We adopt the Tabu Search heuristic proposed in [NUCC03] for this approach and modify its link weight optimization only for NS.

3) **INTRA-AS-ROBUSTTE**: the IGP link weights are optimized to achieve intra-AS load balancing under both NS and intra-AS FSs. The objective of this approach can be formulated as:

$$\underset{W}{Minimise}(U^{intra}_{max\_NS},U^{intra}_{worst\_IntraFSs})=\underset{W}{Minimise}((1\text{-}\beta)U^{intra}_{max\_NS}+\beta U^{intra}_{worst\_IntraFSs}) \qquad (3.4)$$

where $0 \leq \beta \leq 1$. A notable work in this area is [NUCC07] with the consideration of an SLA constraint. We adopt their heuristic for this approach but without considering the SLA constraint. Note that since neither this approach nor **INTRA-AS-TE** account for the hot potato routing effect, the egress points of inter-AS traffic are assumed to be fixed whenever IGP link weight is changed.

4) **INTRA-AS-ROBUSTBGPTE**: the link weights are optimized to achieve intra-AS load balancing under both NS and intra-AS FSs (the same as the problem formulation (3.4)) while taking into account the hot potato routing. The closest related work to this approach is the METL-BGP TE tool [AGAR05]. However, they do not consider the impact of inter-AS link failure on the overall network utilization. To implement this approach, we extend the heuristic in [NUCC07] by incorporating the hot potato routing.

Approach	**TE for Normal State?**	**Robust to intra-AS link failure?**	**Consider HPR?**	**Robust to inter-AS link failure?**
INVCAP	No	No	No	No
INTRA-AS-TE	Yes	No	No	No
INTRA-AS- ROBUSTTE	Yes	Yes	No	No
INTRA-AS- ROBUSTBGPTE	Yes	Yes	Yes	No
JOINT-ROBUSTTE	Yes	Yes	Yes	Yes

Table 7 Various IGP Weight Optimization Approaches

3.1.3.2 Performance Metrics

The following performance metrics are used to evaluate and compare the proposed and the alternative approaches. For each of these metrics, lower values are better than high values:

- Intra-AS MLU under NS
- Worst case intra-AS MLU under intra- and inter-AS FSs
- Inter-AS MLU under NS and its worst case under intra- and inter-AS FSs

3.1.3.3 Intra-AS MLU under NS Performance Evaluation

Figure 35 (a,b) show intra-AS MLU for the 50-PoP and 100-PoP topologies under NS. The x-axis represents the normalized intra-AS offered load, i.e. the total intra-AS traffic volume normalized by the total intra-AS capacity. Note that all the results with *MLU>1.0* are not achievable. However they are illustrated for comparison purpose.

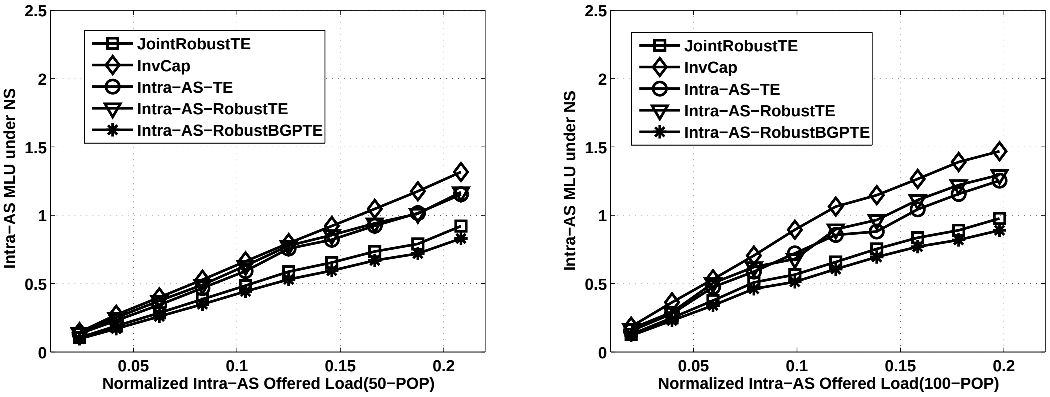


Figure 35 Intra-AS MLU under NS

From both figures we observe that **InvCap** is the worst performer, which is expected since it does not perform link weight optimization for achieving load balancing. **Intra-As-Te** and **Intra-As-RobustTe** perform better than **InvCap** but worse than the other two. In fact, even though these approaches aim to minimize the intra-AS MLU under NS or FSs according to their objective functions, they do not take the effects of HPR into account in their IGP link weight optimization. As a result, the actual routing of traffic in the network can be different from what was produced from the optimization, which may result in sub-optimal performance. With the explicit consideration of HPR, the joint robust TE and **Intra-As-RobustBgpTe** approaches outperform the others. However, the joint robust TE approach performs slightly worse (about 9%-10% for 50-PoP and 8%-10% for 100-PoP) than **Intra-As-RobustBgpTe**. This is because it attempts to optimize the intra-AS MLU under the inter-AS utilization constraint, whereas **Intra-As-RobustBgpTe** does not consider it. Adding such constraint reduces the number of feasible candidate egress points and therefore leads to fewer available IGP routes that can be selected by the traffic. This may result in the situation where many traffic flows traverse the same link, thereby significantly increasing its utilization. Nevertheless, as will be shown in the following sections, the joint robust TE significantly improves the intra- and inter-AS MLU under FSs at this small cost of performance degradation under NS.

3.1.3.4 Intra-AS MLU under Intra- and Inter-AS FSs Performance Evaluation

Figure 36 shows the worst-case intra-AS MLU across all intra- and inter-AS FSs. This Figure show that **InvCap** and **Intra-As-Te** appear to have the worst performance across all intra-AS FSs since they were not designed to be robust against intra-AS link failures. After these two approaches, **Intra-As-RobustTe** has the worst performance due to the ignorance of hot potato routing effects as it has been explained in the previous section. This reveals that hot potato routing is an essential consideration in the robust TE design. Therefore, **Intra-As-RobustTe** is the best performer in this case. Compared to it, the joint robust TE approach has slightly higher intra-AS MLU by about 7%-11% for 50-PoP and 8%-13% for 100-PoP. This is because **Intra-As-RobustTe** optimizes only for *intra-AS FSs* whereas the optimization objective of the joint robust TE covers not only intra- but also inter-AS FSs. The two set of FSs may conflict with each other: reducing the intra-AS link utilization under intra-AS FSs may increase the utilization under inter-AS FSs. As a result, we may not be able to obtain the best intra-AS MLU in exchange for achieving a compromised solution for inter-AS FSs, and this is explained in the next section. Figure 37 shows that the joint robust TE is the best performer regarding the worst-case intra-AS MLU across all inter-AS FSs (about 23%-33% for 50-PoP and 17%-21% for 100-PoP better than **Intra-As-RobustBgpTe**, the second best approach). The reason is that it is the only TE approach that is designed to be robust against inter-AS link failures. Failure of inter-AS links can cause egress point changes and reroute the traffic through highly utilized parts of the network that overloads some intra-AS links. This explains why the four alternative approaches perform significantly worse than the joint robust TE approach under inter-AS link failures.

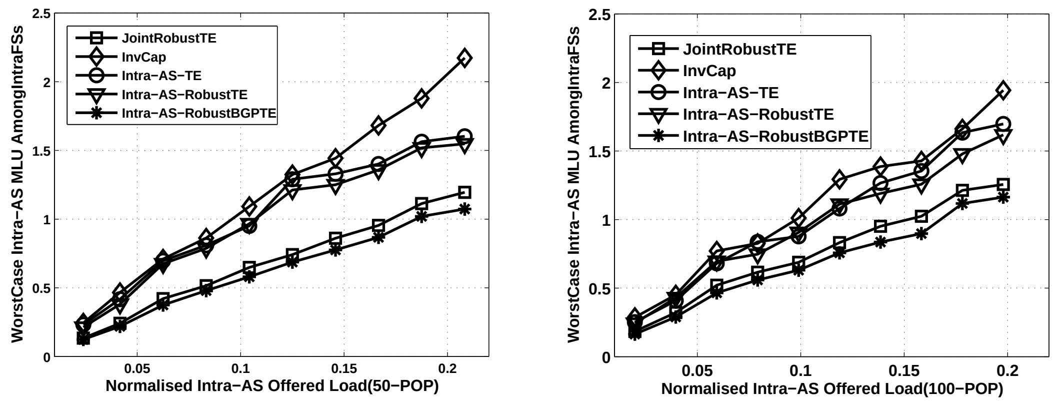


Figure 36 WorstCase Intra-AS MLU across Intra-AS FSs

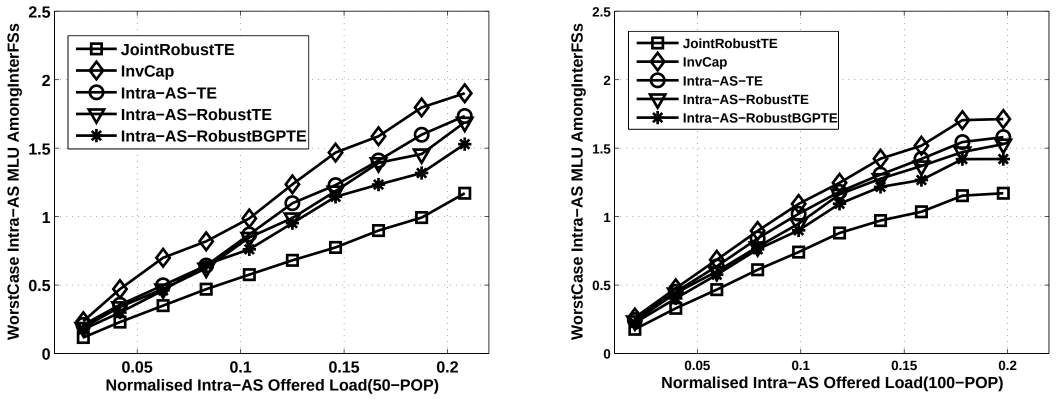


Figure 37 WorstCase Intra-AS MLU across Inter-AS FSs

3.1.3.5 Inter-AS MLU under NS, Intra- and Inter-AS FSs Performance Evaluation

Figure 38, Figure 39 and Figure 40 show the inter-AS MLU under NS and its worst case performance under intra- and inter-AS FSs. The x-axis represents the normalized inter-AS offered load, i.e. the total inter-AS traffic volume normalized by the total inter-AS link capacity. The values indicated by arrows are the inter-AS utilization constraint values (i.e. ε) as calculated according to Section 3.1.2.4. A general observation of the figures is that if the TE approach considers neither inter-AS load balancing under NS nor the impact of link failure on the utilization of inter-AS resources, like those four alternative approaches, a significant amount of traffic may be unpredictably assigned to some egress points and possibly cause severe congestion there. For instance, Figure 38 shows that the joint robust TE is the best performer regarding the inter-AS MLU under NS (about 23%-32% for 50-PoP and 21%-28% for 100-PoP better than **Intra-AS-RobustBgpTe**, the second best approach).

Moreover, by comparing Figure 39 and Figure 40, we can see that intra- and inter-AS link failures are equally contributed to the high utilization of inter-AS links. Hence, the robust TE approaches that neglect either intra- or inter-AS link failures may not make their performance truly robust. On the contrary, by considering both intra- and inter-AS link failures along with hot potato routing, our joint robust TE approach improves all the performance metrics. More specifically, its worst case inter-AS MLU across all intra-AS FSs and across all inter-AS FSs are about 22%-31% and 18%-36% respectively better than **Intra-AS-RobustBgpTe**, the second best approach for 50-PoP. Also for 100-PoP and the same performance metrics, it is about 27%-34% and 23%-35% better.

As mentioned in Section 3.1.2.4, we start with $\varepsilon = 0.1$. However the local search cannot find a feasible solution that satisfies the constraint until ε is increased to 0.2 and 0.3 for the 50 and 100-PoP topologies respectively. Note that, in practice, all the results with $\varepsilon > 1$ are undesirable due to egress point overload and potential packet losses. Nevertheless, even under this situation, the amount of overload is much smaller than the other alternative approaches.

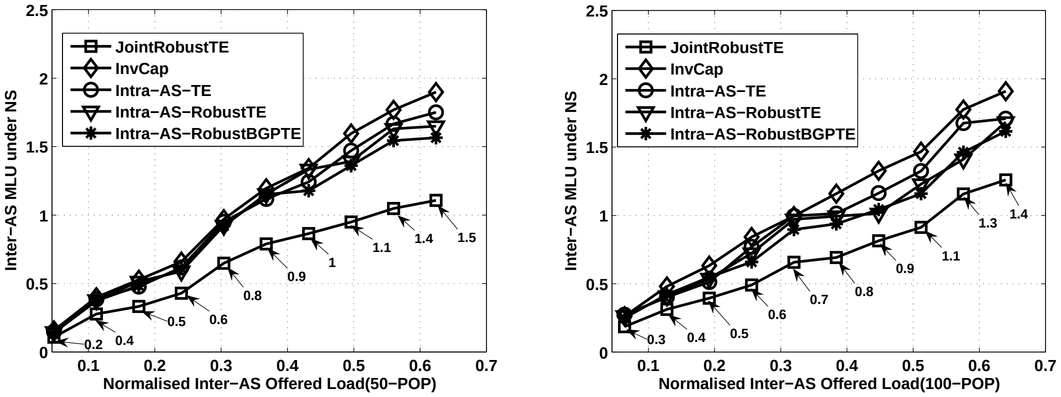


Figure 38 Inter-AS MLU under NS

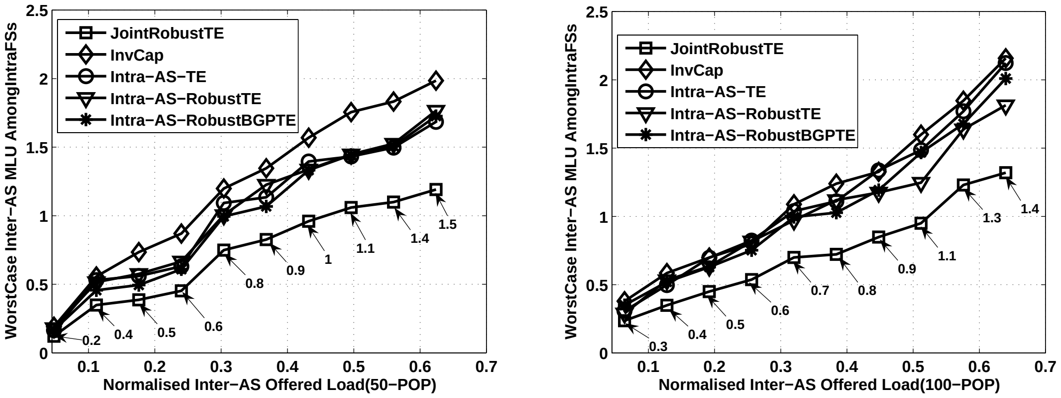


Figure 39 WorstCase Inter-AS MLU across Intra-AS FSs

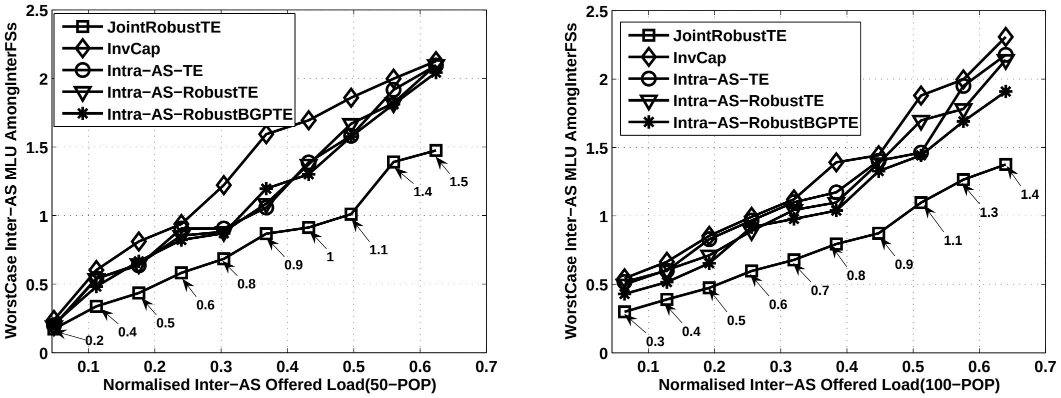


Figure 40 WorstCase Inter-AS MLU across Inter-AS FSs

3.1.3.6 Overall Performance

At the cost of a small performance degradation of the intra-AS MLU under NS, the joint robust TE approach significantly outperforms the other alternatives in terms of the worst-case intra- and inter-AS MLU across all FSs.

For those alternative approaches, **INVCAP** performs the worst in all the performance metrics. Although **INTRA-AS-TE** and **INTRA-AS-ROBUSTTE** have considered optimization for NS and intra-AS FSs, they can only perform better than **INVCAP** due to the ignoring of both hot potato routing effects and complete link failure scenarios. Clearly, **INTRA-AS-ROBUSTBGPTE** attempts to improve these deficiencies by incorporating the effects of hot potato routing. However, it does not perform well compared to our joint robust TE approach due to the ignoring of inter-AS link failures and hot potato routing impact on the overall network resource utilization.

3.1.4 Conclusions

In summary, our evaluation results show that our joint robust TE approach achieves high robustness of TE performance against transient intra- and inter-AS link failures. The other alternative approaches, however, do not satisfy all these objectives at the same time and hence their performance is less robust to link failures. In fact, based on the improved performance of the joint robust TE approach, we suggest that for the robust TE design: (1) intra- and inter-AS transient link failures should be considered together, and (2) the routing changes of hot-potato routing under normal and post-failure states should not be neglected when making changes to IGP link weights.

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 have 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 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 so emulate two networks topologies (ASes). In fact we'll 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 in one hand the virtual router technology as some limitations regarding performances and can only simulate a small number of routers (typically fifteen) and in 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 optimized for these tests. For example, the topology of a customer network dually connected to an INP using a hierarchical route reflector topology would be:

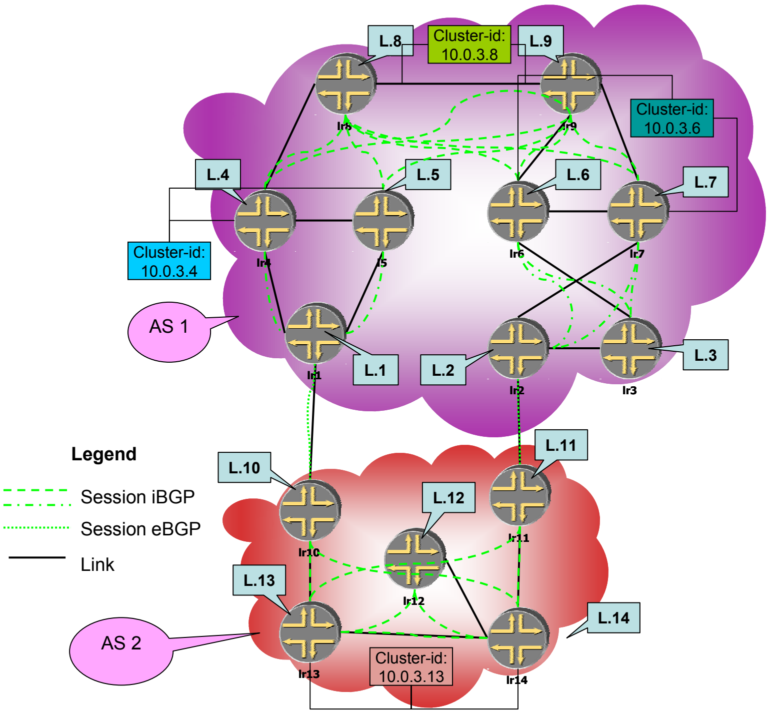


Figure 41 Example of a customer network dually connected to an INP using a hierarchical RR

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 don't need any traffic matrix as for time accuracy only a limited set of flows will be used. (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, we'll inject BGP routing tables from a real Internet network.

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

Table 8 Selection conditions

3.2.3.3 Structure by subgroups of elementary test

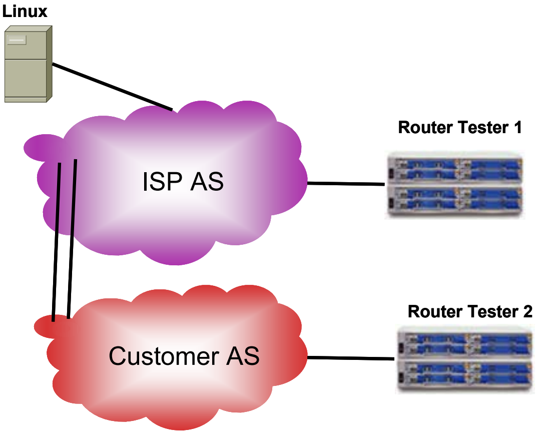


Figure 42 BGP Testbed diagram

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 route 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 link 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 loss & 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, updates messages sent and received on all routers are captured.
- Finally, 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 & status of the BGP session in 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 that 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.

Table 9 Reference sub-groups

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.

Table 10 REF / CURRENT reference subgroups

3.2.3.3.1.1 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

Table 11 REF / CURRENT / IP reference subgroups

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

This subgroup defines two different customers <> ISP architectures:

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

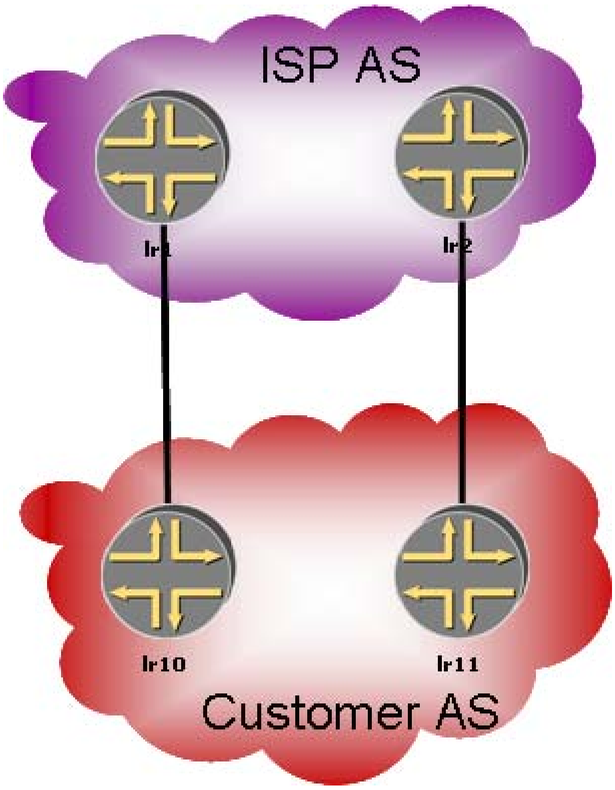


Figure 43 2CE-2PE architecture

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

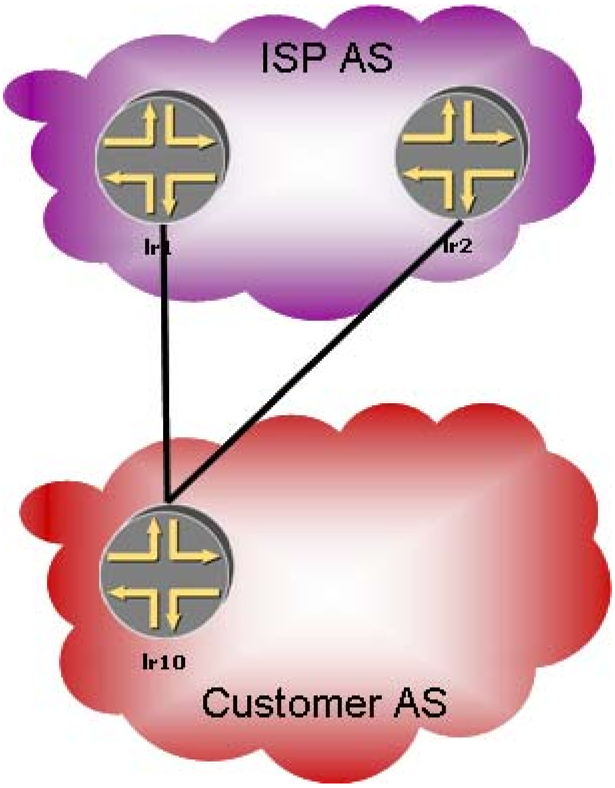
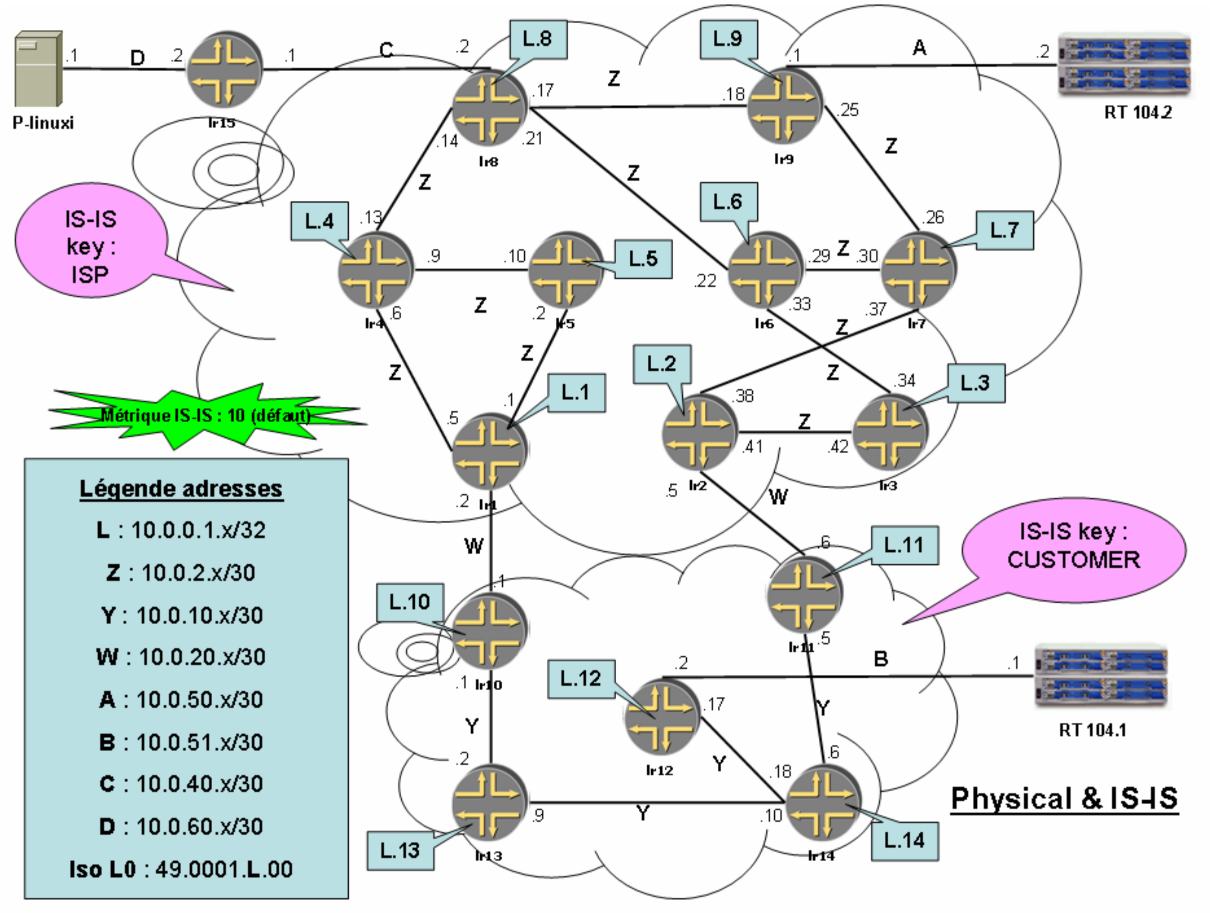


Figure 44 2PE-1CE architecture

The physical and ISIS architecture used is the following:

(a) 2CE-2PE architecture:



Légende adresses

L : 10.0.0.1.x/32

Z : 10.0.2.x/30

Y : 10.0.10.x/30

W : 10.0.20.x/30

A : 10.0.50.x/30

B : 10.0.51.x/30

C : 10.0.40.x/30

D : 10.0.60.x/30

Iso L0 : 49.0001.L.00

Figure 45 2CE-2PE physical and ISIS architecture

(b) 2PE-1CE architecture:

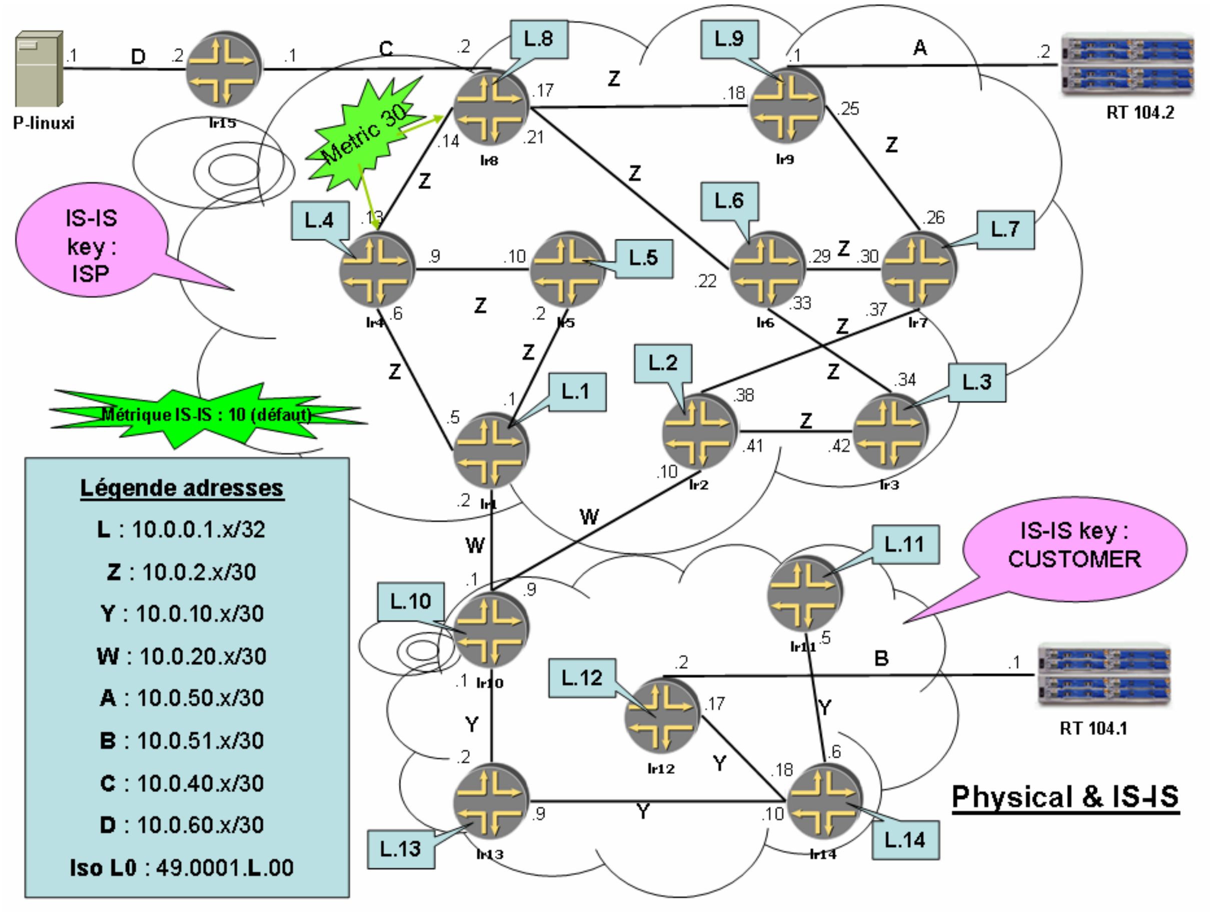


Figure 46 2PE-1CE physical and ISIS architecture

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 send 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 link.

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. The details for these topologies are explained in the section 16.

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.

Table 12 REF / CURRENT / IP / eBGP reference subgroups

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

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

(a) Full mesh topology: the routers in each AS are fully meshed with the others by using iBGP sessions.

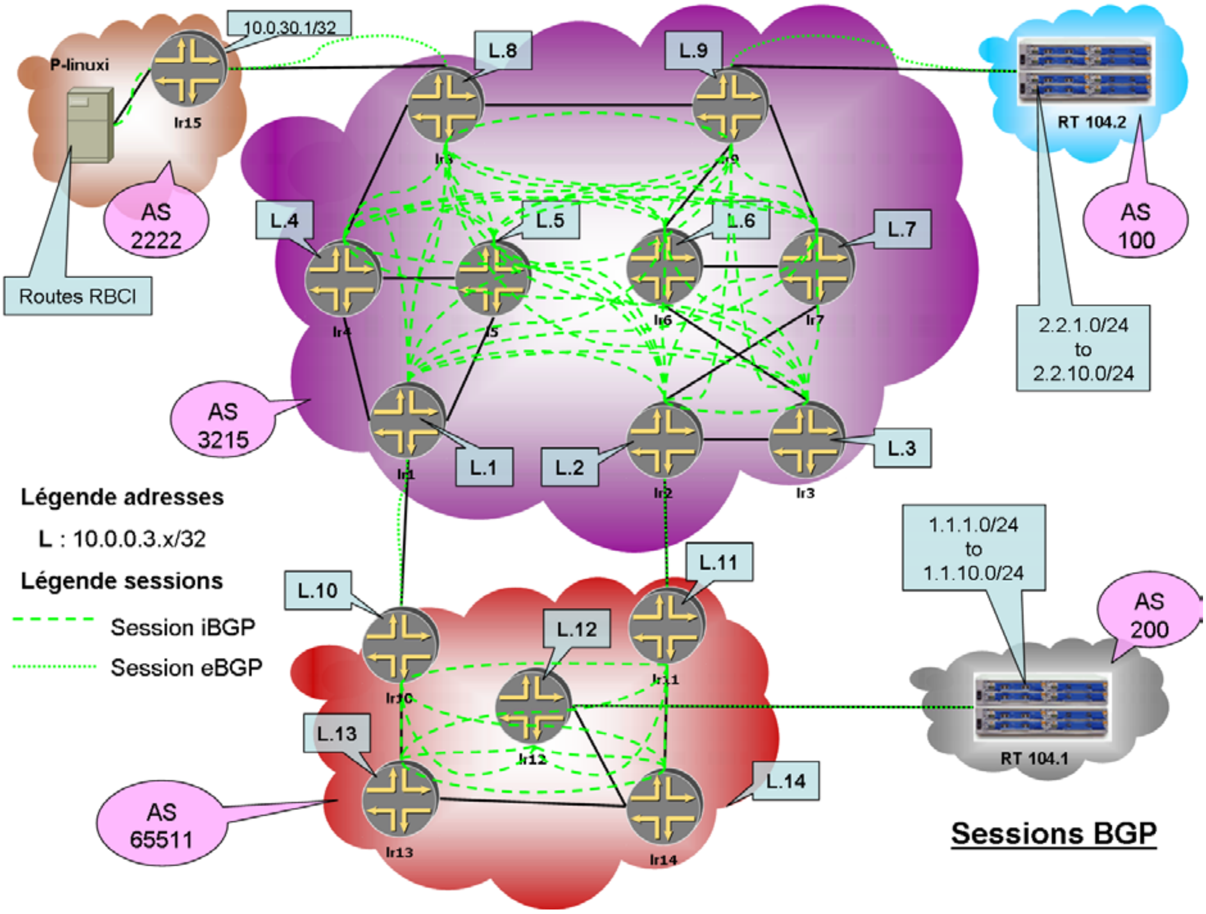


Figure 47 Full mesh topology

(b) Route Reflector topology

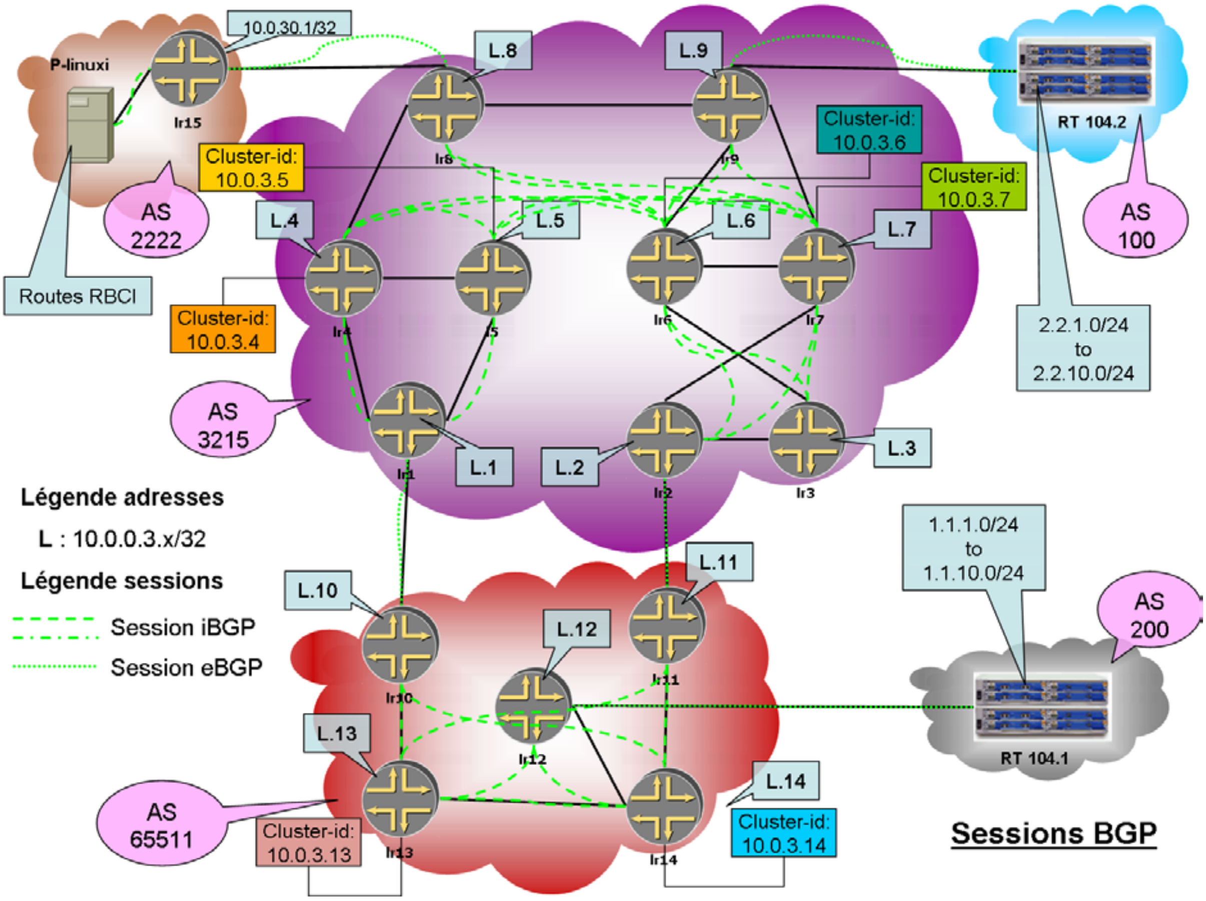


Figure 48 Route Reflector topology

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.

(c) Hierarchical Route Reflector topology with different Cluster-ID for each RR.

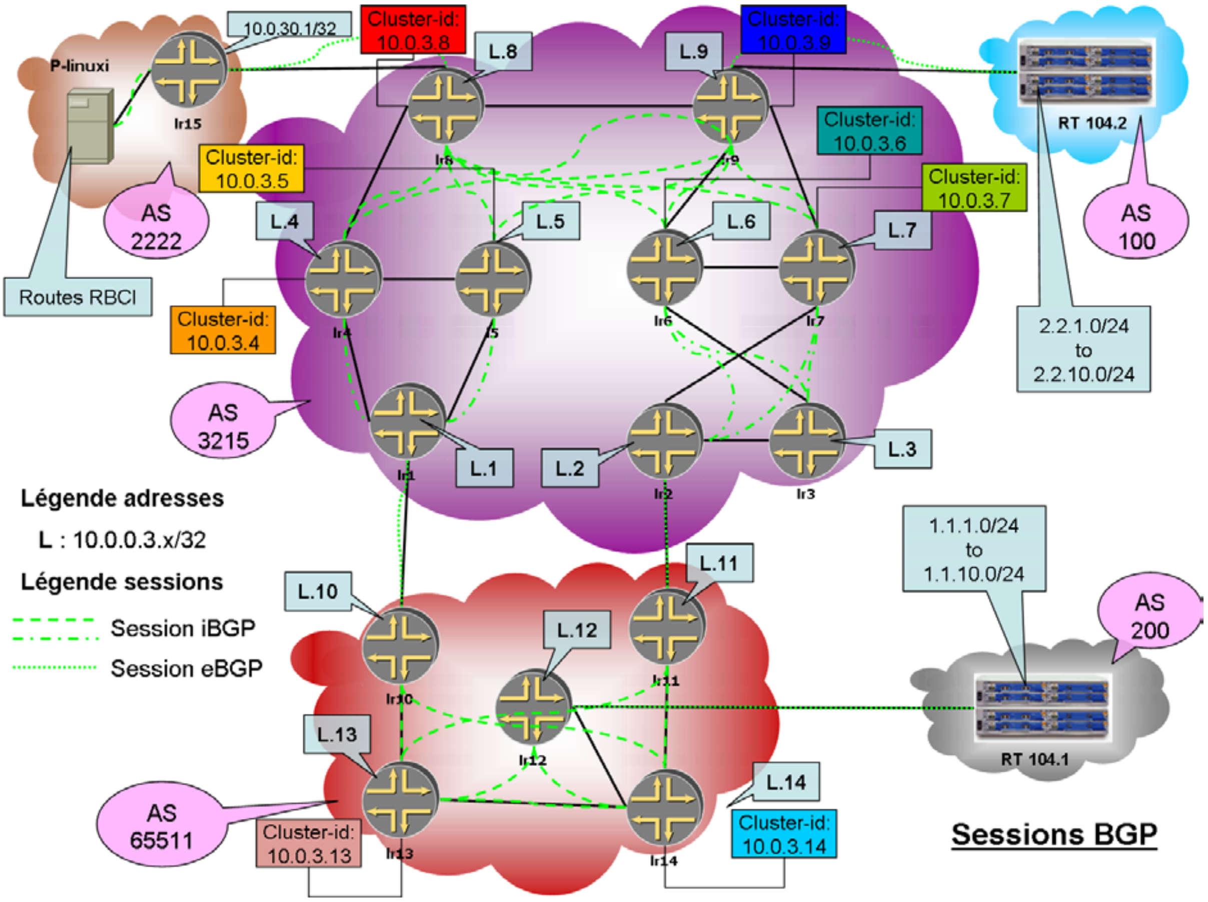


Figure 49 Hierarchical Route Reflector topology with different cluster-ID for each RR

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.

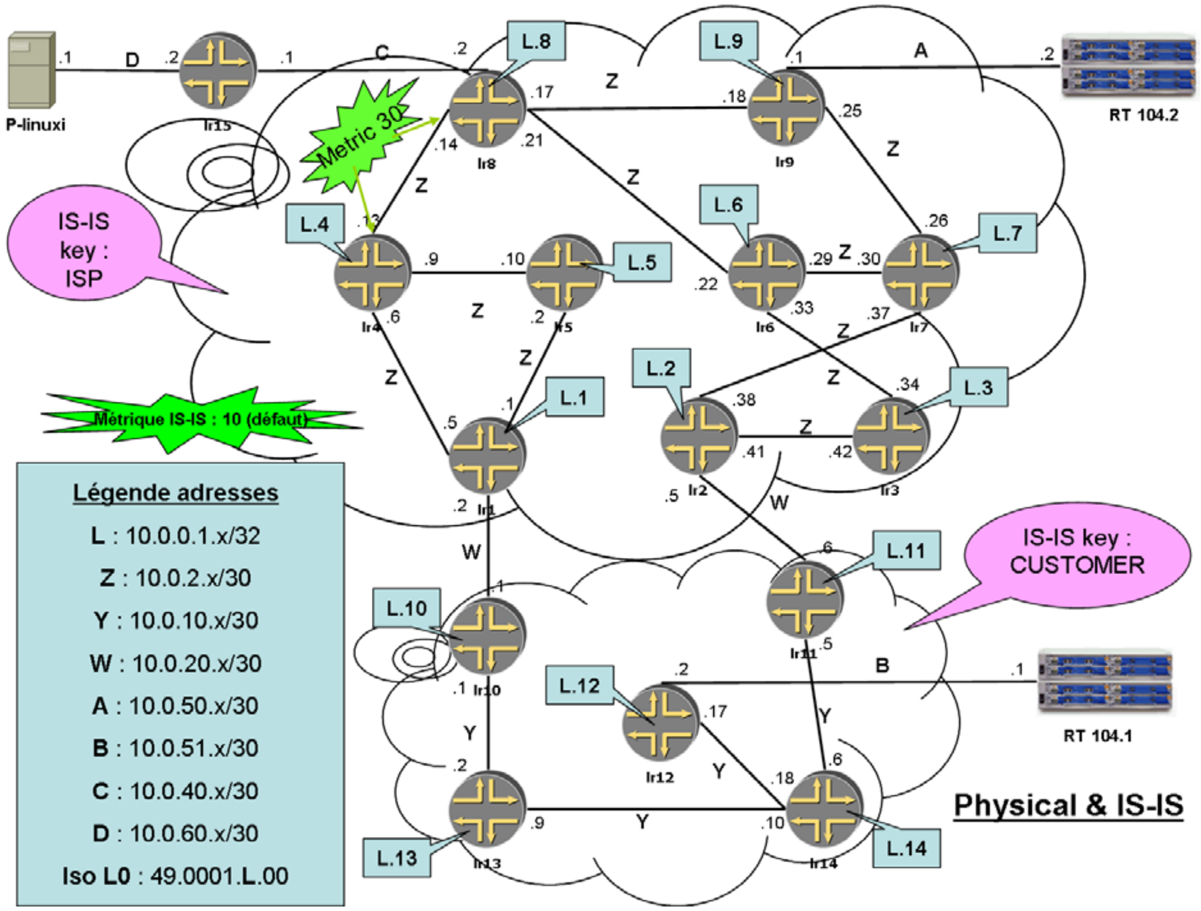


Figure 50 Physical and ISIS architecture of the HRR topology

(d) Hierarchical Route Reflector topology with the same Cluster-ID for each pair of redundant Reflector Routers.

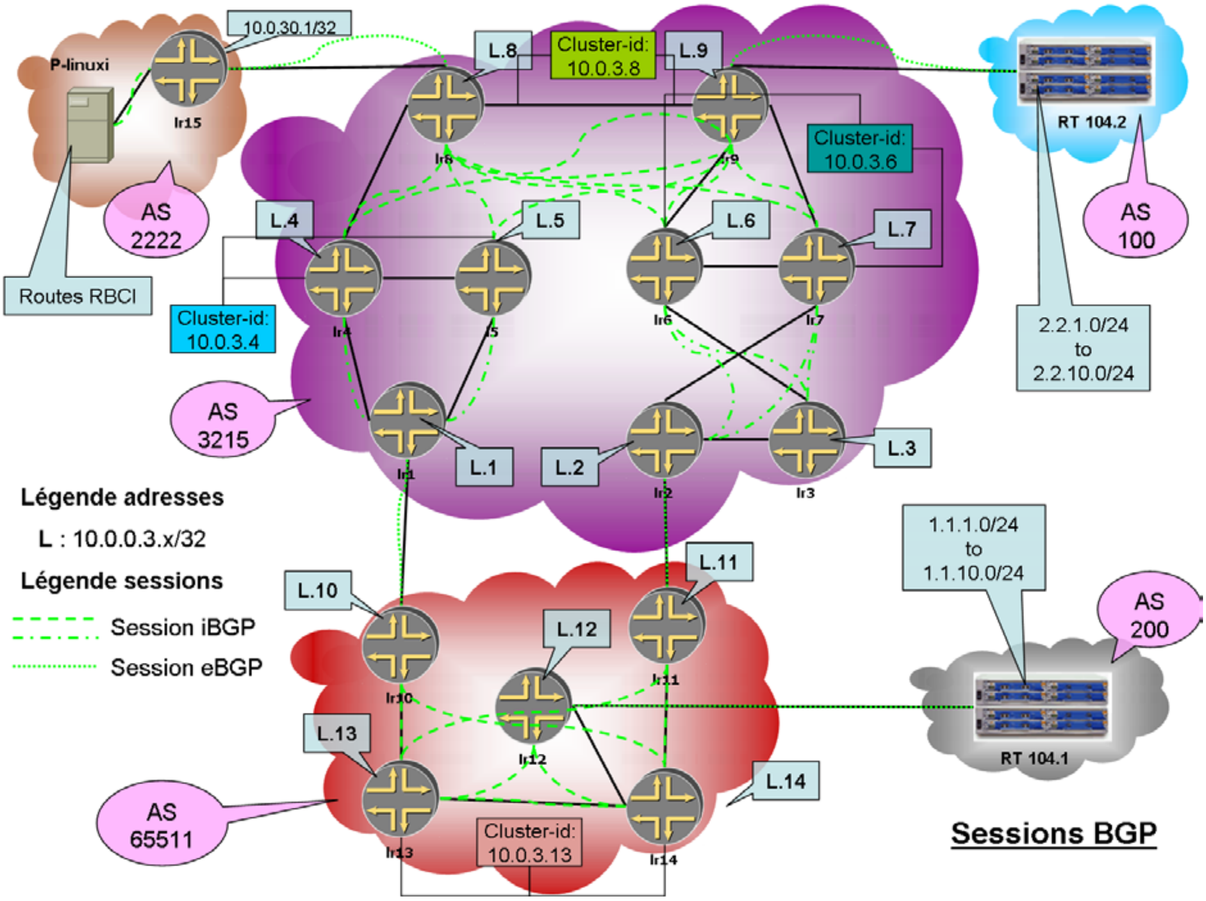


Figure 51 Hierarchical Route Reflector topology with the same Cluster-ID for each pair of redundant RR

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.

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.

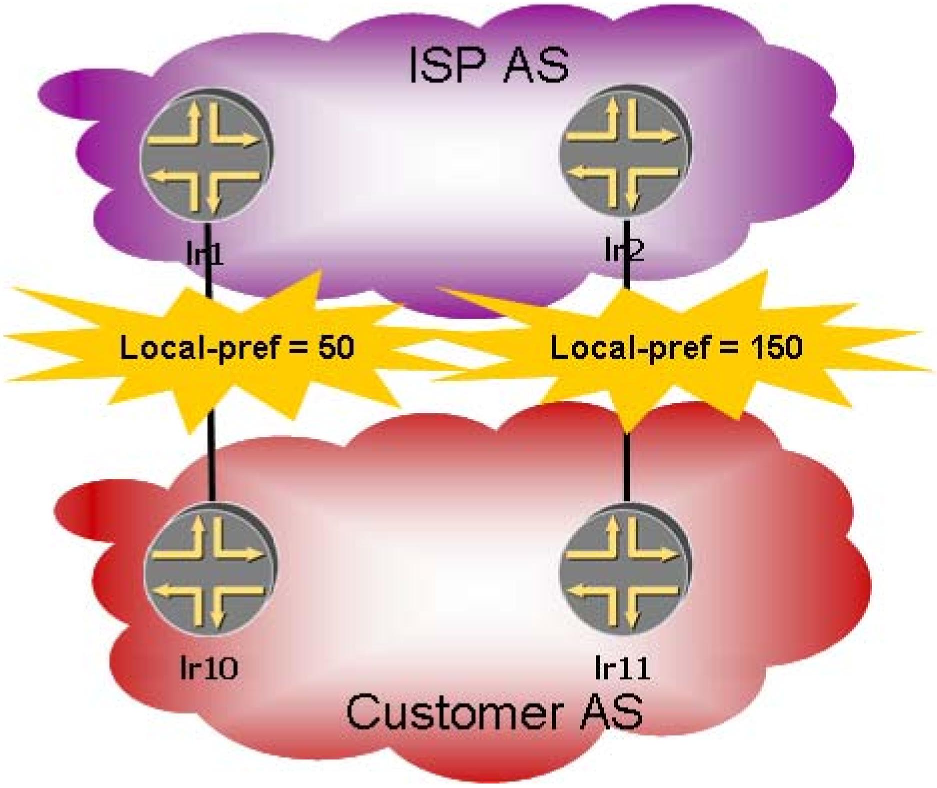


Figure 52 Local preference attribute forcing scenario

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 /	M	Normal	The iBGP topology used in the customer AS is a

Ref. of test	Condition of selection	Priority	Object
iBGP IGP / HRR_same_Cluster_Id			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.

Table 13 REF / CURRENT / IP / iBGP reference subgroups

3.2.3.3.1.2 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.

3.2.3.3.1.3 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.

Table 14 REF / CURRENT / VPN / eBGP reference subgroups

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.

Table 15 REF / CURRENT / VPN / iBGP reference subgroups

3.2.3.3.2 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 Test-bed

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 that 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	

Table 16 Description of used equipment

3.2.5 Test procedures

3.2.5.1 Tests with the current BGP behaviour

3.2.5.1.1 eBGP topology with 2PE-1CE:

The commands to deactivate the link are:

```
JM7B@JM7B# deactivate logical-routers lr10 protocols bgp group ebgp neighbor 10.0.20.10
JM7B@JM7B#commit
```

The commands to activate the link are:

```
JM7B@JM7B# activate logical-routers lr10 protocols bgp group ebgp neighbor 10.0.20.10
JM7B@JM7B#commit
```

3.2.5.1.2 eBGP topology with 2PE-2CE:

The commands to deactivate the link are:

```
JM7B@JM7B# deactivate logical-routers lr11 protocols bgp group ebgp
JM7B@JM7B#commit
```

The commands to activate the link are:

```
JM7B@JM7B# activate logical-routers lr11 protocols bgp group ebgp
JM7B@JM7B#commit
```

The commands to deactivate the BGP protocol are:

```
JM7B@JM7B# deactivate logical-routers lr11 protocols bgp
JM7B@JM7B#commit
```

The commands to activate the BGP protocol are:

```
JM7B@JM7B# activate logical-routers lr11 protocols bgp
JM7B@JM7B#commit
```

3.2.5.2 Tests with the Planned Maintenance BGP behaviour

3.2.5.2.1 eBGP topology with 2PE-1CE:

To induce a planned maintenance:

```
JM7B@JM7B#delete logical-routers lr10 protocols bgp group ebgp export BGP
JM7B@JM7B#set logical-routers lr10 protocols bgp group ebgp import PM-ebgp-import
JM7B@JM7B#set logical-routers lr10 protocols bgp group ebgp export PM-export
JM7B@JM7B#commit
```

Then the link is shutdown as above:

```
JM7B@JM7B# deactivate logical-routers lr10 protocols bgp group ebgp neighbor 10.0.20.10
JM7B@JM7B#commit
```

Then the link is reactivated:

```
JM7B@JM7B# activate logical-routers lr10 protocols bgp group ebgp neighbor 10.0.20.10
JM7B@JM7B#commit
```

And the planned maintenance is deleted:

```
JM7B@JM7B#set logical-routers lr10 protocols bgp group ebgp export BGP
JM7B@JM7B#delete logical-routers lr10 protocols bgp group ebgp import PM-ebgp-import
JM7B@JM7B#delete logical-routers lr10 protocols bgp group ebgp export PM-export
JM7B@JM7B#commit
```

3.2.5.2.2 eBGP topology with 2PE-2CE

To induce a planned maintenance:

```
JM7B@JM7B#delete logical-routers lr11 protocols bgp group ebgp export BGP
JM7B@JM7B#delete logical-routers lr11 protocols bgp group ibgp-rr export BGP
JM7B@JM7B#set logical-routers lr11 protocols bgp group ibgp-rr export PM-ibgp-export
JM7B@JM7B#set logical-routers lr11 protocols bgp group ebgp export PM-export
JM7B@JM7B#commit
```

Secondly the link is shutdown:

```
JM7B@JM7B# deactivate logical-routers lr11 protocols bgp group ebgp
JM7B@JM7B#commit
```

Then the link is reactivated:

```
JM7B@JM7B# activate logical-routers lr11 protocols bgp group ebgp
JM7B@JM7B#commit
```

And finally the planned maintenance is deleted:

```
JM7B@JM7B#set logical-routers lr11 protocols bgp group ebgp export BGP
JM7B@JM7B#set logical-routers lr11 protocols bgp group ibgp-rr export BGP
JM7B@JM7B#delete logical-routers lr11 protocols bgp group ibgp-rr export PM-ibgp-export
JM7B@JM7B#delete logical-routers lr11 protocols bgp group ebgp export PM-export
JM7B@JM7B#commit
```

3.2.5.3 *BGP policy used for the Planned Maintenance behaviour*

The policies used to induce a planned maintenance are the following:

For lr2 (the router with the link impacted by the planned maintenance performed on lr10 or lr11)

```
policy-statement PM-export {
    term 1 {
        from {
            protocol bgp;
            community maintenance;
        }
        then {
            local-preference 0;
            next-hop self;
            accept;
         }
      }
      term 2 {
         from protocol bgp;
         then {
            next-hop self;
         }
      }
   }
   community maintenance members 3215:6666;
```

And the planned maintenance policy is applied in export of the iBGP session of lr2:

```
  group ibgp-rr {
         […]
         export PM-export;
        […]
```

For lr10 (in case of 1CE-2PE topologies):

```
   policy-statement PM-ebgp-import {
      term 1 {
         from neighbor 10.0.20.10;
         then {
            local-preference 0;
            accept;
         }
      }
   }
   policy-statement PM-export {
      term 1 {
         then {
            community add maintenance;
            accept;
         }
      }
   }
   community maintenance members 3215:6666;
}
```

For lr11 (in case of 2CE-2PE topologies):

```
   policy-statement PM-ibgp-export {
      term 1 {
         from neighbor 10.0.20.5;
         then {
            local-preference 0;
            accept;
         }
      }
      term 2 {
         from protocol bgp;
         then {
            next-hop self;
         }
      }
   }
```

```
policy-statement PM-export {
   term 1 {
      then {
         community add maintenance;
         accept;
      }
   }
}
community maintenance members 3215:6666;
```

3.2.5.3.1 Minor modifications

A minor correction has been made in the planned maintenance Policies after the tests. Although this change does not have any impact on the results, it has to be reported to correct an improperly configured element.

The policy PM-export applied on logical router lr11 for 2PE-2CE topologies and on logical router lr10 for 2PE-1CE topologies must be changed for the following one:

```
policy-statement PM-export {
   term 1 {
      then {
         community add maintenance;
      }
   }
   term 2 {
      from {
         route-filter 10.0.3.0/24 orlonger;
      }
      then accept;
   }
}
```

The Policy PM-ibgp-export must also be changed for the following one:

```
policy-statement PM-ibgp-export {
   term 1 {
      from neighbor 10.0.20.5;
      then {
         local-preference 0;
      }
   }
   term 2 {
      then {
         next-hop self;
      }
   }
   term 3 {
      from {
         route-filter 10.0.3.0/24 orlonger;
      }
      then accept;
   }
}
```

The policy PM-export applied on logical router lr2 must be changed for the following one:

```
policy-statement PM-export {
   term 1 {
```

```
            from {
               protocol bgp;
               community maintenance;
            }
            then {
               local-preference 0;
               next-hop self;
                        }
         }
         term 2 {
            from protocol bgp;
            then {
               next-hop self;
            }
         }
         term 3 {
            from {
               route-filter 10.0.3.0/24 orlonger;
            }
            then accept;
         }
      }
```

Theses policies will filter the ISIS loopback as it has been done previously when testing current behaviour of IP/BGP, MPLS/BGP and MPLS/BGP/L3VPN and advertise the BGP loopback.

Nevertheless, the only problem with the previous policies was the advertisement of the ISIS loopback and the non advertisement of the BGP loopback of logical router lr2 & lr10 or lr11, which is not a problem for our test but is an improper behaviour in actual networks.

3.2.6 Tests results

3.2.6.1 For the streams going upward

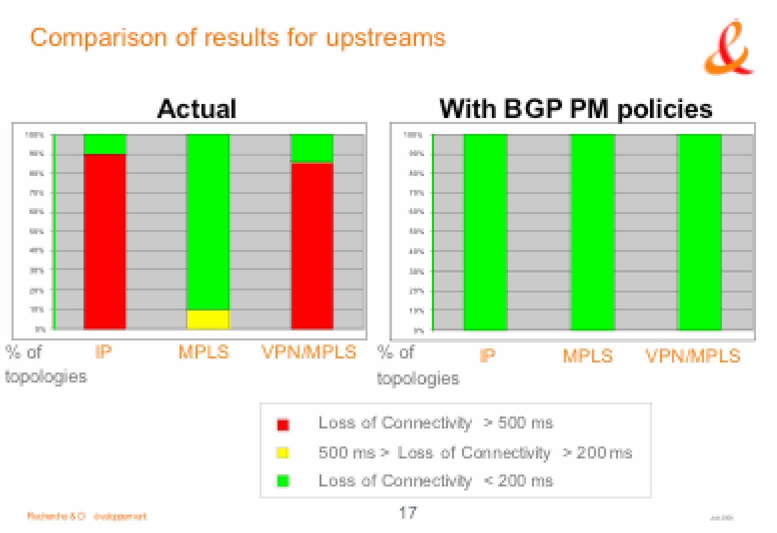
Forwarding plane	topology	Current BGP behaviour	Planned Maintenance BGP behaviour	Gain %
IP	current\IP\eBGP\2PE-1CE\ (HRR_same_Cluster_ID iBGP)	0,0000	0,0000	0,00
IP	current\IP\iBGP\IGP\Full-mesh	1,8794	0,0000	100,00
IP	current\IP\iBGP\IGP\RR	2,0587	0,0000	100,00
IP	current\IP\iBGP\IGP\HRR_diff_Cluster_Id	1,7077	0,0000	100,00
IP	current\IP\iBGP\IGP\HRR_same_Cluster_Id	2,1702	0,0000	100,00
IP	current\IP\iBGP\LP\Full-mesh	1,9828	0,0000	100,00
IP	current\IP\iBGP\LP\RR	2,1554	0,0000	100,00
IP	current\IP\iBGP\LP\HRR_diff_Cluster_Id	2,1839	0,0000	100,00
IP	current\IP\iBGP\LP\HRR_same_Cluster_Id	2,0547	0,0000	100,00
IP	current/IP/ iBGP IGP/HRR_same_Cluster_Id_down_BGP	0,5165	0,0000	100,00
MPLS	current\MPLS\eBGP\2PE-1CE	0,0000	0,0000	0,00
MPLS	current\MPLS\iBGP\IGP\Full-mesh	0,0000	0,0000	0,00

MPLS	current\MPLS\iBGP\IGP\RR	0,0000	0,0000	0,00
MPLS	current\MPLS\iBGP\IGP\HRR_diff_Cluster_Id	0,0000	0,0000	0,00
MPLS	current\MPLS\iBGP\IGP\HRR_same_Cluster_Id	0,0000	0,0000	0,00
MPLS	current\MPLS\iBGP\IGP\HRR_same_Cluster_Id_down_BGP	0,4786	0,0000	100,00
MPLS	current\MPLS\iBGP\LP\Full-mesh	0,0000	0,0000	0,00
MPLS	current\MPLS\iBGP\LP\RR	0,0000	0,0000	0,00
MPLS	current\MPLS\iBGP\LP\HRR_diff_Cluster_Id	0,0000	0,0000	0,00
MPLS	current\MPLS\iBGP\LP\HRR_same_Cluster_Id	0,0000	0,0000	0,00
VPN/MPLS	current\VPN\eBGP\1CE-2PE	0,0000	0,0000	0,00
VPN/MPLS	current\VPN\iBGP\IGP\1RD	1,4037	0,0000	100,00
VPN/MPLS	current\VPN\iBGP\IGP\2RD	1,9704	0,0000	100,00
VPN/MPLS	current\VPN\iBGP\LP\1RD	1,6421	0,0000	100,00
VPN/MPLS	current\VPN\iBGP\LP\2RD	1,6773	0,0000	100,00
VPN/MPLS	current\VPN\inter-AS-optionB\IGP	16,3654	0,0000	100,00
VPN/MPLS	current\VPN\inter-AS-optionB\LP	17,6338	0,0000	100,00

Table 17 Upward results

- In case of a single homed dual attached customer with a single router connecting to the ISP but redundant links (i.e. 1CE-2PE), there is no packets lost in the upstream (AS200 to AS100) since the CE router already knows the alternative route and originates the shutdown of the eBGP session and no convergence is necessary for all the others routers of the AS (the next-hop is the same).
- The behaviour above is the same for all MPLS tests since lr11 also already knows the alternate path via lr10 and originates the shutdown of the eBGP session. The topology of the AS client is a simple topology where lr13 & lr14 both choose a different next-hop and so advertise a different path to lr11. Since lr13 and lr14 does not perform an IP lookup in MPLS forwarding mode, no transient loops are possible (in opposition with IP).
- For VPN / MPLS, A loss of connectivity is noticed. For the Client AS, the forwarding mode is IP as for the first tests with a RR topology. For the RR in IP, the test gives a result of 2.0587 seconds. For VPN /MPLS, the average value is 1.6734 seconds which is slightly less. I do not understand why since VPN / MPLS forwarding normally induces more calculus for the routing engine.
- For inter AS VPN, the loss of connectivity is far more important than for any other topology. This is probably due to the complexity of the operation to be done in case of this topology. We probably have reached the limit of the logical router design…
- When the BGP process is shutdown instead of the BGP session, the loss of connectivity is less important for IP forwarding.

With BGP planned maintenance, when IP forwarding is used, transient loops happen. These transient loops induce a small loss of connectivity. Logically the loss is the smallest for a full-mesh iBGP topology. In the case of the Customer AS, RR topologies = HRR with different cluster ID. Nevertheless the loss is not the same even if the Customer topology is the same. This might be induced by the iBGP topology of the ISP, which is not the same and thus induces overhead in the CPU, which influences the loss in the Customer AS.



Comparison of results for upstreams

🟥 Loss of Connectivity > 500 ms
🟨 500 ms > Loss of Connectivity > 200 ms
🟩 Loss of Connectivity < 200 ms

17

Figure 53 Comparison of upward results

If MPLS is used, the actual behaviour is quiet good for a topology as simple as the topology of the Customer AS. Nevertheless the IP & MPLS VPN forwarding behaviour is quiet bad for nearly all the cases. The use of planned maintenance policies clearly improves the behaviour of the convergence. With this simple topology were loops are limited, the loss of connectivity is cancelled.

The actual gain for each mode of forwarding is calculated further on.

3.2.7 For the streams going downward

The topology in the ISP AS is more complicated that the very simple one of the Customer AS and as a consequence the results are far more complicated.

Forwarding plane	topology	Current BGP behavior	Planned Maintenance BGP behavior	Gain %
IP	current\IP\eBGP\2PE-1CE\ (HRR_same_Cluster_ID iBGP)	1,2489	0,4032	67,72
IP	current\IP\iBGP\IGP\Full-mesh	0,2489	0,0647	74,02
IP	current\IP\iBGP\IGP\RR	1,2207	0,2566	78,98
IP	current\IP\iBGP\IGP\HRR_diff_Cluster_Id	1,1465	0,1985	82,69
IP	current\IP\iBGP\IGP\HRR_same_Cluster_Id	0,5602	0,2962	47,12
IP	current\IP\iBGP\LP\Full-mesh	2,2983	0,3909	82,99
IP	current\IP\iBGP\LP\RR	2,4078	0,3338	86,14
IP	current\IP\iBGP\LP\HRR_diff_Cluster_Id	2,1801	0,2124	90,26
IP	current\IP\iBGP\LP\HRR_same_Cluster_Id	2,1330	0,1367	93,59
IP	Current/ IP/ iBGP IGP/HRR_same_Cluster_Id_down_BGP	3,6738	0,1703	95,37
MPLS	current\MPLS\eBGP\2PE-1CE	1,0795	0,0000	100,00
MPLS	current\MPLS\iBGP\IGP\Full-mesh	0,0000	0,0000	0,00
MPLS	current\MPLS\iBGP\IGP\RR	0,9833	0,0000	100,00

MPLS	current\MPLS\iBGP\IGP\HRR_diff_Cluster_Id	3,3675	0,0000	100,00
MPLS	current\MPLS\iBGP\IGP\HRR_same_Cluster_Id	3,5411	0,0000	100,00
MPLS	Current\MPLS\iBGP\IGP\HRR_same_Cluster_Id_down_BGP	0,6315	0,0000	100,00
MPLS	current\MPLS\iBGP\LP\Full-mesh	1,9484	0,0000	100,00
MPLS	current\MPLS\iBGP\LP\RR	1,7090	0,0000	100,00
MPLS	current\MPLS\iBGP\LP\HRR_diff_Cluster_Id	3,2660	0,0000	100,00
MPLS	current\MPLS\iBGP\LP\HRR_same_Cluster_Id	3,2560	0,0000	100,00
VPN/MPLS	current\VPN\eBGP\1CE-2PE	3,9174	0,0000	100,00
VPN/MPLS	current\VPN\iBGP\IGP\1RD	1,5797	0,0000	100,00
VPN/MPLS	current\VPN\iBGP\IGP\2RD	2,3201	0,0000	100,00
VPN/MPLS	current\VPN\iBGP\LP\1RD	7,1522	0,0000	100,00
VPN/MPLS	current\VPN\iBGP\LP\2RD	5,3388	0,0000	100,00
VPN/MPLS	current\VPN\inter-AS-optionB\IGP	5,1972	0,0000	100,00
VPN/MPLS	current\VPN\inter-AS-optionB\LP	8,7612	0,0000	100,00

Table 18 Downward results

- The relative complexity of the ISP AS induces a lot of transient loops during the BGP convergence, this is shown by the results of the IP forwarding mode and the loss of connectivity still measured with the planned maintenance policies.
- The presence of a Local_Pref attribute generally increases the interruption time. This is particularly true with the full-mesh and the L3VPN architectures.
- The result in MPLS with a full-mesh iBGP topology shows that if the PE routers know the alternative route; there won't be any loss due to transient loops.
- It can be noticed that the loss of connectivity for MPLS forwarding is less than IP forwarding for "simple" topologies (full-mesh & RR) even if local_pref attribute is used in lr1 to hide the alternative route. Nevertheless for HRR topologies, the loss of connectivity is more important for MPLS forwarding. This may be due to the sequential nature of JunOS (i.e. FreeBSD processes) and the use of logical routers: with more complex topologies, more calculations are done thus delaying the update of the FIB and as a consequence raising the loss of connectivity time.
- Similar upstream interruption times are measured for the IP and VPN architectures (except for full-mesh). This behaviour was expected since the client architecture is exactly the same for all theses topologies.
- For the shutdown of the whole BGP process, strange behaviour can be noticed which I cannot explain like in the upstream case.
- As for up streams, the MPLS /VPN loss of connectivity is far more important than the other time probably for the same reason.
- The application of the Planned Maintenance policies solves the loss of connectivity for MPLS & MPLS VPN which is not surprising since lr2 & lr10/lr11 go on forwarding during the AS convergence and since transient loops are eliminated by the use of MPLS. It is to be noted that since LDP was used, the tunnels lr1 <> lr2 and lr11 <> lr10 were automatically provisioned, if RSVP were to be used, these tunnel should be provisioned to reproduce the same behaviour.
- Surprisingly, using 2 RD in VPN MPLS does increase the loss of connectivity for 3 cases (both up streams and 2RD with IGP metrics).

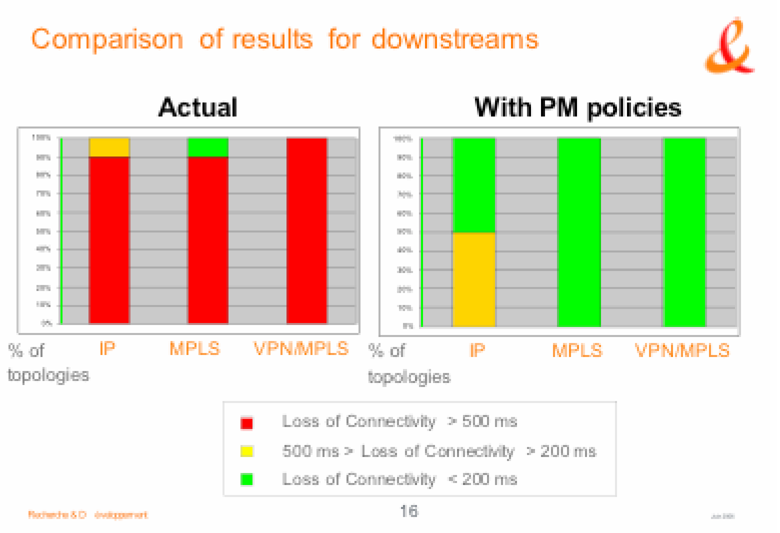


Figure 54 Comparison of downward results

For a complex topology, the actual BGP convergence induces a lot of loss of connectivity for nearly all the topologies tested. Nevertheless this behaviour is much better with planned maintenance policies. In IP forwarding mode, there is still some loss of connectivity detected but they are mainly due to transient loops during the convergence mainly caused by the propagation of the messages and the Route Reflector information selection.

3.2.8 Summary

In summary, the gain brought by using a planned maintenance strategy is the following:

Mean gain (%)	IP	MPLS	MPLS/VPN	Mean Total
Down Stream	85,61	100	100	96,54
Up Stream	100	100	100	100
Mean Total	92,81	100	100	98,27

Table 19 Results summary

It can be seen that Planned Maintenance policies on the overall achieve a high gain even if some losses still exist in IP. It is to be noticed that the mean gain is calculate for the topologies were a gain was possible. A topology with no loss in current behaviour will not be affected by planned maintenance.

3.2.9 Conclusion

When MPLS forwarding is used, including for BGP/MPLS VPN, the planned maintenance strategy can achieve a hitless shutdown with 0 packet loss.

When IP forwarding is used, the BGP planned maintenance strategy significantly reduces the loss of connectivity. However, we still have a loss of connectivity, which is dependant of the iBGP and network topology, probably caused by transient routing loops during the iBGP convergence.

Results are therefore very encouraging and it seems that if the availability requirement is very high (such as for VoIP), the use of MPLS combined with the planned maintenance evolution could result in a zero packet loss during BGP planned maintenance operations.

3.3 IP Tunnelling

The validation and evaluation of the IP Tunnelling solution is made up of four different parts. The first part, described in Section 3.3.1, is based on simulations exploring the stability and scalability of the Tunnel Service (TS), which is based on the LISP proposal ([LISP07]). The second part, described in Section 3.3.2, evaluates the TS prototype, by showing some measurements performed on the AGAVE integrated test-bed (which is described in Section 4.1). The third part, in Section 3.3.3, shows an evaluation of the Tunnel Service Controller (TSC), which is based on the IDIPS proposal. Finally, the forth part, described in Section 3.3.4, shows how the TS and TSC interact, through some measurements performed on the AGAVE Integrated Test-bed.

3.3.1 LISP (Tunnel Service) Simulation-based Experimentation

3.3.1.1 Objectives

The LISP Protocol that we use as Tunnel Service used in the AGAVE Project is based on the idea of separating the IP address space in EID and RLOCs. Where EID stands for End-Host-Identifier, thus it consists in the IP address used to identify univocally the end-host, while RLOC stand for Routing LOCator, thus "locating" the attachment point (for routing purposes) of the EID. This separation of the IP space implies that there is the need to distribute and store mappings between identifiers and locators on caches placed on border routers. Based on Netflow [NETFLOW] traces, collected on the border router of UCL.be campus network, the cost of maintaining the locator/ID mapping caches has been estimated.

3.3.1.2 Experimental Setup

Figure 55 shows the setup used to collect the traces of our campus network. The network uses a full class B (i.e., /16) prefix block and is connected to the Internet through a border router that has a 1 Gigabit link toward the Belgian National Research Network (Belnet). Netflow provides a record for each flow, containing information like the timestamp of the connection establishment, the duration, the number of packets, and the amount of bytes transmitted.

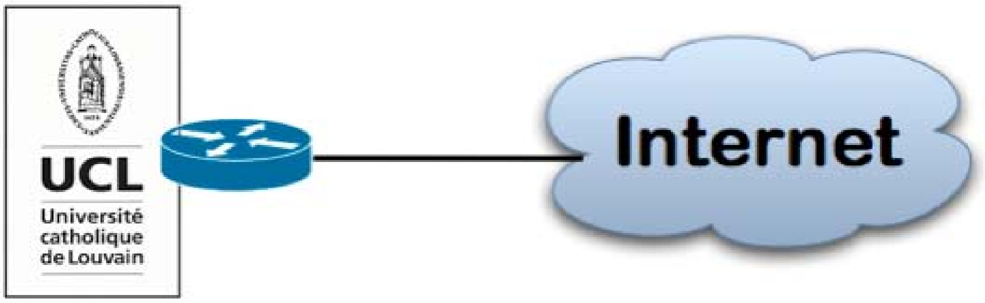


Figure 55 NetFlow traces collection setup

A two-step post-processing method is used to analyze the collected traces. In a first step we analyze traces using the flow-tools [FLOWT], then, in a second step, we use our own filtering software to refine the results. We developed a small software program able to emulate the behaviour of such a cache, which can be fed with the Netflow traces we collected. This allows us to evaluate various parameters (e.g., size, hits, misses, timeouts, etc) of the cache itself, but also to make some estimations of the lookup traffic.

In our analysis, we assume that the granularity of the EID-to-RLOC mapping is the prefix blocks assigned by RIRs. We call it /BGP granularity. In particular, we used the list of prefixes made available by the iPlane Project [IPLANE], containing around 240,000 entries. Using /BGP granularity means that each EID is first mapped on a /BGP prefix. The cache will thus contain /BGP to RLOC mappings.

Since we use /BGP as a granularity for EID-to-RLOC mappings, we first explored the characteristic of incoming and outgoing traffic using the number of correspondent prefixes as a metric. Figure 56 shows a one-day plot of the number of correspondent prefixes/minute, for both incoming and outgoing flows where the union of the prefixes represents the total number of correspondent prefixes. The union operation avoids counting twice prefixes toward/from which there are bidirectional flows.

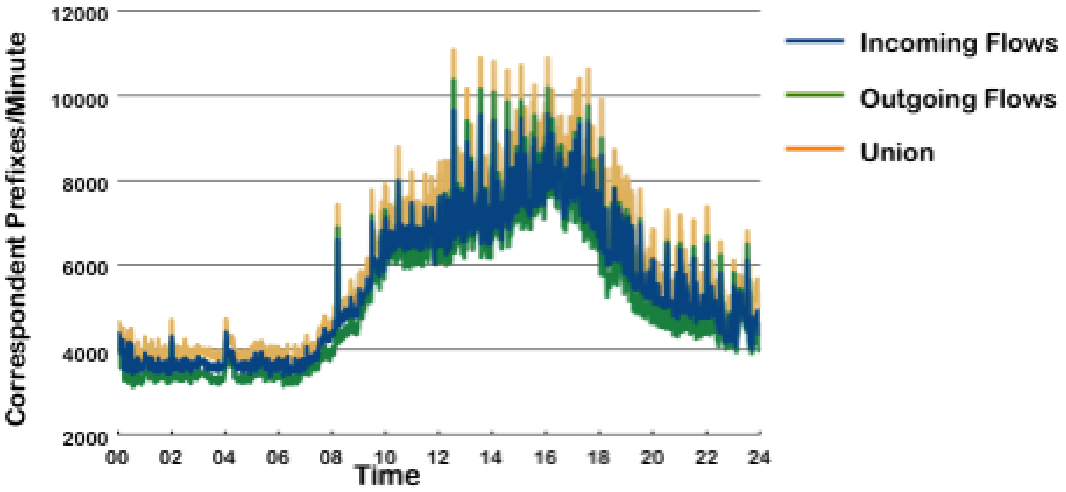


Figure 56 Correspondent prefixes daily snapshot

LISP creates an entry in its cache no matter the direction of the flow and whether it is uni- or bi-directional. This means that when the timeout of the entries is set to a value larger than one minute, the total number of correspondent prefixes, shown in Figure 56, represents the lower bound of the size of LISP's cache; hence, representing the minimum number of entries that are present.

3.3.1.3 Test Results

A timeout is associated to each LISP cache entry. Indeed, if an entry is not used for at least the time set in the timeout, the entry is considered expired and it is purged. We performed LISP cache emulations using three different values for the timeout, namely three (3), thirty (30), and three hundred (300) minutes. The evolution of the LISP cache size for the different values of the timeout is presented in Figure 57.

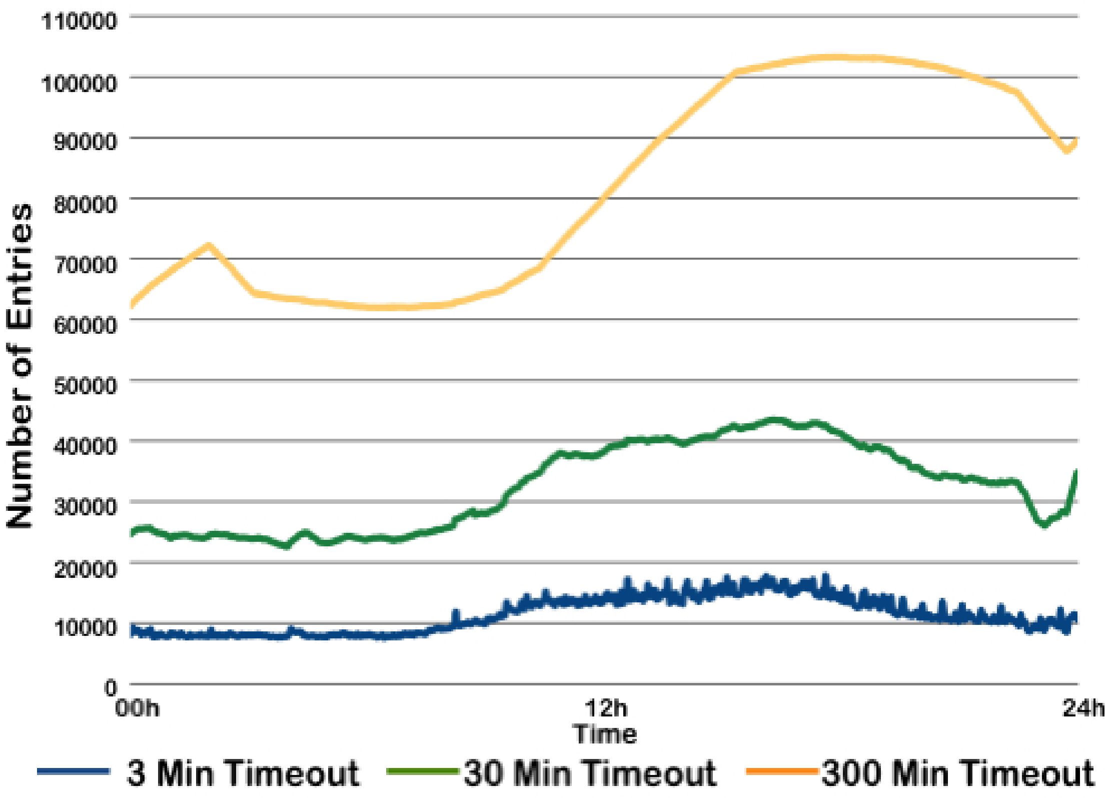


Figure 57 Daily evolution of the LISP cache size for different timeout values

The size of the cache is expressed in number of entries and follows the day/night cycle of the traffic. The range of this cycle for the cache size depends on the timeout value. When using a three minutes timeout, the number of entries ranges from around 7,500 during the night, up to around 18,000 during the day. This means that the size of the cache has an increase of 140% between night and day. In the case of a thirty minutes timeout, the number of entries ranges from roughly 22,500 during the night, up to around 43,500 during the day. It can be observed that, as expected since entries live longer, the average size of the cache is larger, while the size during the day is almost the double (93% actually) compared to the night period. This is not the case when using a three hundred minutes timeout. Indeed, the number of entries ranges from 62,000 up to 103,000, meaning an increase of around 60%. Thus, the longer the timeout value, the larger the average cache size and the smaller is the percentage of variation between night and day.

Figure 58 shows the hit ratio that is obtained with the above mentioned cache sizes. As can be observed, even for small cache sizes, with a three minutes timeout, the hit ratio seldom falls to values lower than 92%. This suggests that increasing the size of the cache by using large timeout value is not effective, since with a cache 10 times bigger we can gain only 8% at most.

Due to multi-homing, an EID (or a /BGP prefix as in our case) can be associated to more than one RLOC. Assuming that each set of EIDs (/BGP prefix) can be represented by 5 bytes, i.e. IP prefix and prefix length. Concerning RLOCs, we can consider that they have a size of 6 bytes. This because LISP associates to each RLOC (4 bytes IP address) two values: its Priority (one byte) and its Weight (one byte) [LISP07]. These parameters are supposed to be used for traffic engineering purposes. With these values we can estimate the size of the cache in bytes as $S = E \times (5 + N \times 6 + C)$. Where S is the size of the cache expressed in bytes, E is the number of entries, N the number of RLOCs per EID, and C represents the overhead in terms of bytes necessary to build the cache data structure.

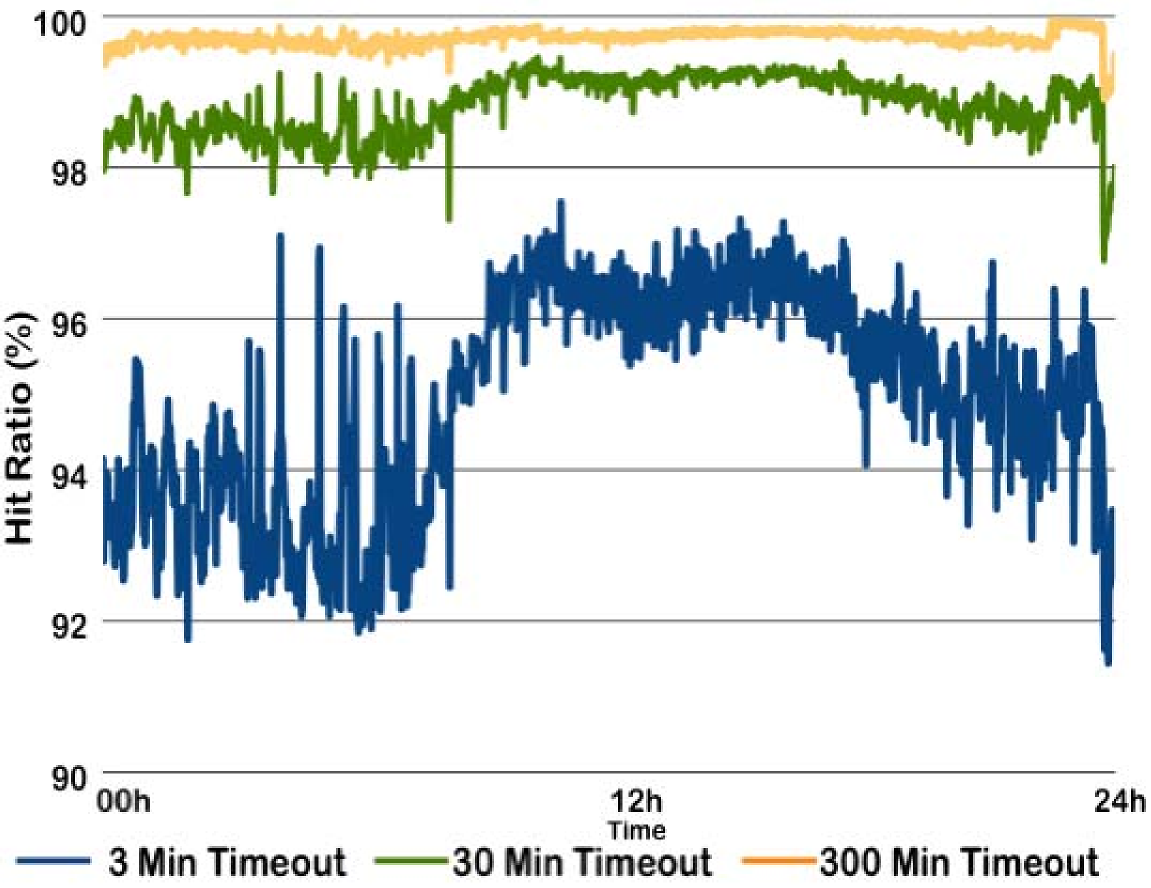


Figure 58 Hit ratio for mapping cache with three different timeout values

Timeout	Period	1 RLOC	2 RLOCs	3 RLOCs
3 Minutes	Night	139	183	227
	Day	334	440	545
30 Minutes	Night	417	550	681
	Day	807	1062	1317
300 Minutes	Night	1132	1490	1847
	Day	1917	2522	3127

Table 20 LISP cache size in Kbytes

Assuming the cache is organized as a tree, C can be set to 8 bytes, just the size of a pair of pointers.

Table 20 provides estimation, for both day and night periods and for all the three timeout values, of the cache size, expressed in KBytes, when for each /BGP prefix there are up to three RLOCs. Depending on the timeout value, the size of the cache can range from a bit more than a hundred KBytes, up to few MBytes.

In order to better understand the dynamics of the LISP cache, we plot in Figure 59 the Cumulative Distribution Function (CDF) of the entries' lifetime. Obviously the distribution is lower bounded by the timeout value, as it can be seen in the figure.

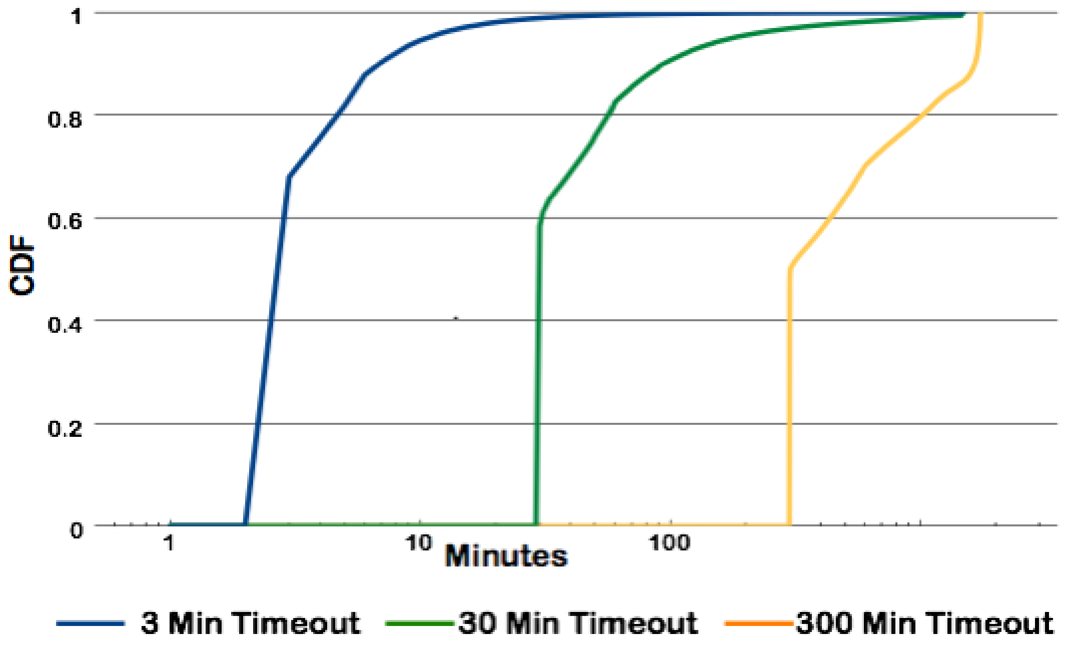


Figure 59 LISP cache entries lifetime

What is interesting to remark is that the large majority of the entries have a lifetime slightly higher than the timeout value. This means that most of the flows have a very small duration and are directed or come from prefixes that are not contacted so often. On the other hand, the distribution shows also that a small number of entries can have a lifetime as long as the whole period of observation, i.e., 24 hours. This does not mean that there are flows of such a length, but that there are a small number of prefixes that are contacted as often as the length of the timeout. This observation is also corroborated by the fact that, as it can be seen in the figure, the larger the timeout, the lower is the "knee" of the CDF. Meaning that more and more flows help to keep alive a larger number of cache entries.

The measurements described in the previous section concern parameters related to the LISP cache itself. Nevertheless, we are also able to estimate the number of lookups/minute our border router would issue. In the context of what is called a PULL model, a look up query to the mapping distribution system is issued whenever there is an outgoing flow for which there is no corresponding entry in the cache, i.e., there is no EID-to-RLOC map present for the destination prefix. Thus, counting the number of cache miss for outgoing flows gives us the estimation we look for.

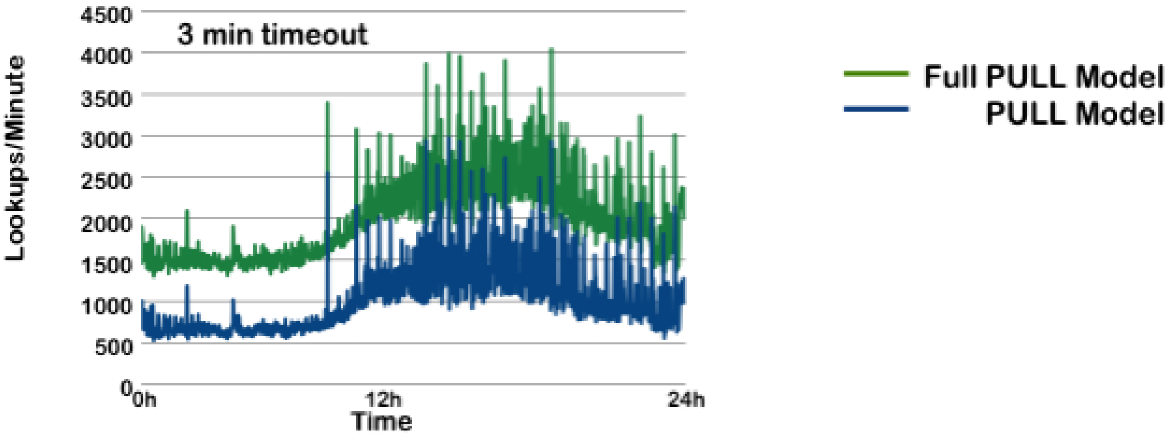


Figure 60 Number of mapping request per minute

Figure 60 shows the result of this counting for a daylong period for the 3 minutes timeout. Note that, insofar, we did not take into account incoming flows that generate a cache miss, since, in the context of LISP, there is no lookup query generated. Indeed, the mapping can be retrieved, by looking at the source address of the outer LISP header and the source address of the inner IP header. This, however, brings to light a limitation of the LISP proposal. Indeed, we are emulating a cache that has a /BGP granularity, while from incoming packet it can be retrieved only a single /32 EID-to-RLOC map. In order to populate the cache with the correct entries there other possible solution is to perform a mapping look up also for incoming flows. We call such a behaviour Full PULL model, and the results of the evaluation for the 3 minutes timeout are also presented in Figure 60. This solution means an increase of the number of queries of 150% during night period and to have spikes 30% higher during the day.

The locator/ID separation paradigm is based on tunnels set up between RLOCs, which introduce an overhead in terms of traffic volume. As a final evaluation, we measured this overhead, for both incoming and outgoing traffic, when LISP is used. Remark that the size of the prepended LISP header is the same for all the variants. Figure 61 shows a one daylong report of the volume of traffic expressed in Mbit/sec. Positive values are for outgoing traffic, while negative values for incoming traffic. As the figure shows, the overhead introduced by the tunnelling approach consists in few Mbit/sec. For outgoing traffic this means an overhead that ranges from 15% during the night down to 4% during the day. For incoming traffic this means an overhead that ranges from 10% during the night down to 2% during the day. Remark that this overhead does not depend on the mapping function or the mapping distribution protocol.

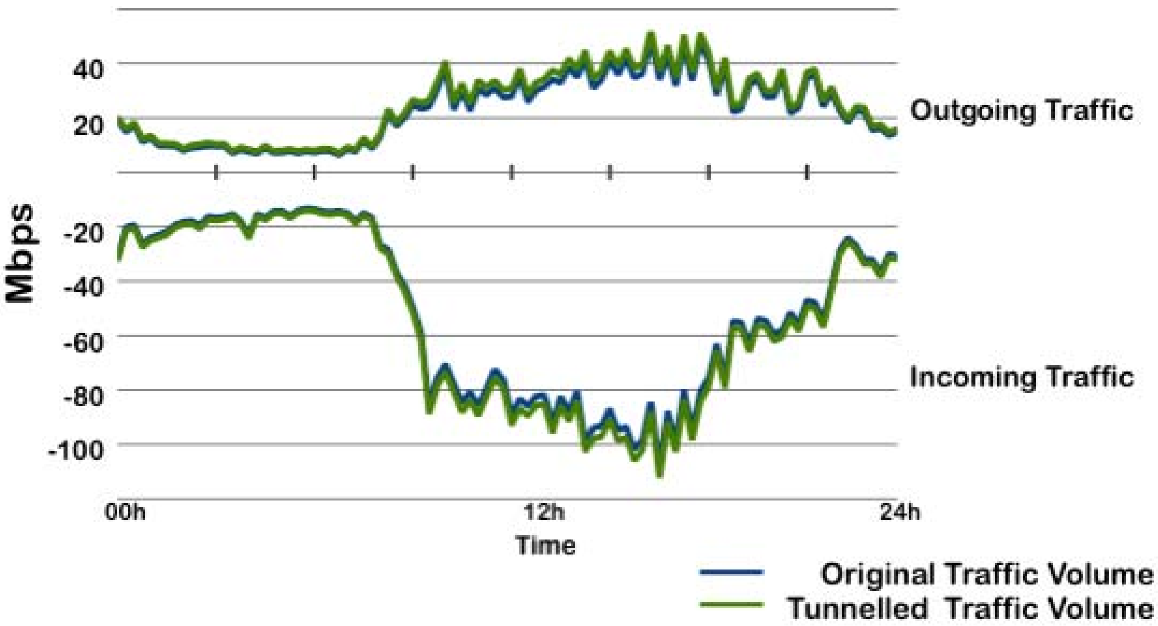


Figure 61 LISP tunnelling overhead

3.3.1.4 Conclusions

In the framework of the AGAVE Project we performed extensive evaluation of the LISP protocol, which we use as Tunnel Service. We based our analysis on real Netflow traces collected from UCL.be campus network. We fed the traces to a software program, emulating the behaviour of the LISP cache.

The analysis is done in the context of a PULL model, as suggested by LISP, but we extended it to a more general full PULL model. We observed that the size of the cache maintaining the mappings can be limited in size by using a relatively small timeout for the entries. Nevertheless, this increases the traffic generated for mapping lookups, which is never negligible. Maintaining large caches would reduce the amount of lookup traffic, however, a huge amount of flows have a short lifetime, thus a large number of cache entries are used to forward a small amount of traffic.

Finally, we also observed that the overhead introduced by the tunnelling, on which the locator/ID separation paradigm is based, does not pose any problem since it is quite small. Further details and results can be found in [IANN07].

3.3.2 LISP (Tunnel Service) Experimentation

3.3.2.1 Objectives

The LISP protocol has been implemented on a FreeBSD platform. The overall architecture of the implementation can be found in [OLISP] and in [D3.2]. This implementation has been first validated locally, i.e., by simply connecting two LISP enable machines and performing simple tests in order to verify and validate correct functioning of all implemented features. In the next section we give an overview on these implemented features and the software utility implemented aside the LISP protocol itself in order to monitor its correct behaviour. Then we describe some measurements we performed on the AGAVE integrated test-bed in order to verify the robustness of the implementation.

3.3.2.2 LISP Validation

LISP defines two different databases to store mappings between EID-Prefixes and RLOCs. The "LISP Cache" stores short-lived mappings in an on-demand fashion when new flows start. The "LISP Database" stores all the local mappings, i.e., all the mappings of the EID-Prefixes behind the router. For efficiency purposes, the two databases are merged together in a single radix tree data structure [TCPIP]. This allows having an efficient indexing structure for all the EID-Prefixes that need to be stored in the system. EID-Prefixes that are part of the LISP Database are marked by a MAPF_LOCAL flag, indicating that they are EID-Prefixes for which the mapping is owned locally. Thus, from a logical point of view the two "databases" are still separated. Actually there are two radix structures in the system, one for IPv4 EID-Prefixes and another for IPv6 EID-Prefixes.

In line with the UNIX philosophy and to give the possibility for future mapping distribution systems running in the user space to access the kernel's map tables a new type of socket, namely the "mapping sockets", has been defined. Mapping sockets are based on raw sockets in the new AF_MAP domain and are very similar to the well known routing sockets ([TCPIP][NETPROG]). A mapping socket is easily created in the following way:

```
      #include <net/maptables.h>
      int s = socket(PF_MAP, SOCK_RAW, 0);
```

Note that <net/maptables.h> is the header file containing all the useful data structures and definitions. Once a process has created a mapping socket, it can perform the following operations by sending messages across it:

- MAPM_ADD: used to add a mapping. The process writes the new mapping to the kernel and reads the result of the operation on the same socket.
- MAPM_DELETE: used to delete a mapping. It works in the same way as MAPM_ADD.
- MAPM_GET: used to retrieve a mapping. The process writes on the socket the request of a mapping for a specific EID and reads on the same socket the result of the query.

The messages sent across mapping socket for the above operations all use the same data structure, namely map_msghdr{}:

```
struct map_msghdr {                /* From maptables.h
                                    */
           u_short map_msglen;     /* to skip over non-understood
                                    * messages
                                    */
           u_char  map_version;    /* future binary compatibility
                                    */
           u_char  map_type;       /* message type */
           int     map_flags;      /* flags, incl. kern & message,
                                    *  e.g. DONE
                                    */
           int     map_addrs;      /* bitmask identifying sockaddrs
                                    * in msg
                                    */
           int     map_rloc_count; /* Number of rlocs appended to
                                    * the msg
................................. */
           pid_t   map_pid;        /* identify sender
                                    */
           int     map_seq;        /* for sender to identify action
                                    */
           int     map_errno;      /* why failed
                                    */
   };
```

The field map_type can be set only to the type listed above. The fields map_msglen, map_version, map_pid, map_seq, and map_errno have the same meaning and are used in the same way as for the rt_msghdr{} structure for routing sockets. Details about these fields and their use can be found in [TCPIP] and [OLISP].

The map tables and mapping sockets have been locally tested. In help of this task a small software utility has been developed in order to manage the map tables through mapping sockets. This software that we just called "**map**" allows not only to manipulate the mapping tables but also to monitor the activity through constant listening on a mapping socket. Since each change on the map tables or critical event is announced broadcast on all open mapping sockets. The map utility has the following syntax:

> **map** [**−n**] *command* [**−local**] [**−inet** | **−inet6**] *EID* [**−inet** | **−inet6**] *RLOC* [Priority Weight Rechability] [RLOC [Priority Weight Rechability]]

where *EID* is the address of the EID−Prefix (it can be also a full address), **−local** indicates if the mapping should be treated as part of the local mapping database or as part of the cache. Default is cache. The keyword **−inet** and **−inet6** are not optional, they must be used before any address (both EID and RLOC). *RLOC* is the address of the RLOC argument. The *EID* must be specified in the *net*/*bits* format. The values *Priority*, *Weight*, and *Reachability*; are optional. If not declared, the following default values are set:

- **Priority** 255 (Not usable)
- **Weight** 100
- **Reachability** 0 (not reachable)

It is not mandatory to declare all of them, but when declaring one, all the previous must be also declared. This means that to declare a weight the priority must also be declared; and to set the rechability to 1(reachable) both priority and weight must be declared.

For instance, to add in the cache a mapping having several RLOCs, the command is:

```
map add −inet 1.1.0.0/16 −inet 2.2.2.2 1 100 1 −inet 3.3.3.3 2
100 1 −inet 4.4.4.4 3 100 −inet 5.5.5.5
```

The above command associate to the *EID−Prefix* 1.1.0.0/16 the following RLOCs and related *Priority*, *Weight*, and *Reachability* values:

RLOC	Priority	Weight	Reachability
2.2.2.2	1	100	Reachable
3.3.3.3	2	100	Reachable
4.4.4.4	3	100	Unreachable
5.5.5.5	255	100	Unreachable

Table 21 RLOC, priority, weight and reachability association

The correct functioning of the map utility and map tables has also been verified through the mapstat utility. This software allows reading directly the kernel virtual memory and provides a dump of the map tables. Furthermore, the utility provides also some statistics concerning the LISP protocol, e.g., the number of encapsulated and decapsulated packets, drops, etc.

Table 22 shows the dump obtained with the mapstat utility on the UCL.be's LISP machine of the AGAVE Integrated test-bed. While in Table 23 there is an example of LISP statistics obtained on the same machine.

```
Mapping tables

Internet:
EID                Flags    Refs    # RLOC(s)
172.16             US          1    1 10.200.5.2          10   100 R
                                    2 10.200.5.65         20   100 R
                                    3 10.200.5.129        30   100 R
172.17             ULS         1    1 10.210.0.1           1   100 Ri
172.17.1/24        ULS         1    1 10.210.1.1           1   100 Ri
172.17.2/24        ULS         1    1 10.210.2.1           1   100 Ri
172.17.3/24        ULS         1    1 10.210.3.1           1   100 Ri
172.17.4/24        ULS         1    1 10.210.4.1           1   100 Ri
172.17.5/24        ULS         1    1 10.210.5.1           1   100 Ri
172.18             US          1    1 10.230.0.1          10   100 R
                                    2 10.230.1.1          20   100 R
                                    3 10.230.2.1          30   100 R
```

Table 22 Map table dump with mapstat

```
lisp:
        682025 datagrams received
        0 with incomplete header
        0 with bad encap header
        0 with bad data length field
        682025 delivered
        369234 datagrams output
        0 dropped on output
        369234 sent
```

Table 23 LISP statistics

3.3.2.3 Experimental Setup

In order to validate LISP in an inter-domain environment, we performed some measurements on the AGAVE Integrated test-bed (for details about the test-bed please see Section 4.1). From a logical viewpoint, the setup that we used is depicted in Figure 62.

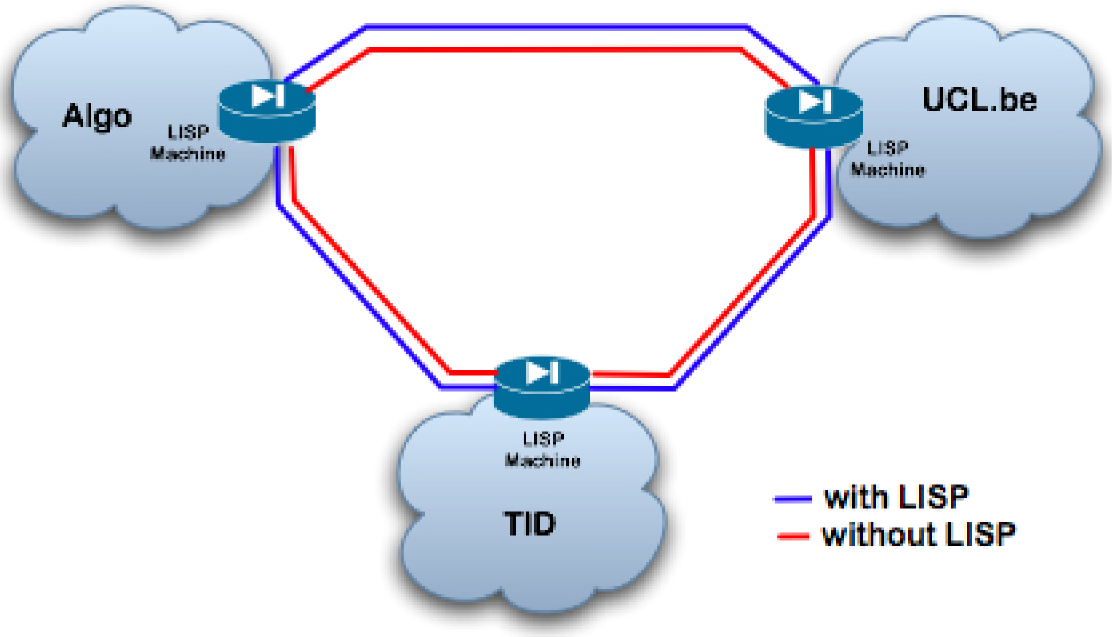


Figure 62 Logical view of the experimental setup for LISP test in the AGAVE Integrated test-bed

The test involves only the three LISP machines deployed on the integrated test-bed. Among these three machines there is a normal virtual link, consisting of GRE tunnels, where by *"normal"* we mean the fact that there is no LISP tunnelling. The endpoints of the links are:

- 10.200.5.2 – 10.210.0.1: between TID and UCL.be.
- 10.200.5.2 – 10.230.0.1: between TID and Algo.
- 10.210.0.1 – 10.230.0.1: between UCL.be and Algo.

On top of this three links LISP tunnelling has been used. When LISP is used the above listed addresses are the RLOCs of the following three different EID-Prefixes (in parenthesis there are the actual EIDs used for the experiment):

- 172.16/16 – TID (172.16.0.1)
- 172.17/16 – UCL.be (172.17.0.1)
- 172.18/16 – Algo (172.18.1.1)

The test consisted in running the `ping` utility to calculate the Round Trip Time (RTT) among each node with end without LISP encapsulation, at the same time. For more than 24h we collected the RTT. The main purpose was to evaluate the delay and jitter overhead introduced the encapsulation/decapsulation operations.

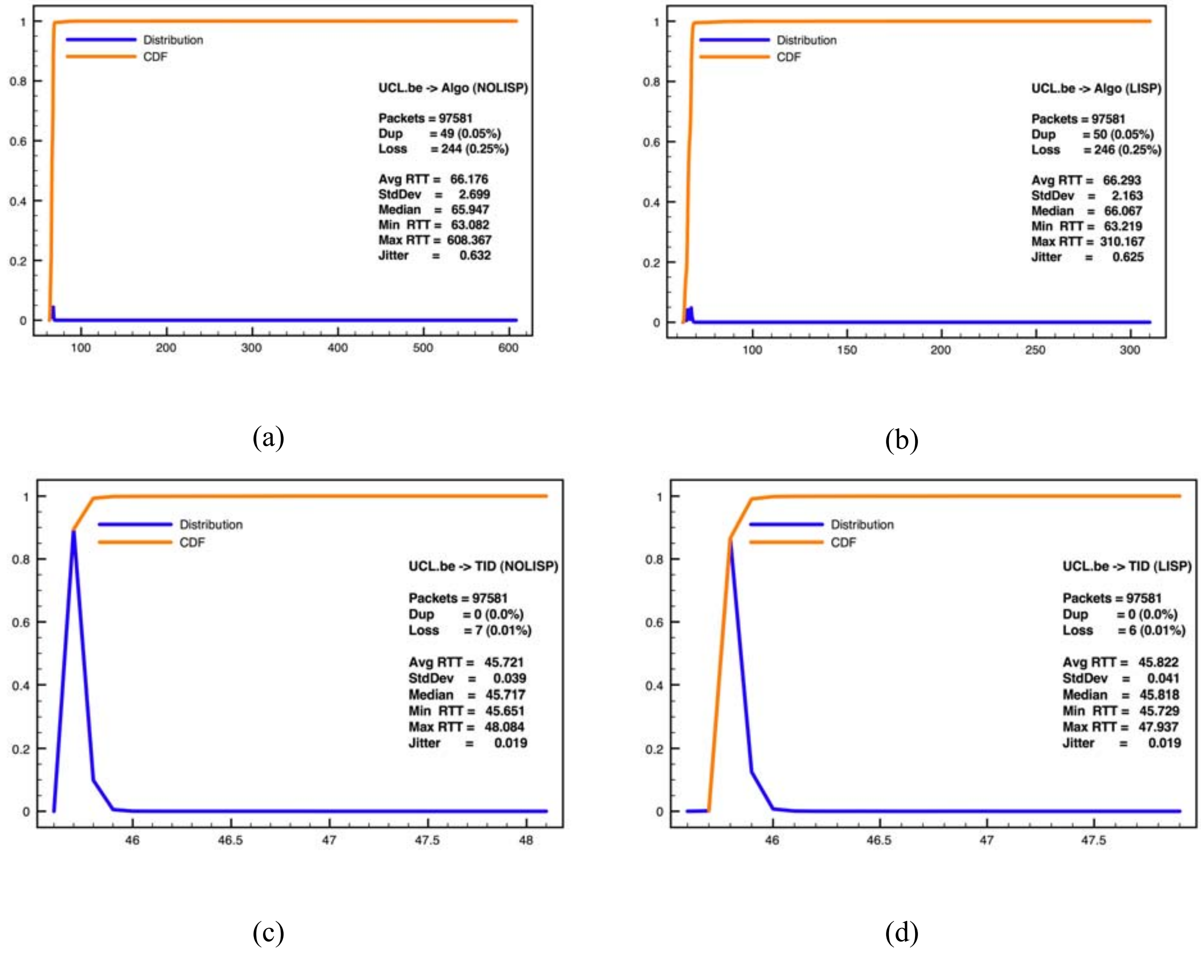
3.3.2.4 Tests Results

After collected the traces from the various partners we calculated the following statistics:

- Number of duplicated packets

- Number of packet loss
- Average RTT
- Standard Deviation
- Median
- Min RTT
- Max RTT
- Average Jitter

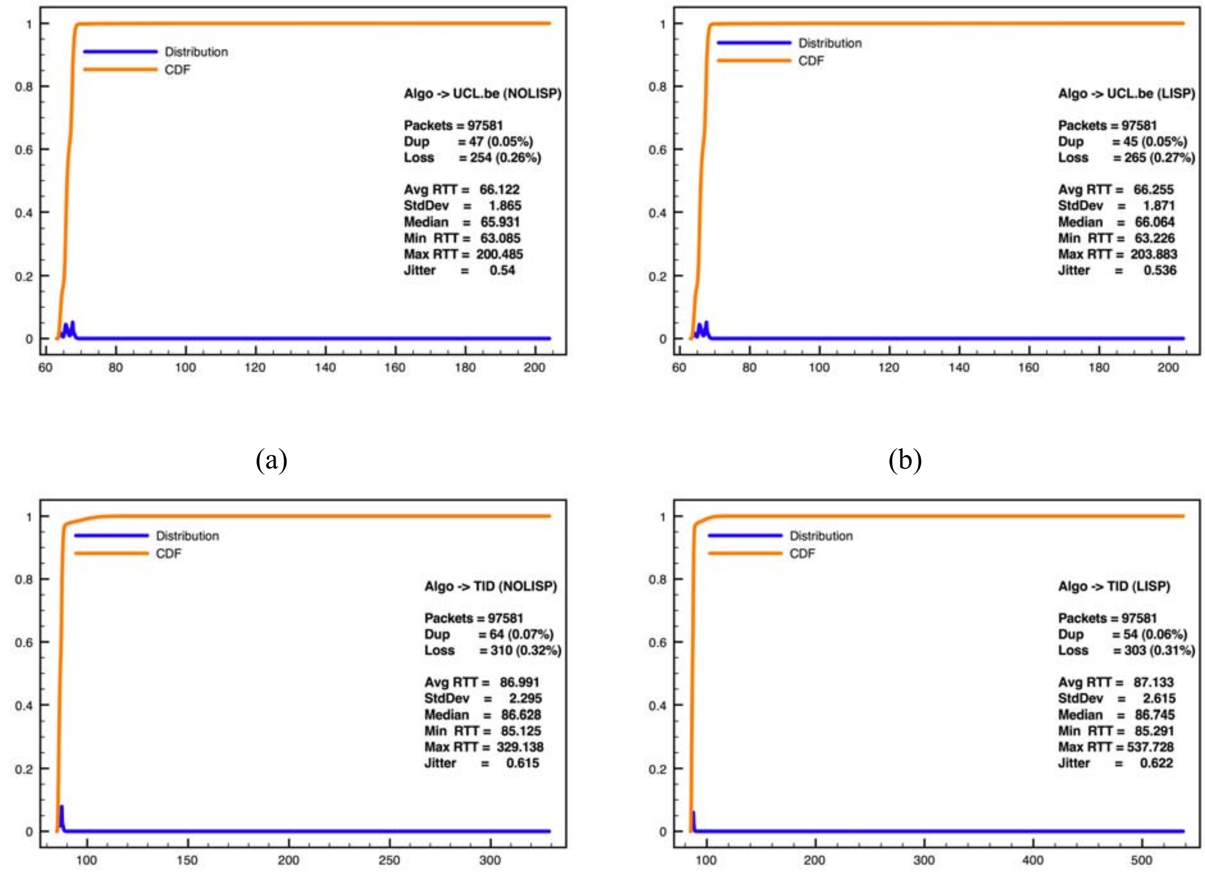
Note that even if calculated, the LISP encapsulation does not affects the number of duplicated packets and the number of packet loss, since these events are independent from the LISP operations. All the above statistics are depicted with distribution and CDF in the Figure 63 for UCL.be, in Figure 64 for Algo, and in Figure 65 for TID. As it can be seen, LISP operations introduce an increase in the RTT of around 0.1 milliseconds. Note that this value is specific to the AGAVE test-bed and the particular hardware that has been used. The main result to be kept is that LISP operation has no critical impact on the RTT, as can be also seen by the fact that distribution of the RTT with and without LISP is basically unchanged.



(a) (b)

(c) (d)

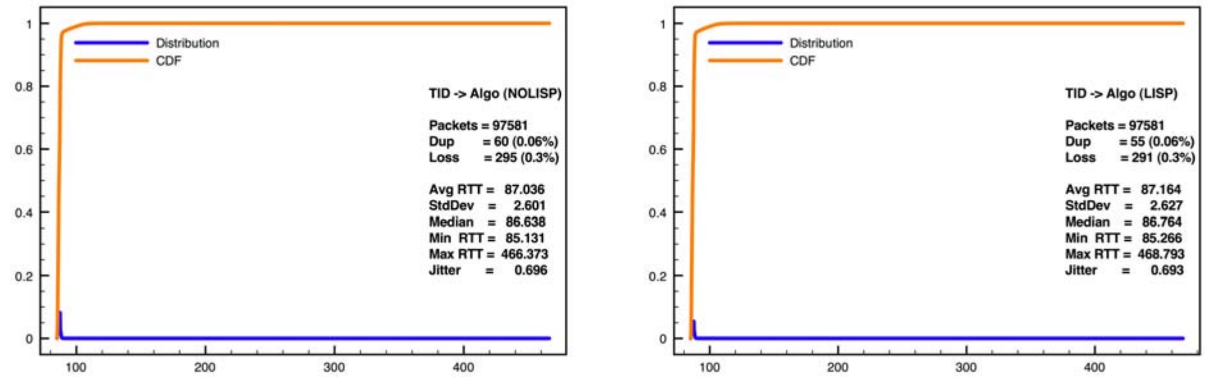
Figure 63 Ping test results from UCL.be



(a) (b)

(c) (d)

Figure 64 Ping test results from Algo



(c) (d)

Figure 65 Ping test results from TID

3.3.2.5 Conclusions

In the framework of the AGAVE Project a real implementation of the LISP protocol, on FreeBSD platform, has been carried out. LISP represents the Tunnel Service in the global AGAVE architecture.

This implementation has been first validated by verifying that all implemented features had the expected behaviour. Then, the prototype has been deployed on the AGAVE integrated test-bed, where other tests, in an inter-domain environment have been carried out. Here we presented some measurements concerning the additional delay that LISP can introduce, due to the encapsulation/decapsulation operations. It can be seen that this additional delay is very small, in the order of 1/10 of millisecond. Test concerning throughput have also been carried out but are presented later, in Section 3.3.4.2.

3.3.3 IDIPS (Tunnel Service Controller) Experimentation

As described in more details in [D3.2], IDIPS is a request/response service in which clients (e.g., LISP routers, P2P clients...) send path ranking requests to IDIPS servers. In the remaining of this document, the word *client* always refers to LISP border routers. IDIPS is a path ranking service where a client sends a list of sources, a list of destinations and optional performance criteria. These criteria determine the path selection algorithm IDIPS should use. If an IDIPS server replies, it gives an *ordered list* of <source IP, destination IP> pairs. A *rank* is associated to each pair and gives the relative interest of a particular pair compared to the other pairs in the list. The lower the ranking, the more attractive the pair is. It is worth to notice that pairs are ordered by rank such that the first entries are the most attractive while the lasts are the worst.

As in LISP, sources and destinations can be IPv4 or IPv6 addresses but also prefixes (where a full IP address is a particular case of an IP prefix). Working with prefixes offers the perspective to reduce the ranking problem space and allows caching.

As the path selection algorithm is centralized behind a single service (i.e., IDIPS) the collect of path information has no scaling factor whatever the number of clients (however, it is impossible to evaluate each possible (i.e., per IP) path individually).

As no path selection algorithm information is revealed to clients, IDIPS offers high flexibility. For instance, the path selection algorithm could be different for each class of client and vary with time. Indeed, changing the path selection algorithm is transparent for the clients (i.e., no additional client-side implementation). In addition, moving the complexity of the path selection out of the routers should avoid wasting router CPU cycles and even permit to implement more complex algorithms.

3.3.3.1 IDIPS in depth

Figure 66 shows the high level design of IDIPS. IDIPS is based on three modules/engines. The first engine, named *Path Information Collector* (PIC) is in charge of gathering information on paths (e.g., performances, policies, costs...) and is described in Section 3.3.3.1.1. Next, collected data must be efficiently represented and stored so that the processing cost of using it to rank paths is minimal. The engine in charge of representing and storing data is the *Knowledge Base* (KB - see Section 3.3.3.1.2). Finally, IDIPS must be able to combine information on paths so that it can rank them. This last engine is the *Decision Engine* (DE - see Section 3.3.3.1.3).

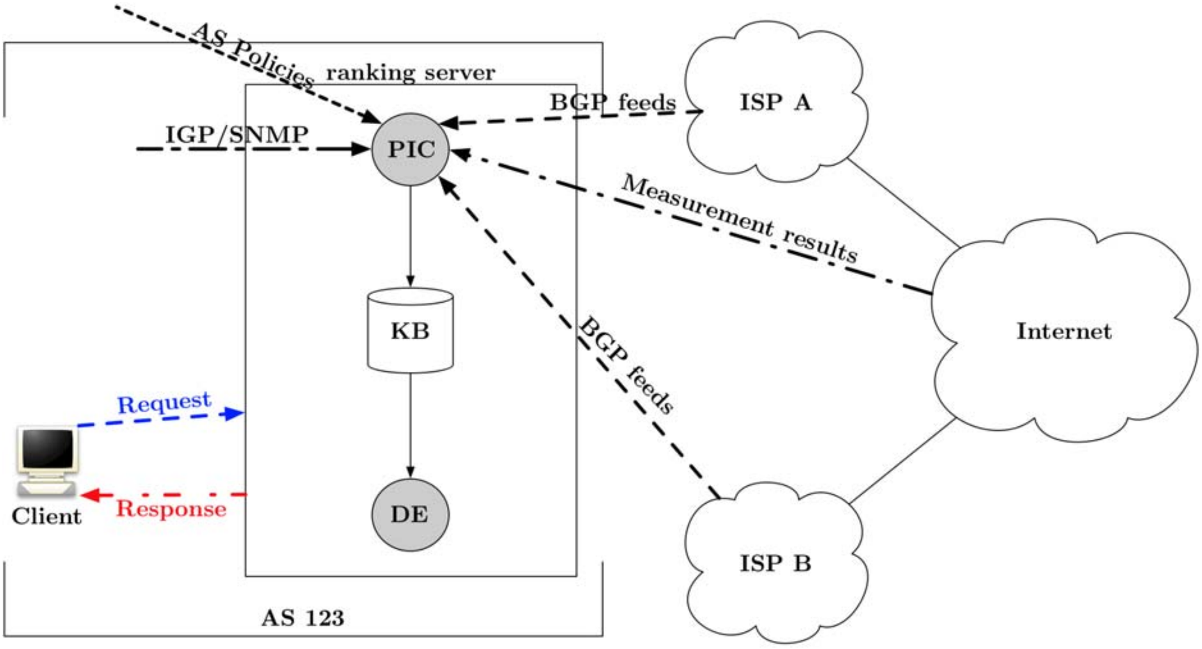


Figure 66 IDIPS overall design

3.3.3.1.1 Path Information Collector

As depicted in Figure 66, PIC collects paths information. Information can be of two types: (i) administrative information and (ii) measurements information. Administrative information characterizes paths with network policies (e.g., firewalls, billing, routers graceful shutdown...) but also routing information (e.g. BGP, IGP, static routes). On the contrary, measurement information characterizes paths with their intrinsic properties. These path properties could be collected with active measurements (e.g., latency, jitter, bandwidth, path diversity) or passive measurements (e.g., TCP trace analysis, traffic matrices, SNMP).

Administrative information characterizes paths based on high-level criteria. For instance, universities are often directly connected to the national research network and use a commercial ISP for accessing the Internet. Furthermore, research networks often replicate important FTP servers like GNU/Linux distributions. Thus, when upgrading GNU/Linux hosts, universities should prefer data from the research network rather than commercial FTPs servers. In that case, administrative information should be sufficient to take efficient decisions. However, for paths performance dependent applications like VoIP, administrative information is not sufficient and performances information must be taken into account. Thus, in this case, the best path will be the one meeting the administrative requirements and the path performances criteria.

In addition to metrics collection, the PIC translates the different metrics into *path attributes* (*attributes* for short). Attributes are a standardized representation of the metrics, independent of their nature. The simplest way to transform metrics in attributes is to convert them into integer values. This idea comes from the LocalPref attribute used by BGP [RFC1771], where complex metrics are summarized as an integer. Attributes value relations are transitive so that the comparison between different unrelated paths is made possible. That is, if A > B and B > C then A > C for a given attribute. IDIPS attributes have no direct meaning: a high attribute value for a path does not mean that this path is preferable to another with a lower value.

3.3.3.1.2 Knowledge Base

Once paths have been characterized, their attributes are stored in the KB. The KB might be seen as a database gathering all attributes of various paths. The KB must face two main challenges. First, it must be possible to get back any path attribute as quickly as possible. Second, given the potentially large number of paths and attributes in the KB, the KB must be as compact as possible.

3.3.3.1.3 Decision Engine

On one hand, the PIC and the KB model paths performances and properties. On the other hand, the DE compares the paths in order to select the best according to some criteria. To do so, the DE defines *Cost Functions* (CFs). A CF returns the cost of a <source IP, destination IP> pair (i.e., a path) for a given criterion. The cost is a numerical value characterizing a path according to one or more metrics. The cost must respect two constraints. First, the lower the cost, the better the path. Second, cost relations have to respect transitivity. As for attributes, transitivity is the key point of CFs as it allows one to estimate the cost of any path independently and then order them after wards. Transitivity allows caching and parallel computation of costs. Another important point of CFs is enabling combinations to create more complicated CFs.

To ranks paths, the DE calls the appropriate CF for each possible path to rank. It then creates the ranked paths list such that the best paths are those with the lowest cost and the worst with the highest. Paths in the returned list are grouped and the list is ordered by rank. The first group of paths in the list contains all the paths with the same lowest cost value. The second group contains those with the second lowest cost and so on. For instance, a possible ordering of the ranked paths A:1, B:2, C:1 and D:3 is (A:1,C:1,B:2,D:3). The ranking does not represent the absolute cost but the relative order of the paths with respect to their cost. For instance, if paths A, B, C and D have a cost of 1, 4, 1 and 7 respectively, the ranking value could be A:1, B:2, C:1, and D:3.

3.3.3.1.4 Path selection algorithms examples

In the following, we present two examples of path selection algorithms. First, we present how to select the path with the more promising bandwidth of at least 1.4Mbps. After, an example of billing is presented.

In the two examples, we assume that the function `update_prefix(src, dst, a, v)` tags path from `src` to `dst` with value `v` for attribute `a`. In addition, function `path_attributes(src, dst)` returns all the attributes of the path from `src` to `dst`.

The most promising bandwidth path selection algorithm is two-folds. First, the PIC part is responsible of estimating the available bandwidth of the paths and to update the bandwidth information stored. If available bandwidth from `src` to `dst` is 2Mbps, the wrapper first converts the available bandwidth into a simple numerical value. Let say that by convention the bandwidth is expressed in Kbps, the value becomes 2,000. If the attribute representing the available bandwidth is called ABW, the KB update is:

```
update_prefix(src, dst, 'ABW', 2000)
```

Second, the `available_bw_cf` CF is added and is such that paths with higher available bandwidth are preferred to those with less available bandwidth. Moreover, only paths with at least 1.4Mpbs available are considered as valid. The CF would be:

```
available_bw_cf(src, dst):int
begin
  attributes := path_attributes(src, dst)
  if(attributes{'ABW'} < 1400)
  begin
    return ERROR
  end
  return (MAX_BW -  attributes{'ABW'})
end
```

This function shows the typical architecture of a CF. First, path attributes are retrieved. Then cost is computed as a function of the attributes. In this example, MAX_BW is the capacity of the best possible path in the network (typically the capacity of the best link). The cost is build as the difference between the MAX_BW constant and the path available bandwidth because the cost must be the smallest for the path with the higher available bandwidth. By construction, the `available_bw_cf` CF is transitive as the available bandwidth attribute relations respect transitivity.

The previous example focuses on measurement information. The next example illustrates a path selection based on administrative information. It is common for providers to charge their clients on the 95th percentile. In this example, we suppose that the client is multi-homed, receives one IPv6 prefix per ISP and uses source routing. We also assume that link costs are already monitored. Then, if the cost of using ISP_A which provides prefix 2001:DB8::/48 is 1,500.00, the KB is updated by:

```
update_prefix(2001:DB8::/48,::/0,'COST',2)
```

If we suppose that the attribute representing the 95th charging cost is named COST and only remember the ceiling value of the cost in kilo dollars. We observe that the use of prefixes instead of addresses offers a simple way to represent a large variety of paths (all the possible paths from the ISP in this example). Let us call `minimize_cost_cf` the CF for that metric. It is implemented by:

```
minimize_cost_cf(src, dst):int
begin
  attributes := path_attributes(src, dst)
  return attributes{'COST'}
end
```

This CF is transitive and respects the minimize costs statement.

3.3.3.2 IDIPS Experimental Evaluation

In this section, we evaluate the performances of our IDIPS implementation. A specific test-bed has been created. On this test-bed, the IDIPS server is connected to a XORP BGP router. XORP is fed with 4 different RouteViews [ROUTE] RIBs, where the number of routes in the 2001, 2003, 2005 and 2007 RIB is 107k, 140k, 183k, and 244k respectively. The server runs on a FreeBSD 5.5 Pentium 4 2.60 GHz computer with 1GB of memory.

We consider a set of 100 clients contacting IDIPS. Each client periodically generates a list of uniform pseudo-random source and destination prefixes and requests the server for a classification. A Poisson distribution with a mean of one second gives the time interval between two requests, of a particular client. The number of source and destination addresses in the requests follows a uniform random distribution of between 0 and 16. We determine 95% confidence intervals for the mean of the server processing time based on the Student t distribution. These intervals are typically, though not in all cases, too tight to appear on the plots. We only focus on requests with at least one valid path, i.e., roughly 3,720,000 requests per test.

The cost function retained for the evaluation is the AS path length; the lower the AS path length the best. We chose this cost function, as its implementation should be representative of most of the common cost functions. Moreover, for randomly chosen prefixes, the BGP attributes can be very different from prefix to prefix, which gives good evaluation of the sorting/ranking/normalization algorithm performances. In the tests, the KB is filled before any query/response cycle.

Figure 67 shows the evolution of the server processing time (in ms, vertical axis) with the number of returned pairs (horizontal axis). The possible pairs are ordered according to the BGP decision process. Neither clients nor the server maintain caches and the time in buffers are considered as a part of the processing time.

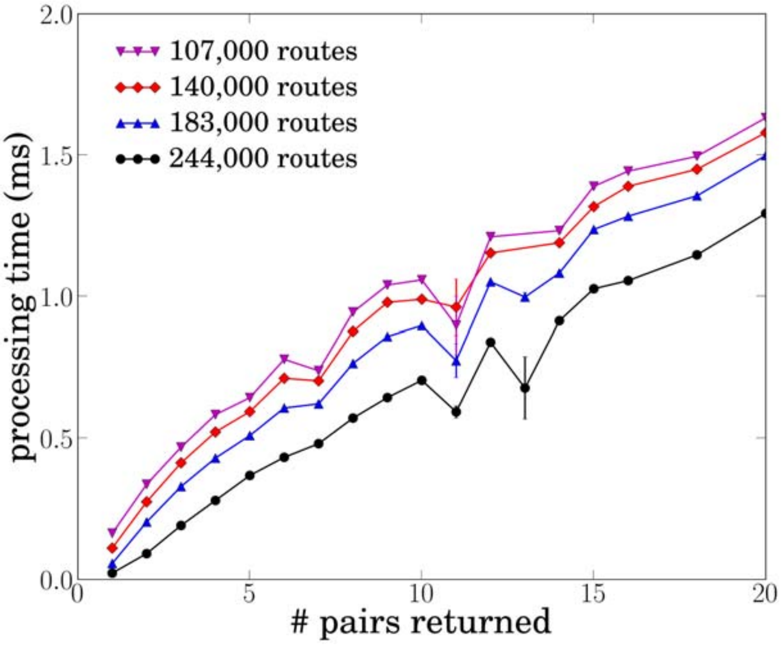


Figure 67 IDIPS processing time evaluation

From Figure 67, we see that the IDIPS server processing time oscillates between less than 0.1ms (when a single pair is returned to the client) and 1.6ms (when 20 pairs are returned). We do not plot results for larger values of pairs returned (the maximum in our tests would be 256, i.e., all the pairs formed by the sixteen sources and sixteen destinations) as it is not realistic. Indeed, in practice, it is very unlikely that an IDIPS server will return more than five or six pairs. However, we notice that the processing time required to return 140 pairs (for the largest number of routes) is 4.16ms. In our tests, the IDIPS server never returns the maximum numbers of pairs. The IDIPS server removes some invalid pairs from the returned list. From our tests, we notice that invalid paths are more frequent for small RIBs than for bigger. We finally notice that the performances of IDIPS do not decrease with BGP tables' growth. On the contrary, we can observe slightly better performances with the biggest BGP tables. Figure 67 shows that confidence intervals are very small which shows that processing time for a given number of returned pairs does not dramatically depends on the addresses involved in the pairs.

It is worth to notice that a list of 100 pairs has no real interest. Indeed, if the client must use the 100th entry in the list, it means that the previous 99 pairs are invalid, which should never occur. We would recommend limiting the number of returned pairs to 16. In such a situation, the processing time is 1.28ms on average. Therefore, from a client point of view, the cost associated to IDIPS is negligible, i.e., not more expensive than a DNS request.

3.3.4 IDIPS and LISP Integration: Test-Bed Experimentation

3.3.4.1 Objectives and Experimental Setup

In Section 3.3.3 there is an overview on how LISP and IDIPS can cooperate. Here, we show a practical experiment we performed on the AGAVE Integrated test-bed (for details about the test-bed please see Section 4.1), however, a logical view of the setup is shown in Figure 68. The test only involves two sub-networks but could be extended to the entire topology. The first sub-network, consisting in the Algo network, simulates a content delivery network. It provides video streaming and large file FTP distribution. The second sub-network, consisting in the UCL.be network, simulates an end-users network. The two networks are connected with LISP tunnels.

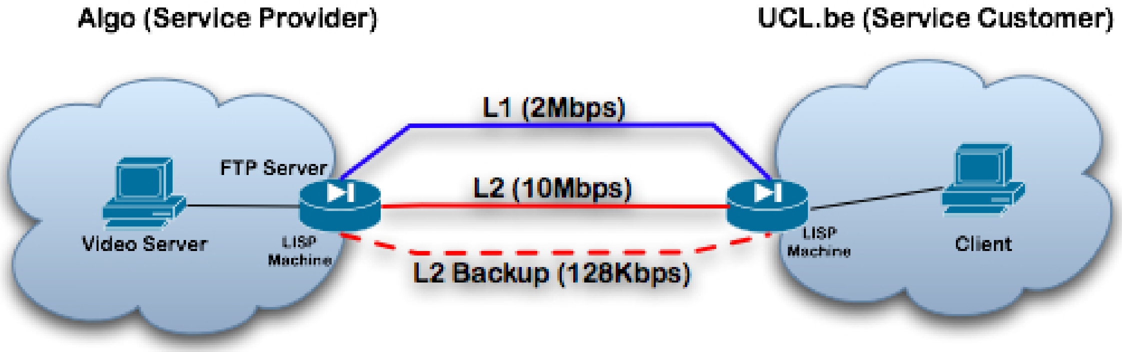


Figure 68 Logical view of the IDIPS + LISP test-bed setup

The two services in the Algo network are provided by the inetd FTP server and a VLC server. Clients at UCL.be subscribe for the two class traffic delivery. FTP is best-effort traffic. On the contrary, video streaming has strong QoS requirements: a video flow must have at least 1.4Mbps and jitter must be as lower as possible. When QoS is not ensured for video stream, the provider must pay the customer due to the non-respect of the QoS agreement.

Three links are simulated between Algo and UCL.be with LISP tunnels whose traffic is shaped using Dummynet [DUMMY]. A Dummynet rule defines the performance of each link. The first link, L1, is a peering link with a 2Mbps capacity. The two ends of the link are 10.230.0.1 and 10.210.4.1. The second link, L2, is a customer/provider link with a 10Mbps capacity. The two ends of L2 link are 10.230.0.1 and 10.210.5.1. Finally, the third is the backup link of L2 and has a capacity of 128Kbps. To simulate the rupture of link L2 and the recovery with the backup link, we just change the Dummynet rules for the <10.230.0.1, 10.210.5.1> pair.

The validation test performed is multi-fold. First, the two links are working properly. Secondly, L2 link breaks down and all its traffic is diverted to the backup link (i.e., change bandwidth rule in Dummynet). Third, all the traffic is diverted to L1. Unfortunately, in such a case the quality of service is not ensured for the video and, finally, best-effort traffic is diverted to the backup link in order to achieve QoS for the video.

3.3.4.2 IDIPS and LISP at Work: Experimental Results

The integration of IDIPS in LISP is straightforward. The only one thing to do is to write a wrapper that translates IDIPS ranking in LISP priorities and update the mapping table with the new priorities. To update the mappings, wrappers use the normal mapping updater provided by the LISP implementation. In the case of LISP, the wrapper uses the mapping socket ([OLISP], [D3.2]).

The implementation works as follow. When a source EID and a destination EID are selected, the corresponding lists of source RLOCs and destination RLOCs are sent to the IDIPS server with the performance criteria. The server then determines the rank of all the possible paths and sends back the ranked list to the wrapper. The wrapper then sets the priority of each RLOC of the EIDs to the ranking value.

In this evaluation, we go step by step to show to potential of using IDIPS with LISP. However, in a real environment, the traffic should directly be diverted in an efficient and cost-effective way.

Two flows are involved in the test. The first flow is the download of a file with FTP and the second is a video streaming. The stream is simulated with Iperf [IPERF], which continuously sends 700 bytes long UDP segments at a rate of 1.7Mbps. Figure 69 shows the dynamic of the flows at the different steps. The x-axis is the normalized time and the y-axis the bandwidth. Initially, the FTP flow is carried by L1 and the video is carried by L2.

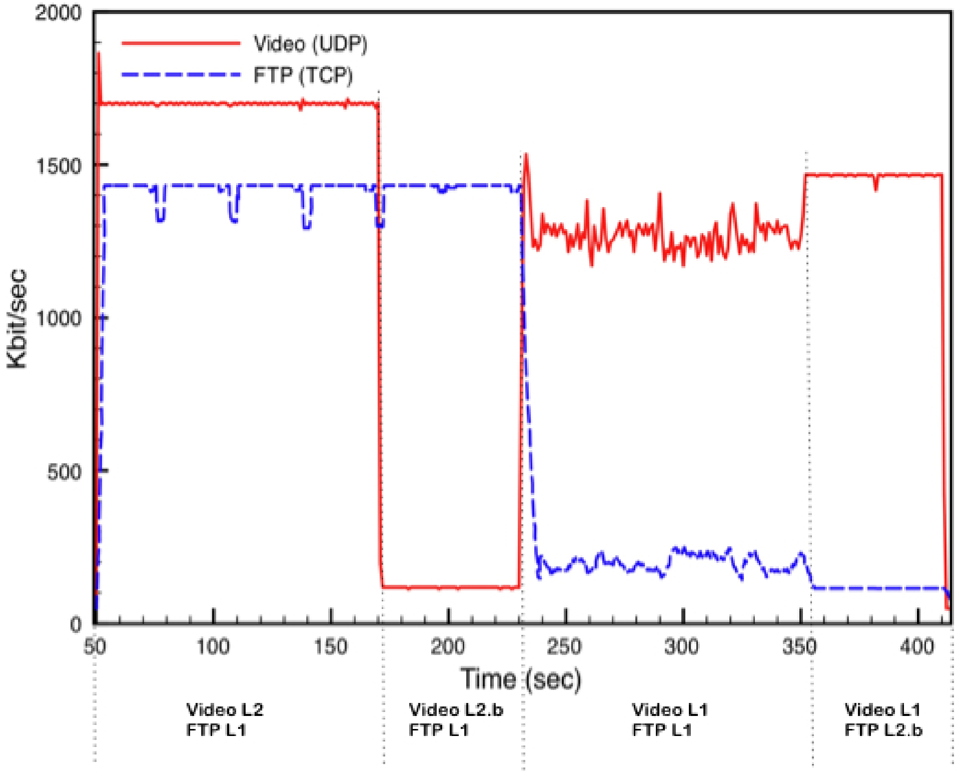


Figure 69 IDIPS+LISP Validation test.

First step: every things ok

Between time 50 and 170 in Figure 69, we observe that the two flows are working as expected. We see that the video as a limited jitter and has enough bandwidth (1.7Mbps). The FTP flow is stable and uses approximately all the available bandwidth (on L1).

Second step: L2 link breaks down

At time 170, L2 breaks down, thus BGP redirects the video to the backup link. The video flow bandwidth falls down to around 100Kbps, which is not sufficient to ensure QoS (1.4Mpbs is required to ensure QoS agreement). FTP traffic is not affected by the failure as it is carried by L1. Up to this point, IDIPS has not yet been used and LISP tunnels carry the flows.

Third step: move all the traffic on L1

This step is fictive and is only used to illustrate the flexibility of IDIPS cost functions. From time 235 to 355, the operator only wants to use links with the maximum capacity and configures IDIPS in order to achieve this requirement. All the traffic goes to L1 as it has the best capacity. The TCP throughput of the FTP flow falls down to around 200Kbps while the video obtains 1.3Mbps. Unfortunately, the link experiences congestion because the streaming server continues to send video at 1.7Mbps. The result is an important jitter for the two flows and important performances degradation for the TCP flow. If jitter is not an issue for best-effort FTP flow, it is really bad for a video. The video QoS is no more ensured and the customer observes service degradation. Video flow degradation has a cost for the provider, as he does not respect the QoS agreement with the customer.

Fourth step: ensure QoS for video

In this last step, the operator takes performances criteria into account as well as real cost. Video stream uses a link with at least 1.4Mbps and low jitter. FTP flow is moved such that the operational cost is minimized. Without QoS agreement, the best choice would the one proposed in the previous step, as L1 is a peering link. However, the provider has to pay the customer when QoS is not respected for video and thus the overall cost of moving FTP traffic on the backup link and video on L1 is lower than moving the two flows on L1.

To move traffic, IDIPS advertises the mapping service to update the priorities of RLOCs in order to divert video to L1 (i.e., give more priority to RLOC 10.210.4.1 for video) and FTP to L2 (i.e., give more priority to RLOC 10.210.5.1 for FTP).

The mapping is updated at time 355 and video obtains a 1.45Mbps rate with a limited jitter and thus agrees with the QoS requirements. TCP throughput falls to 100Kbps but has low jitter.

3.3.4.3 Conclusions

The proof of concept shows that using IDIPS with LISP offers strong perspective for operators as it allows them to modify paths followed by IP flows according to technical but also economical considerations. This ability permits the operators to be ready for future requirements.

Figure 69 also shows an important consequence of LISP tunnelling. As identifiers remain constant all along the steps, TCP connections are never broken and the modification remains transparent for the applications.

In this section we see that combining LISP and a TSC offers strong perspective for operators and customer. We first present IDIPS, a TSC service based on ranked lit of <source, destination> pairs.

We see that IDIPS works with arbitrary attributes, which are used to reflect a network property (e.g., performance, cost, administrative policies). To implement the path selection algorithm IDIPS uses cost functions that get one or more attributes of the source and of the destination, and combines them to rank the <source, destination> pair. Ranking algorithm is only limited by the attributes stored for the sources and the destinations.

In addition, we show that using IDIPS as a TSC is straightforward: convert rank into LISP mapping priorities.

We also highlight that separating the routing, the mapping and the path selection algorithm offers strong perspective for operators (i.e., flexibility, costs, robustness) and for the customer (i.e., performances, costs).

3.4 q-BGP

3.4.1 Objectives

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 investigate the effect of various parameters, algorithms and configurations in a range of scenarios including adaptive policies based on injecting dynamically monitored QoS attributes into BGP UPDATE messages.

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.

3.4.2 Experimental Setup

An overview of the simulation process is as follows. 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 as a series of trigger events. When such a trigger event occurs the simulator repeatedly performs 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 may execute incoming q-BGP message filtering and decision processes and possibly send out new q-BGP messages for those routes that have changed within an epoch; messages which will be acted on by the adjacent ASes in the next epoch. The state of the entire network is stored after each simulation epoch so it can 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).

qBGPSim is the main program that performs the simulation of events that are specified in an input file. Power-law compliant AS topologies generated by BRITE [BRITE] were used. A series of other programs perform ancillary functions and are used to generate events and other input data, specifically the IP prefix distribution (which subnets are assigned to which ASes), 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) assigned to ASes 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. The generator 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 precise distribution tree that was used to create it; other aggregation paths will either not work, or will be excessively general.

DemandFactory: This 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 NIAs with neighbouring INPs have already been agreed. It isn't enough to just randomly assign capacities to inter-domain links, as this capacity won't be in "useful" locations. It is essential that this is approximately correct as it is really this inter-domain capacity that is being optimised. Rather than implementing an entire AGAVE 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, a list of IP network prefixes and the ASes 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.

In these experiments we concentrate on two QoS Attributes (QAs):

- *One-Way Delay (DELAYQA) QoS Attribute:* The expected time for a packet to reach the prefix advertised. When traversing ASs and inter-domain links this value is formed through the concatenation of the various delay contributors:

  ```
  Advertised DELAYQA = incoming advertisement DELAYQA + local NP delay +
  NIA queuing delay;
  ```

  Where local NP delay is the edge-to-edge delay across the current AS and NIA queuing delay is that introduced over the inter-domain link with the AS from which the advertisement was received.

- *Bandwidth (BWQA) QoS Attribute:* The bandwidth available to the prefix specified in the NLRI field. Local QoS classes are assumed to have sufficient bandwidth assigned[1] so the only restriction is the NIA capacity, thus the value advertised becomes:

  ```
  Advertised BWQA = min (incoming advertisement BWQA, offered NIA
  capacity);
  ```

  Where offered NIA capacity is a portion of the bandwidth allocated to the PI on the inter-domain link with the AS from which the advertisement was received.

Throughout the experiments we examine a number of route selection policies, which make use of various combinations of QoS attributes. For added variability of policies we also use a QoS attribute equivalence margin. This margin effectively introduces a comparison granularity to QoS attributes.

In the following simulation runs QA equivalence is calculated by:

```
if( floor( MessageA_QA / QAmargin ) = floor( MessageB_QA / QAmargin ) )
```

[1] NIAs between ASs imply that each AS undertakes to engineer its local resources to honour the QoS and traffic quantity clauses for the PI.

then the incoming message QAs (MessageA_QA and MessageB_QA) are considered equivalent and the decision must be performed on the next metric. For example, applying a one-way delay equivalence margin of 70ms to the example in Table 24, R2 and R3 are equivalent with regards to one-way delay. With a margin of 90ms, R1, R2 and R3 are equivalent.

Route	One-way delay	One-way packet loss rate
R1	150ms	5%
R2	120ms	2%
R3	100ms	3%
R4	200ms	8%

Table 24 Example q-BGP QoS attribute values

The route selection processes examined here are explained below. Note that we are investigating the performance of the dynamic q-BGP route selection process here rather than investigating an administratively-set routing policy, therefore LOCAL_PREF values are assumed to be the same for all routes.

- PI identifier (PIID) only. Routing decisions within the PI are based first on AS Path length, and use ASN (AS number) as a tie-breaker. Given that the simulations focus on a single PI these tests are effectively without any additional QoS information injected into q-BGP and are therefore equivalent to classical BGP. This policy is denoted as PIIDONLY.
- Single QoS attributes of either DELAYQA-only or BWQA-only. The routing decision is performed based on the QoS attribute first (within the boundaries of the QAmargin), and then on AS path length and ASN. These policies are denoted as DELAYQAONLY and BWQAONLY.
- A two level priority scheme where, depending on the priorities specified in the policy, either one of DELAYQA or BWQA is checked first, and then if found equivalent (depending on the bandwidth (BW) and one-way-delay (DEL) parameters and margins) the other QA is checked. If that, too, is equivalent the decision is the based on AS path length and the ASN. These policies are denoted as PRI_BW-*s*_DEL-*t* for when BWQA has a higher priority than DELAYQA, and PRI_DEL-*t*_BW-*s* when DELAYQA has a higher priority than BWQA. The *s* denotes BWQA equivalence margin parameter and *t* the DELAYQA equivalence margin.

3.4.3 Test Results

3.4.3.1 Performance tests

The experiments here were all performed on network topologies of 100 ASs with an average connectivity degree of four unidirectional links. Each set of parameters was repeated 16 times and the results averaged. Error bars are derived from the standard deviation of the mean for each simulation run and not each individual metric. i.e. the error bars on NIA utilisation are the standard deviation of the mean NIA utilisation for each network and not the mean of the standard deviations for all NIAs within each network.

The first set of performance tests demonstrates the effect of varying QA equivalence margins.

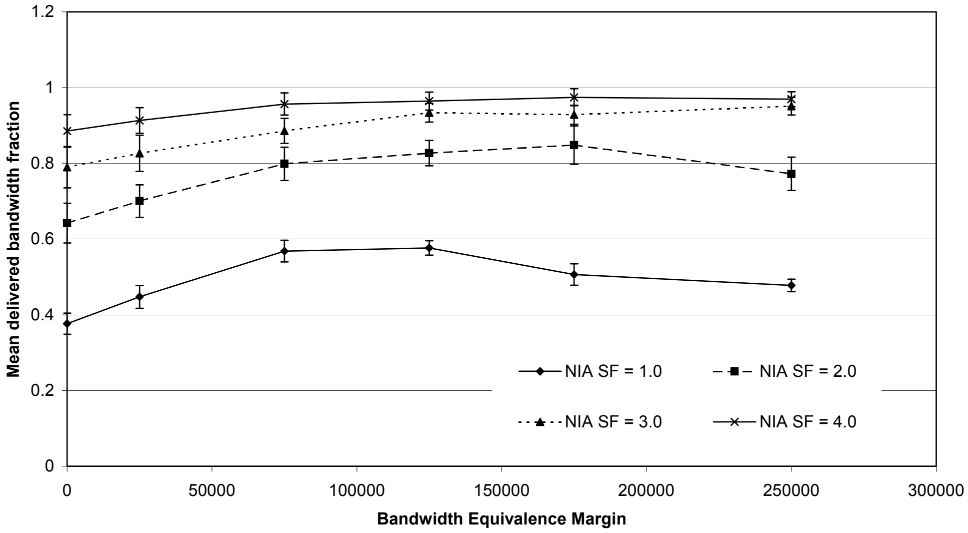


Figure 70 Mean delivered bandwidth fraction for a range of bandwidth equivalence margins under the BWQA-only policy

In Figure 70 we can see that when resources are scarce (NIA SF = 1.0) the delivered bandwidth is low for no margin but as the margin increases the fraction of delivered (to offered) bandwidth improves. This continues until the margin is so large that the majority of route selections then fall to AS path length where the delivered fraction becomes worse again.

We hypothesise that the cause of the poor performance of BWQA-only with small equivalence margins is the convergence of routing paths towards the areas of high capacity resulting in the saturation of those links. As the QA values are static and administratively set they do not change to reflect this saturation and the overall throughput for demands suffers. We refer to this phenomenon as the “QA rush”.

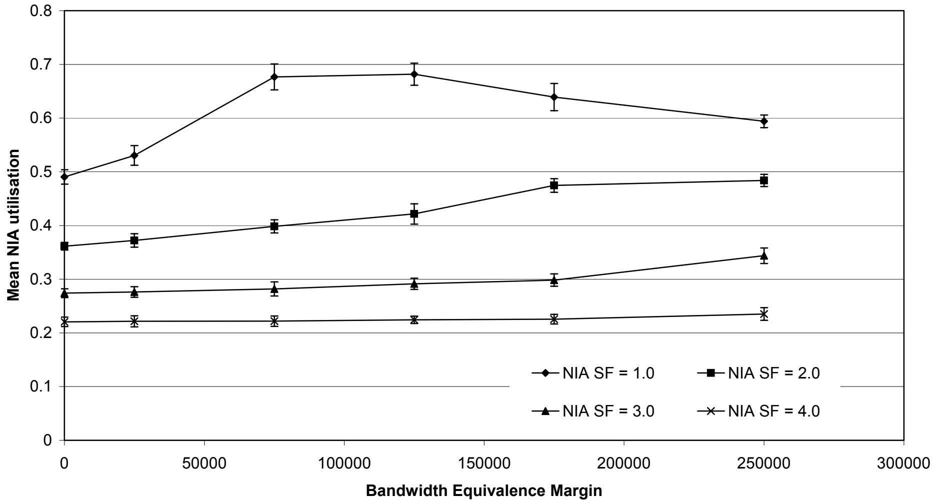


Figure 71 Mean NIA utilisation for a range of bandwidth equivalence margins under the BWQA-only policy

As the equivalence margin increases route selection is no made just on the highest capacity route. Several routes that fall within the margin are considered equivalent and the route is then selected based on the AS path length, so introducing some load distribution over the set of high capacity routes. This can be seen in Figure 71 as an increase in the average NIA utilisation as more of the demand gets through the bottlenecks and the network load is better distributed across the network.

Figure 72 shows a scatter plot of mean delivered delay against mean delivered bandwidth fraction for a range of q-BGP route selection policies. The results are shown for three NIA scaling factors of 1.5, 2.0 and 2.5—the three points that run from left to right on each of the curves. PIIDONLY (1) is a policy based purely on PI id (effectively a single instance of classical BGP, like the current Internet), BWQAONLY (2,3) is a policy based only on a bandwidth QoS attribute, DELAYQAONLY (4,5) is a policy based purely on a delay QoS attribute, and PRI_DEL-*_BW-* (6) is a policy where route selection is based first on delay, and then bandwidth, and PRI_BW-*_DEL-* (7,8) is where selection is first on bandwidth, then delay. The numbers in the policy names denote the equivalence margin size.

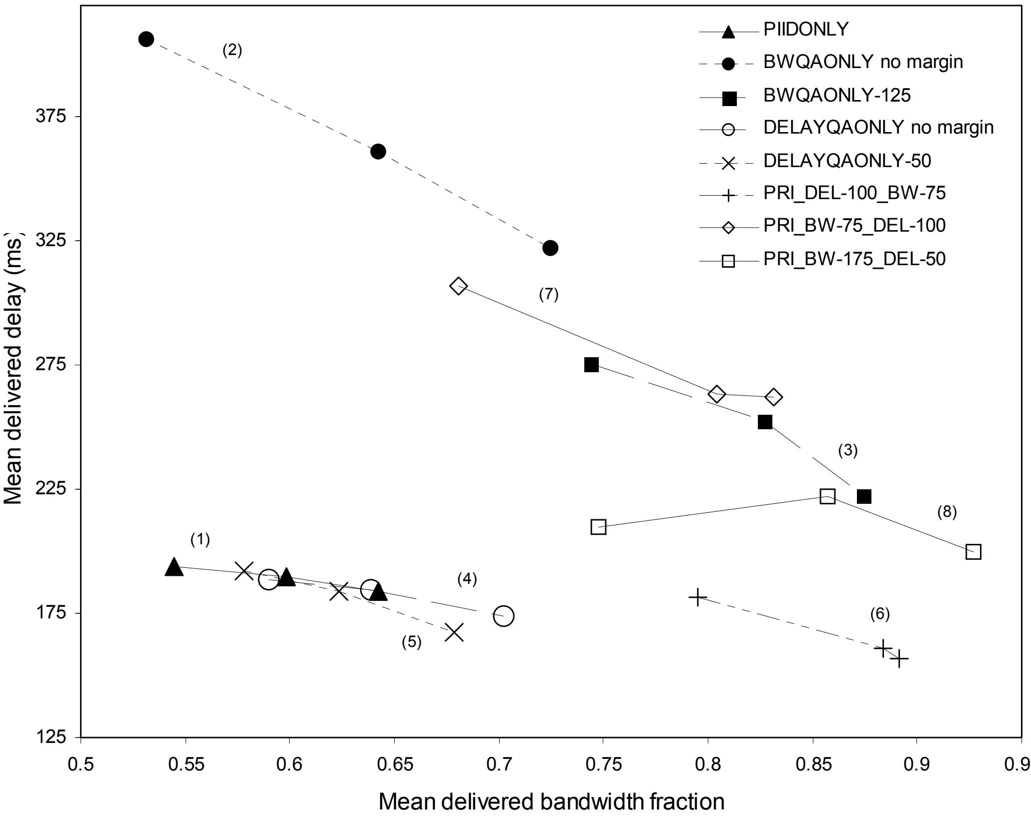


Figure 72 The effect of q-BGP selection policy on delivered delay and delivered bandwidth fraction

Single QoS Attribute q-BGP policy: Bandwidth

The policy of selection based on BWQA-only with no equivalence margin (2) delivers higher bandwidth fractions than PIID-only for higher NIA scaling factors, but performs worse than PIID-only in congested networks, i.e. low NIA scaling factor. The reason for the latter is due to a phenomenon previously described above as *QA-rush:* where resources (ASs, inter-domain links) with the highest advertised QoS (e.g. high capacity, or low delay) are selected more often than those with inferior QoS attribute values, resulting in a greater load on those resources, potentially saturating them and increasing delivered queuing delays and packet loss.

In all cases the adoption of the BWQA-only policy shows worse delivered delay than PIID-only, since it selects the largest capacity route at any cost, causing increased congestion and greater queuing delays. As the QA values are static and administratively set they do not change to reflect this saturation, and consequently the overall performance suffers.

By adding a margin of equivalence, e.g. of 125 bandwidth units as shown for the BWQAONLY-125 (3) curve, the performance is improved in terms of both delivered delay and bandwidth when compared to selection based on the absolute widest path (2). This also out-performs PIID-only (1) in terms of delivered bandwidth fraction but not delay. The policy of using an equivalence margin improves performance by increasing the number of equivalent bandwidth paths and allowing route selection within the set of best bandwidth paths to be done on the basis of AS path-length, thereby adding diversity to the overall routing behaviour.

Single QoS Attribute q-BGP policy: Delay

The policy of selection based on DELAYQA-only (4) shows negligible improvement over selection based on shortest AS path (PIID-only) in terms of delay and only marginal improvement in terms of delivered bandwidth. One of the reasons for this is that in the simulation scenarios – as in the real world – the shortest AS path is often the one with shortest delay. If the simulated inter-AS topology were selected carefully so that the ASs along shortest path routes had large local QoS-class delays then a more marked improvement in performance of the DELAYQA-only selection policy might be observed. This is also why there is little difference in results of DELAYQA-only with and without an equivalence margin: (4) and (5).

Two level priority q-BGP policy

The best performing route selection policies are those that select paths according to both advertised delay *and* bandwidth. PRI_DEL-100_BW-75 (6) is first of all selecting paths on the grounds of smallest advertised delay, with a margin of equivalence of 100 ms, and subsequently selecting between these on the basis of widest advertised bandwidth with a margin of equivalence of 75 bandwidth units, falling back on AS path length and finally AS number if a tie breaker is required. This policy delivers the best overall performance in terms of bandwidth and delay at all three NIA scaling factors.

It is interesting to compare PRI_DEL-100_BW-75 (6) to PRI_BW-75_DEL-100 (7) – i.e. the same bandwidth and delay margins, but with the priority reversed. In the latter (7) case both delivered bandwidth and delay is worse than the former (6) and worse than selection based on BWQA-only with a wider margin of equivalence (3). This is again caused by the QA-rush, and the rush is alleviated by increasing the BWQA equivalence margin (8).

It can be seen that with different margins of equivalence, a selection policy with the same priority order of QoS attributes can deliver significantly improved delay/bandwidth performance. This can be seen by comparing PRI_BW-75_DEL-100 with PRI_BW-175_DEL-50. It appears, therefore, that it is better for the path selection process not to be too narrow in its choice of the set of best paths on the highest priority QoS attribute so that more potential paths are passed to the selection step based on the 2nd priority attribute and therefore a greater chance of finding a good path according to the 2nd priority QoS attribute.

It is therefore important that the policies for selecting QoS-aware paths are carefully considered. We have shown that selecting paths based on bandwidth can deliver higher bandwidth but also significantly higher delay than the best-effort (BE) Internet; and that selecting paths based on delay provides only a small improvement on the BE Internet, since the shortest AS path is effectively the one with the lowest delay. Further, if QoS parameters are compared too strictly, use of QoS information can in fact hinder delivered QoS. However, significant QoS improvements over the BE Internet are obtained if two QoS parameters (delay and bandwidth) are used to select paths, with better overall QoS obtained by prioritising delay over bandwidth, rather than bandwidth over delay.

3.4.3.1.1 Effect of injecting dynamic QoS information into q-BGP

This series of tests considers the impact of injecting dynamically monitored QoS information into q-BGP UPDATES. The same QoS attributes are used as those used in the static case: one-way delay and available bandwidth. The difference is that the values used for NIA queuing delay – the queuing time on the inter-domain link for the NIA in use between the local AS and its neighbour who advertised the route – and NIA capacity – the available bandwidth on the NIA on the same inter-domain link – are no longer statically assigned by the administration but monitored dynamically. On the one hand the use of dynamic information should overcome the problem of QA rush in the static QoS attribute case described above. This is because high quality routes will deliver degraded performance if they become too popular and suffer from congestion – available bandwidth will be reduced and queuing delay will be increased. But on the other hand this comes at the cost of potentially unstable routes: high quality routes may attract large volumes of traffic and therefore experience QoS degradation, which will result in new q-BGP UPDATES to announce the degraded QoS which will make the route less attractive to q-BGP selection processes in remote ASes meaning that the route is no longer selected, meaning less traffic and hence the monitored delay and available bandwidth will improve.

To mitigate the problems of route flapping, q-BGP should dampen its route advertisement and decision making processes. Three mechanisms were deployed to achieve this:

- Use of the equivalence margin in the route selection process as in the static QoS attribute case described previously. Narrow equivalence margins make route selection sensitive to small changes in advertised QoS attribute values and vice versa.
- Deployment of UPDATE-thresholds in the q-BGP advertisement process. This filters out small changes in monitored delay or bandwidth so that q-BGP UPDATES are not triggered unless there are significant changes. The sensitivity of the system to changes in QoS attribute values can be tuned by varying the threshold size. Independent values for delay and bandwidth thresholds may be specified.
- Use of a moving average of monitored values rather than instantaneous values to ensure that short-term changes do not trigger q-BGP UPDATE messages. An exponentially weighted moving average was implemented so that the sensitivity of the average to more recent monitored values as well as the number of samples included in the calculation could be tuned by varying a single parameter: the smoothing factor.

Tests were conducted with a range of smoothing factors, UPDATE-thresholds and equivalence margins and the main observation was that in many cases while the simulation converged the routing state did not converge to a single state: it cycled around a constant set of states. Delivered performance in terms of delays and throughput experience by the demands was degraded compared to the static case. This can be seen in Figure 73. This shows the results of tests on topologies of 50 ASes with average connectivity degree of four unidirectional links. The route selection policy was PRI_DEL-*_BW-*. Each point shows the mean delay and throughput for all demands in the system averaged over nine runs (three different topologies, each with three different full-mesh demand sets). The points show results for a range of equivalence margin settings (all pairings of delay margins of 100, 250, 500 and 750 ms, and bandwidth thresholds of 10K, 75K, 250K and 500K bandwidth units).

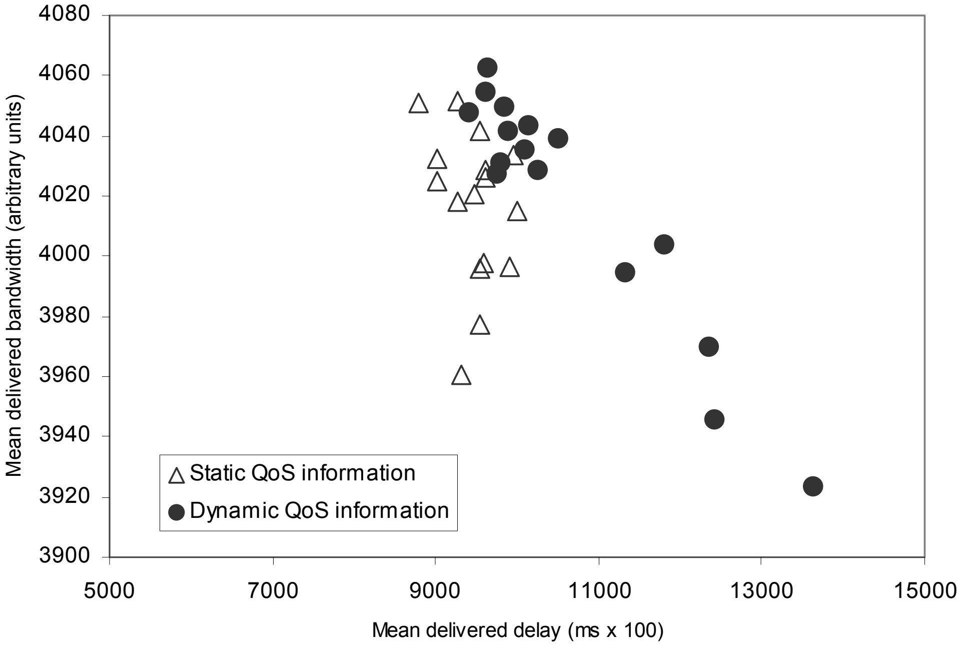


Figure 73 Delivered bandwidth vs. delivered delay in the case of static and dynamic QoS information injected into q-BGP UPDATEs

Although the different equivalence margin cases are not marked in the graph it is clear that in all cases the performance of the system with dynamically monitored QoS information was always worse than in the static case. The explanation for this is that in the dynamic case, after an initial period of settling, the set of routes selected by ASes cycles around a constant set of states. This can lead to inconsistent states between ASes because when an AS has selected a new route and generated an UPDATE message this has to be propagated to adjacent ASes who need to make their own decision on the best available route. This does not happen instantaneously and in the time it takes routes to be propagated to remote ASes the decision in the local AS may have changed due to revised updates it has received, or due to changes in local monitored information.

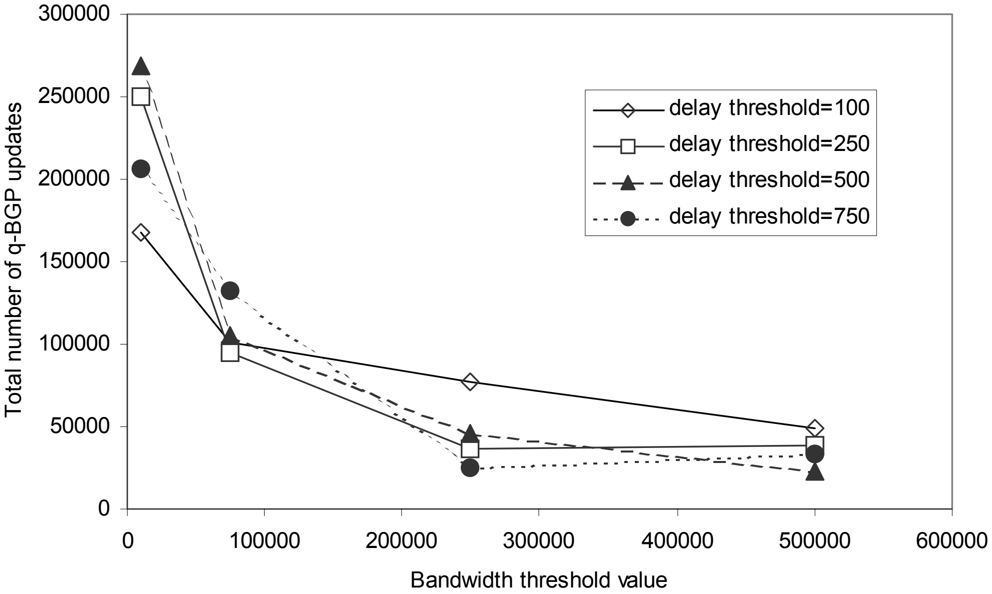


Figure 74 Number of q-BGP UPDATE messages against UPDATE-threshold value

The effect of varying the UPDATE-threshold is shown in Figure 74 and Figure 75. Figure 74 shows results from twelve different cases of UPDATE-threshold values. The number of q-BGP messages sent in the period of bootstrapping and initial settling before "convergence" to a stable cycle of routing states is shown in all cases. The route selection policy was PRI_DEL-500_BW-75 and the tests were conducted on topologies of 50 ASes with average connectivity degree of four unidirectional links. The points show the mean of the total q-BGP messages exchanged across all ASes averaged over nine runs (three different topologies, each with three different full-mesh demand sets). The effect of the bandwidth UPDATE-threshold on the total number of messages is clearly seen –higher threshold values reduce the number of messages and hence the overhead of the exchange of routing information. Figure 75 shows how the total number of q-BGP messages varies with bandwidth UPDATE-threshold in different topology sizes: 100 vs. 50 ASes. As before each point is the mean of nine runs (three different topologies generated with the same parameters each run with three different traffic demand patterns).

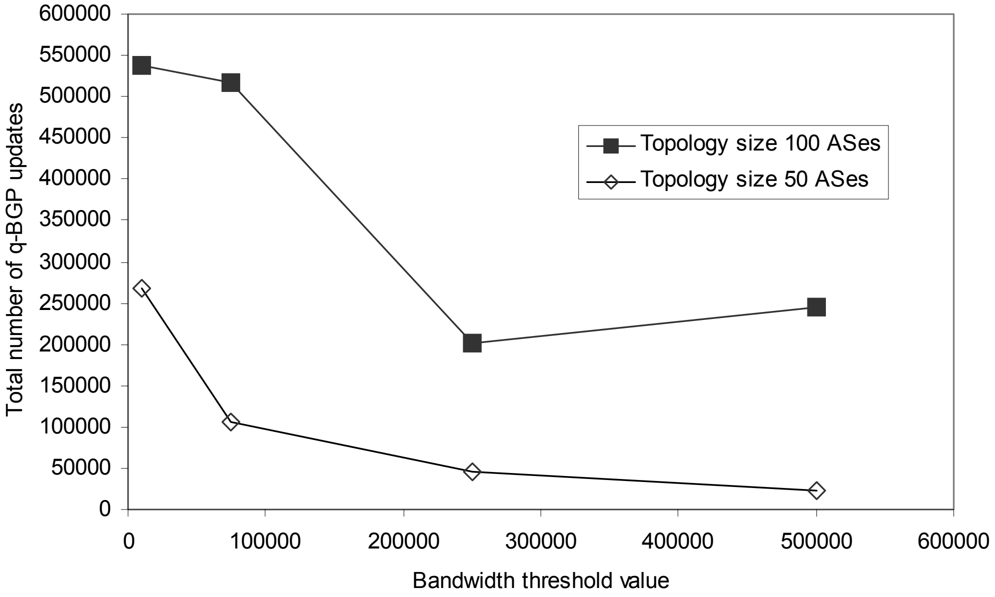


Figure 75 Number of q-BGP UPDATE messages against UPDATE-threshold value for different topology sizes

3.4.3.2 Scalability tests

In this set of simulation runs we examined the behaviour of q-BGP with static QoS attribute values in larger networks. Detailed modelling of internal AS topologies was avoided by assuming that intra-AS traffic treatment caused a constant average one-way delay between all border routers. The average delay was constant for a given AS but varied between ASs with a uniformly random distribution (between 5 and 50 ms). In the set of simulation results presented below it was assumed that no additional delay due to congestion was introduced at either inter- or intra-domain links, which is equivalent to the case of over-provisioned networks—a simplification introduced to reduce computation requirements for large-scale simulation runs. Four different topologies were generated for each topology size (defined by the number of ASs), with other topological parameters, such as degree of connectivity, remaining constant. For each instance of the topology, twelve separate examples of intra-AS one-way delay allocations were created. The results for each topology size are therefore the average over 48 simulation runs of different topologies and intra-AS delay distributions.

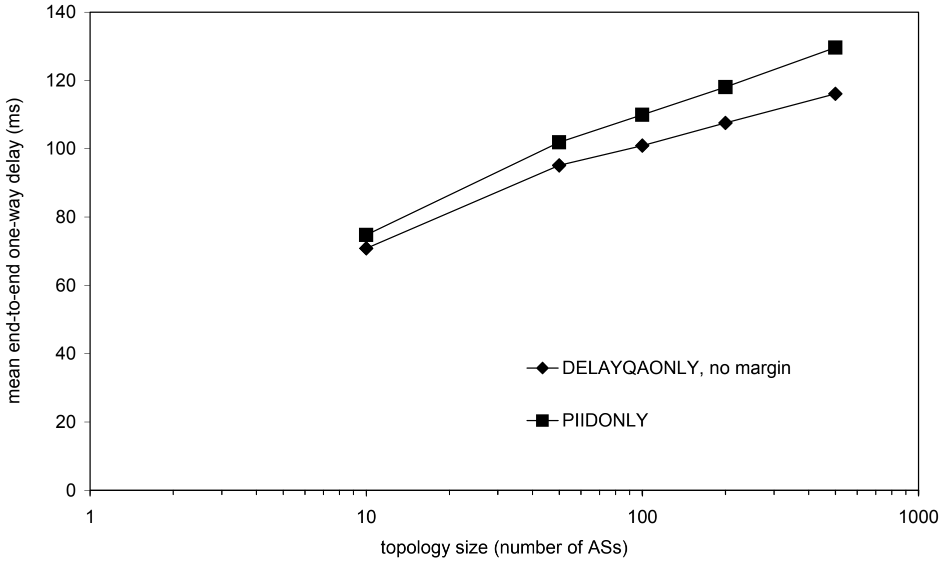


Figure 76 The effect of topology size on the end-to-end delay experienced by demands

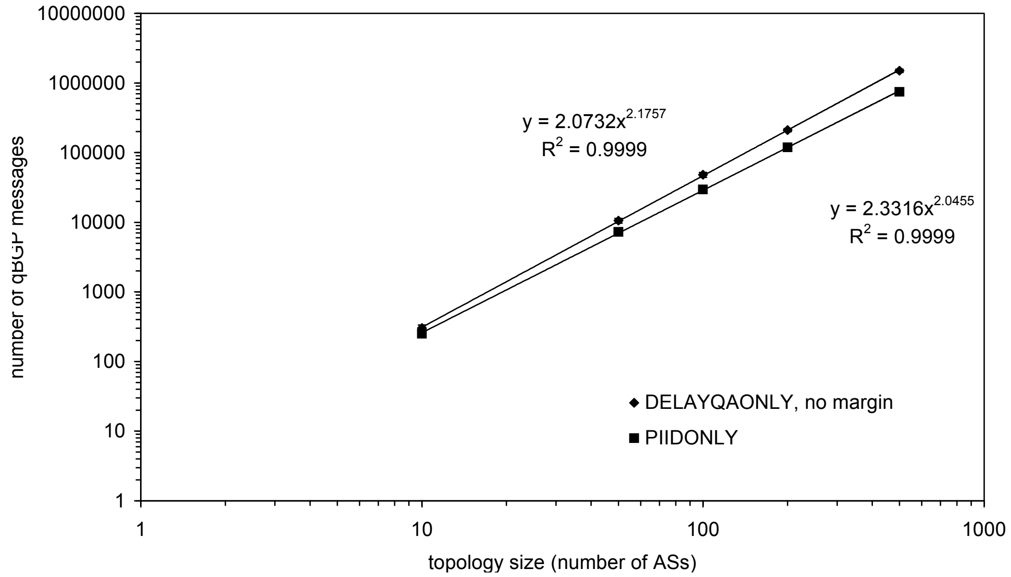


Figure 77 The number of q-BGP messages sent from initialisation until the network settles in a stable state with a full mesh of demands applied

The improvement in delay can be seen in Figure 76 as a function of AS topology size. It can be seen that the benefit of additional QoS information in q-BGP messages increases with topology size. This is due to a greater number of alternative AS paths between a given source-destination pair in a larger topology. Therefore the chances of finding an improved path based on one-way delay are increased.

The use of additional QoS information in q-BGP brings an additional overhead in terms of an increased number of q-BGP UPDATE messages. Figure 77 shows the total number of q-BGP messages sent from the first set of bootstrap messages through to a stable routing configuration. It should be noted that there is no message aggregation in these simulations, either on network prefixes or QoS attributes. When the two plots are extrapolated to a topology size of 18,000 ASs the q-BGP category two routing scheme produces approximately three times as many messages as category one q-BGP. The inclusion of additional QoS information in q-BGP therefore scales, in terms of number of q-BGP messages, in a similar way to q-BGP UPDATES and route selection based on m-QC id only. By this we mean that the number of messages forms a power law with topology size, which is equivalent to the scaling of BGP today.

The main reason for the increased number of messages required for convergence is that the preferred AS path, on QoS grounds, may not always be the shortest one. Imagine, from the perspective of the AS receiving q-BGP UPDATES, that the shortest AS path to a particular destination prefix has three AS hops, but the total one-way packet delay (in the data plane) as reported in q-BGP is significantly greater than an alternative five-hop AS path. According to the q-BGP route selection priority rules, the longer path with a smaller delay should be preferred. The q-BGP message received via the neighbouring AS announcing the 3-hop path is likely to arrive earlier than the one from the other neighbouring AS announcing the 5-hop path, due to the accumulation of processing time and propagation delay of the q-BGP route selection process at each intermediate AS. In the absence of the 5-hop shorter-delay announcement, q-BGP will select the first route and announce this to its peers. On receipt of the subsequent announcement of the shorter-delay path, q-BGP will select the latter route and propagate it to its peers: thereby increasing the total number of q-BGP messages and introducing a transient routing instability. One way of mitigating this would be for ASs to wait for some period to be sure they have received all likely updates, rather than make immediate decisions. This would improve the transient stability of the solution but at the cost of longer convergence times.

3.4.3.3 Stability tests

Table 25 shows convergence time for a range of q-BGP path selection policies with static QoS attribute values for a topology of 100 ASs with a NIA scaling factor of 2.0. Convergence time is measured as the number of simulation epochs required for all ASs to stabilise in terms of their path selection. Convergence is identified when no further q-BGP UPDATE messages are transmitted.

q-BGP Selection Policy	Mean number of simulator epochs until convergence
PIID-only	9.5
DELAYQA-only-50	10.4
DELAYQA-only no margin	10.8
PRI_DEL-100_BW-75	13.2
BWQA-only-125	16.1
PRI_BW-175_DEL-50	16.4
BWQA-only no margin	17.6

Table 25 Convergence time versus q-BGP selection policy

One reason for longer convergence times for some selection policies, e.g. BWQA-only with no margin of equivalence, is that they will determine that a newly arriving q-BGP UPDATE is better that the currently implemented path even if the new path outperforms the current one by only a tiny fraction. This will cause the AS to advertise its new path, which in turn will cause its neighbours to select the marginally better path, causing more q-BGP messages to be generated, and so on. As shown in earlier sections stability is not usually achieved in the case of injecting dynamic QoS attributes into q-BGP UPDATE messages. However in the dynamic case the tuning of equivalence margin, UPDATE-threshold and exponentially weighted moving average smoothing factor can improve stability to some extent.

3.4.4 Conclusions

The results show that performance in terms of delivered end-to-end delay or bandwidth is improved when q-BGP selection policies are employed to select paths based on QoS attributes injected into BGP messages. However, if the equivalence margin of QoS attributes on competing paths is set too small then a degradation of performance compared to that offered by classical BGP selection policies may be observed due to the observed phenomenon of "*QA rush*", where the best routes are quickly overloaded. This can be mitigated by increasing the margin of equivalence so that, while the worst paths are excluded, sufficient quantities of "good" paths are retained so that the subsequent selection between these, based on shortest AS path, results in sufficient routing diversity, which alleviates congestion.

It has been demonstrated that different route selection policies result in different delivered performance. Appropriate policies should, therefore, be selected to implement different Parallel Internets – e.g. delay or bandwidth constrained qualitative classes. It is important to state that this is in addition to any service differentiation implemented by utilising different PHBs/packet forwarding priorities within the routers of each AS. On the other hand this result indicates that end-to-end QoS differentiation is achievable even with homogenous forwarding behaviour.

It has been shown that injecting dynamically monitored QoS information into q-BGP does not improve performance in terms of delivered delay or throughput mainly due to routing instability. A conclusion to draw is that while QoS attributes are beneficial for system performance the frequency with which the QoS attributes are modified by the off-line administration needs to be carefully considered.

While the performance benefits of QoS-based path selection have been demonstrated it has also been shown that the cost of the solution is not prohibitive in terms of the overhead caused by additional q-BGP UPDATE messages. Simulation results of the worst-case value of equivalence margin for the DELAYQA-only q-BGP path selection policy, i.e. no margin, show that the number of q-BGP messages required for stable inter-domain routing scales with AS-topology size in a similar way to classical BGP. When scaled to current Internet topologies the results indicate that only three times the number of UPDATE messages is needed for convergence compared to classical BGP. With larger equivalence margins the total number of messages is reduced.

Stability tests show that convergence times are worst when q-BGP selection policies are most stringent. The adoption of these policies also delivers worse end-to-end performance and it is desirable on the counts of both convergence time and delivered QoS to adopt more moderate equivalence margin values. The results show that when the best performing q-BGP selection policies (in terms of delivered QoS) are adopted, convergence time is in the mid-range of observed values.

4 INTEGRATED PI ENGINEERING EXPERIMENTS

4.1 Integrated Inter-Domain Test-bed

The Partners of the AGAVE Project collaborated in the aim of building a pan-European distributed test-bed. The objective of such an effort was two-fold. On the one hand, this test-bed allowed to inter-connect already existing autonomous test-beds, in particular the MRDV test-bed and the tunnel service (LISP and IDIPS) test-bed. On the other hand, the test-bed allowed performing more extensive tests, making intra- and inter- domain solutions proposed in the AGAVE Project framework working together, proving the feasibility of the NPs/PIs realization.

The AGAVE integrated test-bed is based on a private VPN, using GRE tunnels, connecting partners' networks and creating a virtual AS for each participating partner. The VPN topology is depicted in Figure 78, showing in particular the GRE tunnels connecting partners' sites. Such a setup has multiple advantages. First important advantage is the fact that such a setup is compliant with security policies of all the partners. Using a VPN allows to achieve a certain degree of freedom in the choice of the address scheme, having no restrictions on the *"locators"* and *"IDs"* used in the Tunnel Service. Furthermore, the configuration of routing tables, provides the flexibility to setup reach and complex virtual topologies, allows to perform packet marking, and introduce bandwidth and delay constrains in order to test the solutions in specific conditions.

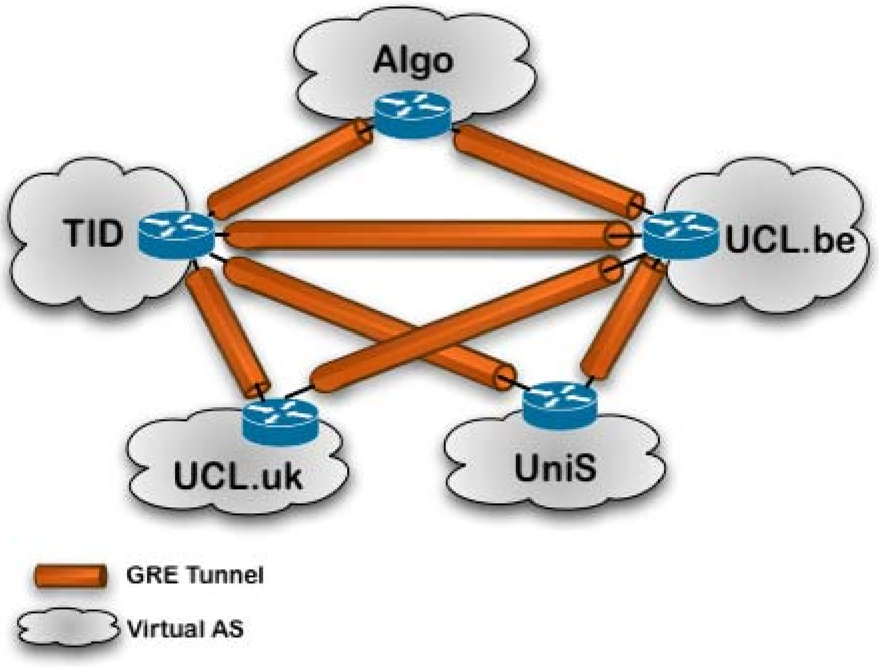


Figure 78 AGAVE Integrated test-bed

On top of the VPN of Figure 78 routing tables have been setup in such a way to have a virtual topology that includes inter-domain paths spanning over several virtual ASes. For each partner, the routing tables used are summarized in Table 26, Table 27, Table 28, Table 29, and Table 30.

TID	
Destination	**Next Hop**
10.220.0.0/16 (UCL.uk)	DIRECTLY CONNECTED
10.230.0.0/16 (Algo)	DIRECTLY CONNECTED
10.240.0.0/16 (UniS)	DIRECTLY CONNECTED
10.210.1.0/24 (UCL.be)	DIRECTLY CONNECTED
10.210.2.0/24 (UCL.be)	10.220.0.1 (UCL.uk)
10.210.3.0/24 (UCL.be)	10.240.0.1 (UniS)
10.210.4.0/24 (UCL.be)	10.220.0.1 (UCL.uk)
10.210.0.0/16 (UCL.be)	DIRECTLY CONNECTED

Table 26 Routing Table of TID Virtual AS

UniS	
Destination	**Next Hop**
10.200.5.0/24 (TID)	DIRECTLY CONNECTED
10.210.0.0/16 (UCL.be)	DIRECTLY CONNECTED
10.230.0.0/16 (Algo)	10.200.5.1 (TID)

Table 27 Routing Table of UniS Virtual AS

UCL.uk	
Destination	**Next Hop**
10.200.5.0/24 (TID)	DIRECTLY CONNECTED
10.230.0.0/16	10.200.5.1 (TID)
10.210.0.0/16	DIRECTLY CONNECTED

Table 28 Routing Table of UCL.uk Virtual AS

Algo	
Destination	**Next Hop**
10.200.5.0/24	DIRECTLY CONNECTED
10.210.0.0/16	DIRECTLY CONNECTED
10.210.4.0/24	10.200.5.1 (TID)
10.210.5.0/24	10.200.5.1 (TID)

Table 29 Routing Table of Algo Virtual AS

UCL.be	
Destination	**Next Hop**
10.220.0.0/16 (UCL.uk)	DIRECTLY CONNECTED
10.240.0.0/16 (UniS)	DIRECTLY CONNECTED
10.200.5.0/26 (TID)	DIRECTLY CONNECTED
10.200.5.64/26 (TID)	10.220.0.1 (UCL.uk)
10.200.5.128/26 (TID)	10.240.0.1 (UniS)
10.210.5.0/24 (TID)	DIRECTLY CONNECTED
10.230.1.0/24 (Algo)	10.220.0.1 (UCL.uk)
10.230.2.0/24 (Algo)	10.200.5.1 (TID)
10.230.0.0/16 (Algo)	DIRECTLY CONNECTED

Table 30 Routing Table of UCL.be Virtual AS

The resulting virtual topology obtained using the above listed routing tables is depicted in Figure 79. The figure also highlights the endpoints of each inter-domain path. These end-points are used as locators for the LISP Tunnel Service. Indeed, as depicted in the figure, in the Algo, TID, and UCL.be premises a LISP router has been deployed. The LISP Tunnel Service, coupled with the IDIPS Tunnel Service Controller, enables flexible inter-domain TE.

Behind each LISP router there is a private sub-network reachable only using the LISP protocol. These sub-networks are:

- 172.16.0.0/16 for TID
- 172.17.0.0/16 for UCL.be
- 172.18.0.0/16 for Algo

Such private clouds contain local test-beds used for the integrated tests. In particular:

- Algo: Behind the LISP router there are three different Linux boxes, used to generate traffic and also to diffuse video streaming.
- TID: Behind the LISP router there is the MRDV test-bed described in Section 2.2.4.
- UCL.be: Behind the LISP router there are a Linux box and a Windows XP Professional box.

The test-bed has been used to perform several measurements, some of them presented in Section 3.3.2 and Section 3.3.4.

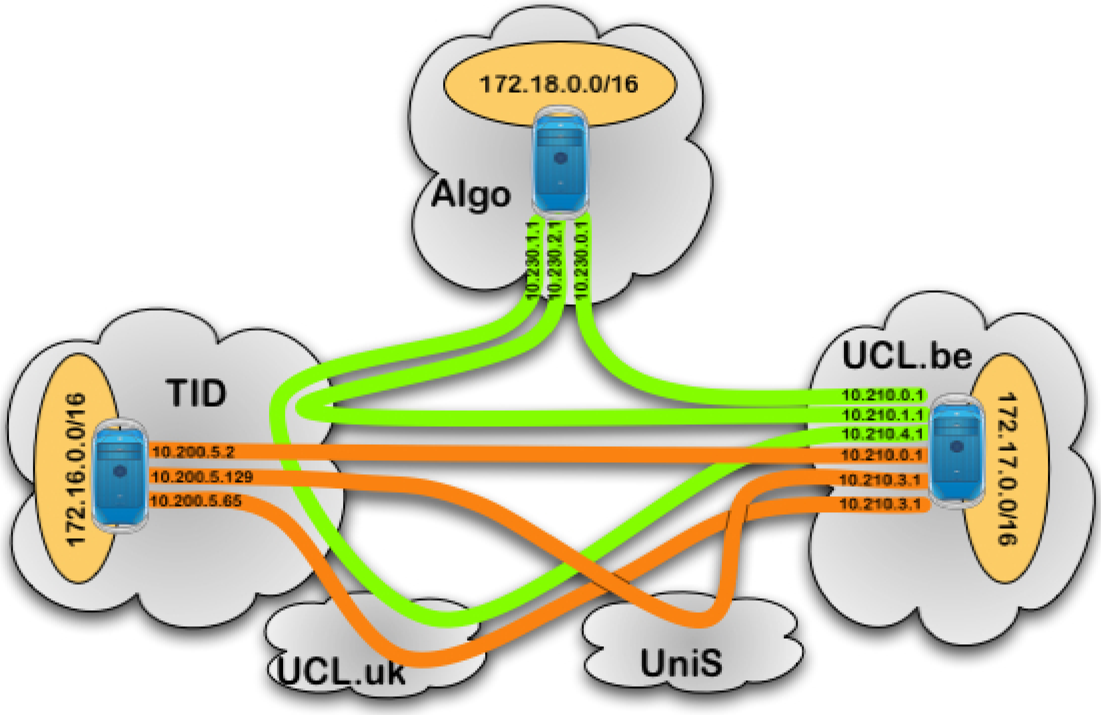


Figure 79 AGAVE Integrated test-bed virtual topology

4.1.1 Scenarios for PI Realization

In the following subsections we describe some PI realizations that can be emulated using the AGAVE integrated test-bed.

4.1.1.1 Multi-path Inter-Domain Routing

A first simple realization consists in putting the LISP Tunnel Service above NPs/PIs level. The example is depicted in Figure 80. In this scenario NP/PI is realized by the inter-domain routing infrastructure. In the present case, the LISP Tunnel Service is used to connect two distant domains, (TID and UCL.be in the specific case) using a single locator on each side. The inter-domain network between the two domains will implement NP/PI in a total transparent way. This means that, at LISP level, all flows are sent to/from the same end-points, without any differentiation. The inter-domain routing will provide traffic differentiation taking advantage of multi-path routing when available.

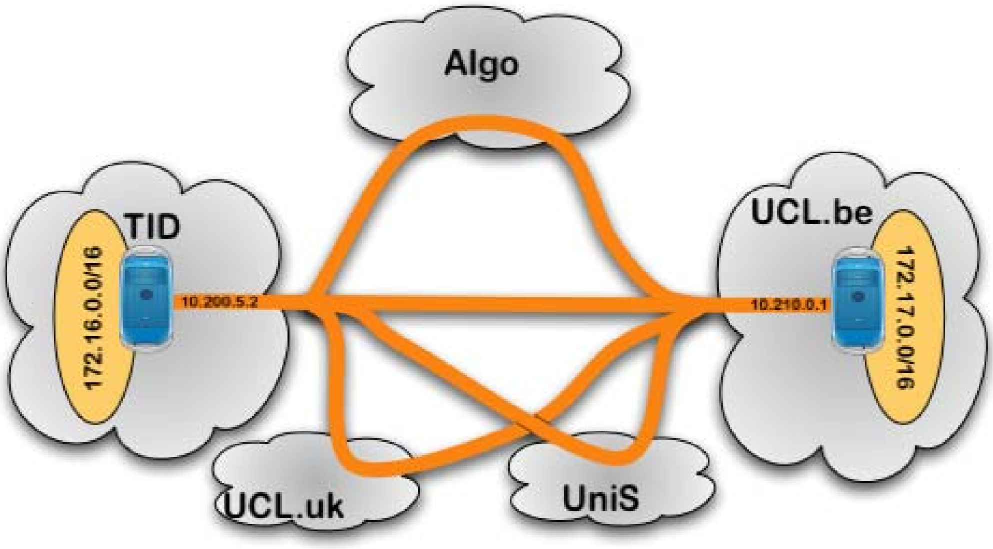


Figure 80 LISP Tunnel Service above NPs/PIs.

4.1.1.2 Optimizing Inter-Domain Path-Selection within a single PI

The second scenario consists in a single PI between two domains, on which thanks to the LISP Tunnel Service and the IDIPS Tunnel Service Controller the path used is selected on a per-flow basis in order to optimize certain criteria. Figure 81 shows an example of this scenario in the context of the AGAVE integrated test-bed. Two separate domains (TID and UCL.be in this specific case) have several different locators (tunnels end-point) that can be used to select different path with different characteristic. Flows can be routed on different paths by simply controlling the mapping between the locators (tunnels end-points) and the EID (the identifiers inside the private networks 172.16.0.0/16 and 172.17.0.0/16). The mappings are managed by IDIPS (the Tunnel Service Controller), which is aware of the performances (e.g., delay and available bandwidth), the capabilities (e.g., service differentiation through DiffServ support), and the resources of the different paths.

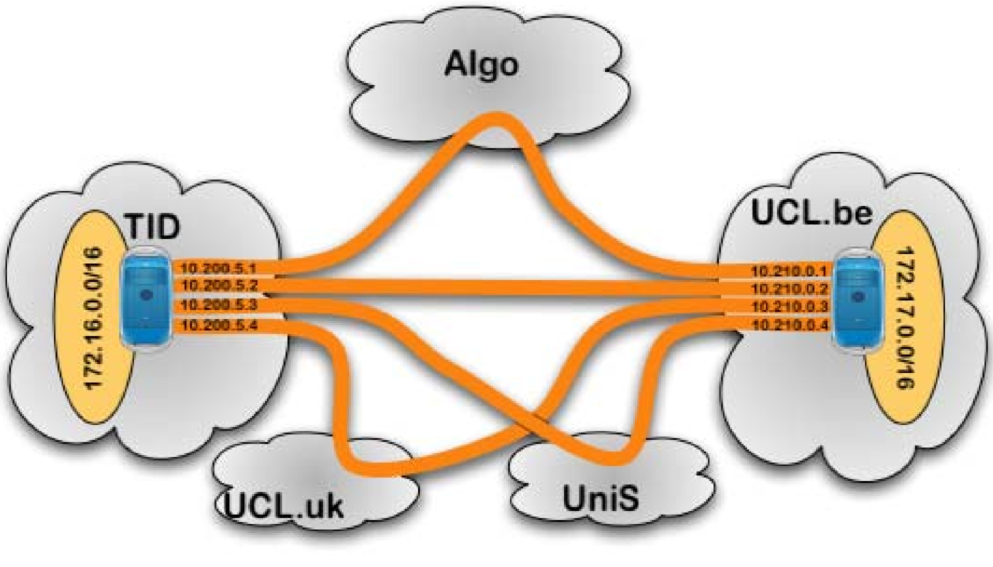


Figure 81 LISP Tunnel Service Path Selection within a single PI

4.1.1.3 Binding NPs with Inter-Domain Paths to form Coexisting Multiple PIs

The third scenario consists in having multiple coexisting PIs that can be bind to different inter-domain NPs. As depicted in Figure 82, two different domains (in the specific case TID and UCL.be) can implement two different PIs. A first PI is *"public"* meaning that is based on the used of several different locators from the two domains that are publicly announced in the mapping distribution system. This means that the resources and links of these locators can be used also by other domains, which in turn means that QoS can be guarantied up to a certain point. For traffic with tight QoS constrains, the two domains use a different PI, consisting in *"private"* locators. These latter are not announced on the global mapping system, rather they are known only among the two domains that will use them, which have also full control of them, hence being able to provide QoS guarantee. These two PIs, can be flexibly used, thanks to the LISP Tunnel Service and the IDIPS Tunnel Service Controller, to support the NP established among the participating domains.

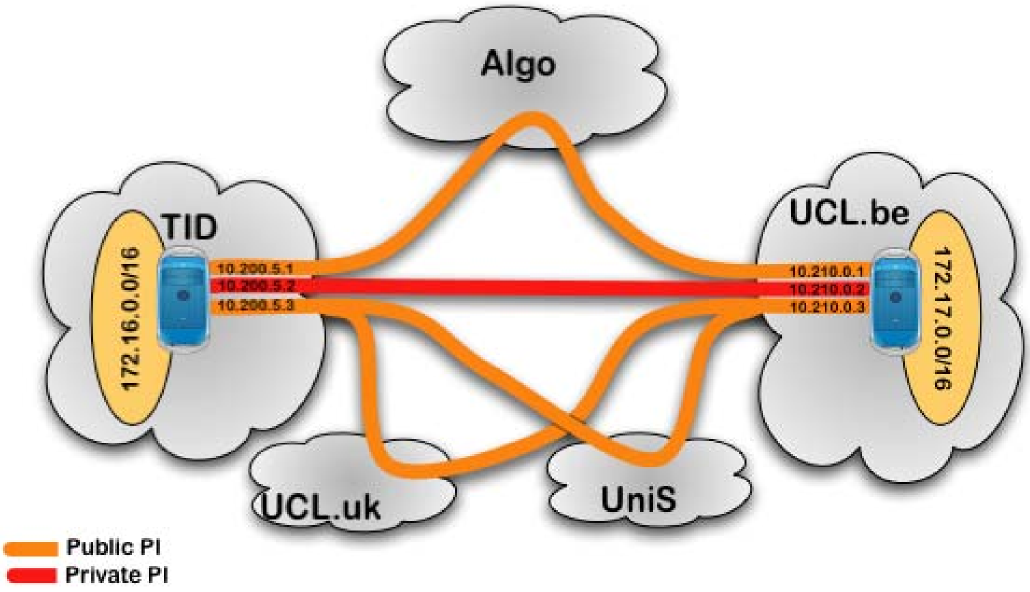


Figure 82 LISP Tunnel Service supporting multiple coexisting PIs

5 CONCLUSIONS

The present document described the performed experimentation activities of the AGAVE Project and their main outcomes. Following the logical classification proposed in D3.1 [D3.1], the main results achieved are summarized in the following.

NP engineering experiments:

- MTR (Multi-Topology Routing), with the proposed AMPLE mechanism, has proved to be able to handle unexpected traffic dynamics in an efficient way, achieving near optimal performance with only a small number of different routing topologies. Furthermore, the proposed scheme shows to be highly efficient also in dealing with single link failures.
- MRDV (Multi-Path Routing with Dynamic Variance) has been extended to support multiple traffic classes, allowing the original MRDV algorithm to generate different Network Planes. Both simulations and test-bed experimentations have validated its implementation on the Quagga router. The tests results show that the new algorithm is able to differentiate traffic prioritizing one class of traffic over the others. Measurements performed show that the Network Plane differentiation provided by MRDV with QoS is able to support high load from Best Effort traffic while guaranteeing performance for Premium traffic.
- NP Emulation Platform experimentation: NPEP shows the validity of the AGAVE specifications in that they (a) can be workable and (b) can lead to working network configurations.

Inter-domain routing experiments:

- Performance in terms of delivered end-to-end delay or bandwidth is improved when q-BGP selection policies are employed to select paths based on QoS attributes injected into BGP messages. Different route selection policies result in different delivered performance. Appropriate policies should, therefore, be selected to implement different Parallel Internets. Injecting dynamically monitored QoS information into q-BGP does not improve performance mainly due to routing instability. A conclusion to draw is that while the use of QoS attributes in inter-domain routing is beneficial for system performance the frequency with which QoS attributes are modified needs to be carefully considered.
- Measurements concerning resilience-aware BGP/IGP TE interactions in a simulation environment revealed that the tunnelling mechanism with judicious selection of tunnel endpoint can achieve high fast reroute coverage and improve post-failure load balancing.
- BGP Planned Maintenance experimentation in test-bed, when MPLS forwarding is used, including for BGP/MPLS VPN, can achieve a hitless shutdown with 0 packet loss. In the case of IP forwarding, it significantly reduces the loss of connectivity.

  IP tunnelling experimentation based on simulation suggests that the size of the cache maintaining the mappings can be limited in size by using a relatively small timeout for the entries. The overhead introduced by the tunnelling, on which the locator/ID separation paradigm is based, does not pose any problem since it is quite small. The implementation has of LISP, deployed on the integrated test-bed shows that the delay added by the encapsulation/decapsulation operation is very limited. LISP tunneling approach coupled with the IDIPS controller has shown through presented tests that allow flexible TE capabilities for realizing PI, by separating the routing, the mapping and the path selection algorithm, offering strong perspective for operators (i.e., flexibility, costs, robustness) and for the customer (i.e., performances, costs).

6 REFERENCES

[ABILENE] The Abilene Network topology: http://www.cs.utexas.edu/~yzhang/research/AbileneTM/

[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

[BRES03] T.C. Bressound et al, “Optimal Configuration for BGP Route Selection,” Proc. IEEE INFOCOM, 2003.

[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.

[CHAN83] V. Chankong et al, Multiobjective Decision Making-Theory and Methodology, Elsevier, New York, 1983.

[D3.1] AGAVE deliverable D3.1, Initial Specification of Mechanisms, Algorithms and Protocols for Engineering the Parallel Internets and Implementation Plan, IST AGAVE Project, http://www.ist-agave.org

[D3.2] AGAVE deliverable D3.2, Specification of Mechanisms, Algorithms and Protocols for Engineering the Parallel Internets, IST AGAVE Project, http://www.ist-agave.org

[D4.1] AGAVE deliverable D4.1, Test Specification and Experimentation Plan, IST AGAVE Project, http://www.ist-agave.org

[DUMMY] Luigi Rizzo, Dummynet: a simple approach to the evaluation of network protocols, ACM Computer Communication Review, Vol. 27, Num. 1, Pages 31-- 41, 1997.

[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.

[FLOWT] Flow-Tools. http://www.splintered.net/sw/flow-tools/

[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.

[IANN07] L. Iannone and O. Bonaventure. On the Cost of Caching Locator/ID Mappings. 3rd Annual ACM CoNEXT Conference. December 2007.

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

[IPLANE] iPlane: An Information Plane for Distributed Services. http://iplane.cs.washington.edu/

[IPERF] C.-H. Hsu and U. Kremer. IPERF: A framework for automatic construction of performance prediction models. In Workshop on Profile and Feedback-Directed Compilation (PFDC), Paris, France, October 1998.

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

[LISP07] D. Farinacci, V. Fuller, and D. Oran, "Locator/id separation protocol (LISP)," Internet Draft, April 2008, available online at: www.ietf.org/internet-drafts/draft-farinacci-lisp-07.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.

[NOBEL-DIM] Europe 28 nodes reference network scenario", IST NOBEL Project, http://www.ist-nobel.org

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

[NETFLOW] Cisco IOS NEtFlow – Cisco Systems. http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html

[NETPROG] Stevens, W., Fenner, B., and A. Rudoff, "UNIX Network Programming, The Sockets Networking API.", Addison-Wesley Professional Computing Series Volume 1 - Third Edition, 2004.

[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.

[NUCC07] A. Nucci et al., "IGP Link Weight Assignment for Operational Tier-1 Backbones," IEEE/ACM Transactions on Networking, October 2007.

[OLISP] L. Iannone and O. Bonaventure, OpenLISP Implementation Report, Internet Draft IETF Network Working Group, draft-iannone-openlisp-implementation-00.txt, February 2008.

[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.

[RFC1771] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4),IETF RFC 1771.

[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.

[ROUTE] University of Oregon Route Views Project. http://www.routeviews.org/

[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.

[SPRIN04] N. Spring et al., "Measuring ISP Topologies with Rocketfuel," Proc. ACM SIGCOMM, 2004.

[TCPIP] Wright, G. and W. Stevens, "TCP/IP Illustrated Volume 2, The Implementation.", Addison-Wesley Professional Computing Series, 1995.

[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.

[UHLI05] S. Uhlig et al, "Tweak-it: BGP-based Interdomain Traffic Engineering for Transit ASs," Proc. NGI Conference, 2005.

[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.

7 ACRONYMS

AMU	Average Maximum link Utilization
AS	Autonomous System
ASN	AS Number
ATC	Adaptive Traffic Control
BE	Best Effort
BGP	Border Gateway Protocol
BGP PM	BGP Plan Maintenance
BWQA	BandWidth QoS Attribute
CE	Customer Edge
CoS	Class of Service
CBR	Constant Bit Rate
CDF	Cumulative Distribution Function
CF	Cost Function
CPA	Connectivity Provisioning Agreement
CPU	Central Processing Unit
DE	Decision Engine
DiffRouting	Differentiated Routing
DiffServ	Differentiated Services
eBGP	external BGP
ECMP	Equal Cost Multi Path
EID	End host Identifier
FC	FRR Coverage
FDoI	Full Degree of Involvement
FRR	Fast Re-Routing /Fast Re-Route
FS	Failure State
FTP	File Transfer Protocol
GA	Genetic Algorithm
GEANT	Pan-European Gigabit Research and Education Network
GLPK	GNU Linear Programming Kit
GNU	GNU is Not Unix
HMU	Highest Maximum link Utilization
HPR	Hot Potato Routing
HRR	Hierarchical Route Reflector
HTML	HiperText Markup Language
iBGP	internal BGP
IDIPS	ISP-Driven Informed Path Selection

IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
INP	IP Network Provider
IP	Internet Protocol
IRTF	Internet Research Task Force
IS-IS	Intermediate System to Intermediate System
ISP	Internet Service Provider
KB	Knowledge Base
LISP	Locator/ID Separation Protocol
LSP	Label Switched Path
LoC	Loss of Connectivity
L3VPN	Layer 3 VPN
METL-BGP	IGP Metric assignment TooL-BGP
MLU	Maximum Link Utilization
MPLS	Multi Protocol Label Switching
MRDV	Multipath Routing with Dynamic Variance
MT-IGP	Multi-Topology IGP
NIA	Network Interconnection Agreement
NP	Network Plane
NPEP	Network Plane Emulation Platform
NS	Network Simulator / Normal State / Network Service
OC	Optical Carrier
OCCI	Oracle C++ Call Interface
OLWO	Offline Link Weight Optimisation
OSPF	Open Shortest Path First
PE	Provider Edge
PHB	Per Hop Behaviour
PI	Parallel Internet
PIC	Path Information Collector
PIID	PI IDentifier
PL-SQL	Procedural Language - Structured Query Language
PNO	Proportion to Near-Optimal performance
PoP	Point of Presence
QA	QoS Attribute
qBGP	QoS-enhanced Border Gateway Protocol
QoS	Quality of Service
RIB	Routing Information Base

RLOC	Routing LOCator
RR	Route Reflector
RT	Routing Topology
SNMP	Simple Network Management Protocol
SP	Service Provider
STM	Synchronous Transport Module
TCP	Transmission Control Protocol
TE	Traffic Engineering
TM	Traffic Matrix
ToS	Type of Service
TS	Tunnel Service
TSC	Tunnelling Service Controller
TTL	Time To Live
UDP	User Datagram Protocol
VPN	Virtual Private Network
VRF	Virtual Routing and Forwarding
XORP	eXtensible Open Router Platform