MESCAL

Management of End-to-end Quality of Service
Across the Internet at Large

IST-2001-37961

D3.2: Final Experimental Results: Validation and Performance Assessment of Algorithms and Protocols for Inter-domain QoS through Service-driven Traffic Engineering

Document Identifier: MESCAL/WP3/ALGO/D3.2/final	
Deliverable Type: Report **Deliverable Nature:** Public	**Contractual Date:** 31 August 2005 **Actual Date:** 4 July 2005

Editor:	Eleni Mykoniati, Algonet S.A.
Authors:	*FTR&D:* M. Boucadair, P. Morand *TRT:* H. Asgari, R. Egan *UCL:* J. Griem, D. Griffin, J. Spencer *UniS:* S. Georgoulas, K. H. Ho, M. Howarth, P. Trimintzios, N. Wang *Algo:* P. Georgatsos, I. Liabotis, E. Mykoniati
Abstract:	The deliverable presents the tests undertaken and the results produced for validating and assessing MESCAL's solution for inter-domain QoS delivery. Experimentation was carried out through simulations and in testbeds comprised of experimental Linux-based routers. It covered functional and performance testing, cost/benefit, scalability, stability, usability tests, of the following components: q-BGP, the enhanced inter-domain QoS routing protocol, regarding its employment in providing QoS in the Internet; inter- and intra-domain traffic engineering for uni- and multicast traffic, pSLS negotiations and handling functions for producing the required TE info and c/pSLS admission control. For each of these component tests, results and the conclusions drawn are presented. The deliverable also includes a scalability analysis of the MESCAL approach.
Keywords:	Inter-domain QoS, Experimentation, Validation, Performance Assessment, Results.

France Telecom Research and Development	FTR&D	Co-ordinator	France
Thales Research and Technology	TRT	Principal Contractor	UK
University College London	UCL	Principal Contractor	UK
The University of Surrey	UniS	Principal Contractor	UK
Algonet SA	Algo	Principal Contractor	Greece

Project funded by the European Community under the
"Information Society Technology" Programme (1998-2002)

Executive Summary

This deliverable presents the experimentation work undertaken for validating and assessing the performance of the functionality pertaining to the MESCAL solution for inter-domain QoS delivery.

The MESCAL solution relies on interactions between adjacent providers at the service layer, for establishing agreements for QoS traffic exchange, pSLSs, and at the network (IP) layer for finding, determining and maintaining suitable inter-domain QoS routes. The commonly used inter-domain routing protocol BGP has been enhanced to convey QoS related information. In addition, the solution specified the required service management and traffic engineering functionality per provider domain. Three technical options of the general MESCAL solution have been specified to meet the QoS requirements of different service types. Solution option 1 provides for loose (qualitative) QoS guarantees across the Internet, while solution option 2 delivers statistical guarantees (i.e. not per flow but per flow aggregate) for quantitative QoS targets, in addition to qualitative QoS guarantees. Solution option 3 is suitable for services requiring hard QoS guarantees. The technical aspects and details of the MESCAL solution are included in deliverables [D1.1], [D1.2] and [D1.3].

Experimentation was carried out in either physical testbeds, comprised of Linux-based routers, or simulated networks and covered functional validation and performance assessment aspects in terms of cost/benefit, scalability and stability assessment.

The deliverable presents the tests, results and conclusions drawn regarding the following functional aspects of the MESCAL inter-domain QoS delivery solution, including:

- Behaviour of the specified q-BGP protocol and associated route selection process;
- Off-line inter-domain TE algorithms and their coupling with intra-domain TE;
- Off-line intra-domain IP-based QoS TE algorithms;
- Off-line intra- and inter-domain multicast TE algorithm;
- SLS Mgt functions -pSLS modelling, negotiation, translation and request handling- and admission control on c/pSLS invocations;
- 'In-router' deployment and operation of q-BGP and delivery of inter-domain QoS with loose guarantees according to the specified solution (option 1) in a realistic network set-up (testbed);
- Delivery of inter-domain QoS with hard QoS guarantees through the establishment of inter-domain LSPs (MPLS tunnels) based on the concept of PCSs (Path Computation Systems) according to the specified solution (option3) in a realistic network set-up (testbed).

Furthermore, the deliverable includes a scalability analysis of the overall MESCAL solution approach. The scalability analysis addressed a number of aspects of the MESCAL solution, including:

- The extent and complexity of message flow/processing for pSLS set-up during the negotiation phase. In this respect, a comparison between the CADENUS model, see [CADENUS], and the MESCAL model in serving service requests is made;
- An analysis of the MESCAL QoS peering model in terms of the number of pSLSs required for large networks;
- An analysis of the number and granularity of QoS Classes required for the MESCAL solution options.

The results of the tests undertaken prove the validity and feasibility of the MESCAL inter-domain QoS delivery solution and the proposed algorithms/schemes and protocols. They show that better performing routes for carrying QoS traffic can be established through the proposed approach (q-BGP exchanges, following pSLS establishment), compared to using standard BGP. The specified traffic engineering and service handling functions performed well, giving favourable results compared to ad-hoc configurations/solutions or alternative schemes.

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

1.1 Background

MESCAL addresses the problem of IP QoS-based service delivery across different provider domains. MESCAL adopts a hop-by-hop, cascaded model for the interactions between providers, at the service and network (IP) layers. Interactions at the service layer aim at the establishment of agreements for QoS traffic exchange, pSLSs in MESCAL terminology, to allow providers to expand the topological scope of their offered QoS-based services beyond the boundaries of their domains. Interactions at the IP layer are required to enable providers to find, determine and maintain suitable QoS routes for forwarding traffic in the Internet. In addition to appropriate protocols for supporting these interactions, MESCAL has specified the required service management and traffic engineering functionalities per provider domain to the end of effectively supporting these interactions, while optimising the utilisation of the network resources.

Driven by the different levels of QoS guarantees on packet transfer performance and bandwidth that could be provided to services – loose, statistical and hard contractual QoS guarantees – three corresponding technical solution options have been specified. As such, each solution option suits the needs of a different service type, targeting different customer/user segments and requiring different levels of operational complexity and scalability. Solution option 3 is suitable for services requiring hard QoS guarantees but with the inherent limitation that it cannot scale to the mass market (size of the Internet). Following the aggregate philosophy of DiffServ networks, solution option 1 has been designed to provide for loose, qualitative QoS guarantees across the Internet, while solution option 2 delivers statistical guarantees (i.e. not per flow but per flow aggregate) for either quantitative or qualitative QoS targets. The technical targets, aspects and constraints of the three MESCAL solution options have been presented in [D1.1], while suitable protocols and algorithms are described in [D1.2] and [D1.3].

Technical work in the MESCAL project is split over 3 work packages (WPs), and follows a phased approach: a theoretical phase followed by an experimentation-driven design and implementation phase and subsequently by an experimentation and dissemination phase. WP1 – Functional Architecture and Algorithms – specifies the inter-domain solution, per-domain architecture and related protocols and algorithms. WP2 – System Design and Implementation – develops aspects of the specified functionality subject to experimentation and required testing components. WP3 – Integration, Validation and Experimentation – sets up the experimentation infrastructure, testbeds and simulators, and conducts experiments with the purpose to validate and assert on performance of the specified functionality.

1.2 Scope of the Deliverable

The deliverable presents the tests, results and conclusions drawn regarding the validity and feasibility of the proposed inter-domain QoS delivery solution and its specified functional aspects; q-BGP, intra- and inter-domain traffic engineering and pSLS-aware service handling functions.

1.3 Organisation of the Deliverable

The rest of this document is structured as follows:

Chapter 2 presents the overall experimentation approach of the project.

Chapter 3 presents tests and results regarding the specified off-line TE algorithms; intra- and inter-domain traffic engineering for uni- and multi-cast traffic. As inter- and intra-domain traffic engineering inter-depend, tests regarding alternative ways for their coupling are also included.

Chapter 4 focuses on the specified protocol, q-BGP, for inter-domain QoS routing. Simulation-based tests and results are presented to assess q-BGP behaviour and acquire insight into intrinsic aspects of its operation in Internet-like topologies. It also presents testbed-based tests to verify the validity of the

q-BGP implementation and operation in a realistic network environment to realise the proposed inter-domain QoS delivery solution. Furthermore, it includes testbed results regarding the computation of QoS-constrained paths using the PCS-based approach.

Chapter 5 presents tests and results regarding the specified c/pSLS-aware service handling functions. It focuses on pSLS negotiations with emphasis on automated client-side logic and on cSLS admission control taking into account inter-domain considerations.

Chapter 6 analyses the proposed solution for providing QoS in the Internet from a scalability perspective.

Chapter 7 summarises the conclusions drawn from the undertaken tests.

Appendix A outlines the testbed topology and set-up used in the tests and appendix B presents the detailed validation tests undertaken in the testbed and the results yielded.

2 EXPERIMENTATION FRAMEWORK

2.1 Experimentation Activities

Experimentation is an essential aspect of MESCAL work to verify and validate the overall project objectives. Table 1 depicts the related activities undertaken by the project, in terms of the:

- Functional aspect under test (with reference to the functional architecture in [D1.1] and algorithm/protocol specifications in [D1.2]).
- Type of the environment where experimentation will be undertaken.
- Category of the experiments to be carried out.

Component/ Sub-system	Environment	Reference Section	Functional	Benefit/Cost	Scalability	Stability	Usability
Offline Inter-domain TE	Simulation	3.1	x	x	x		
Offline Intra-domain TE	Simulation	3.2.4	x	x	x	x	
Offline Intra-domain Multicast TE	Simulation	3.4	x	x	x	x	
Data Plane	Testbed	4.2	x				
Dynamic Inter-domain TE q-BGP	Simulation	4.1		x	x	x	
	Testbed	4.2.4	x	x		x	x
Dynamic Inter-domain TE PCS	Testbed	4.3.4	x	x			x
pSLS Ordering	Simulation	5.1	x	x			
SLS Order Handling	Simulation	5.2	x				
SLS Invocation Handling	Simulation	5.3	x	x		x	

Table 1: Experimentation Activities

2.1.1 Experimentation Environment

Experimentation activities were carried out both in testbed and simulated network environments, as appropriate to the aspect under test and the experimentation objectives. Specifically, experiments were undertaken:

- In a testbed comprised of Linux-based routers that enabled the incorporation of the specified inter-domain QoS routing protocol, q-BGP. The project testbed is provided by FTR&D and is located in Caen, France.
- In simulators, which, depending on the aspect of the network environment they simulate, can be distinguished into:
  - Dynamic network operation simulation engines, simulating the dynamics of network behaviour at a level of abstraction appropriate to the experiment e.g. at a packet, flow or protocol or control-plane activity levels.
  - Static network environment simulation tools, simulating the static aspects of the network environment i.e. the context in which the network is to operate; such aspects include network topology, number of supported QoS-classes, established service agreements, aggregate QoS traffic demands and QoS traffic generation patterns.

Three incremental phases of testbed-based experimentation have been identified, as discussed below:

- The objective of the first phase was to deploy an operational testbed including several ASs exchanging BGP-based inter-domain routing information between them and exercise the notion of meta-QoS-classes within separate autonomous systems.
- The objectives of the second phase were to deploy q-BGP and the associated route selection algorithm, as specified by MESCAL, thus creating a prototype of the solution option 1 (loose end-to-end guarantees on multiple meta-QoS-class planes) and verify its operation.
- The objectives of the third phase were to deploy PCSs and their communication protocol on top of set-up of phase 2 and validate the machinery for computing QoS paths across domains.

2.1.2 Experimentation Categories

As for their objectives, experimentation activities fall under the following commonly recognised categories:

- Functional validation experiments, aiming at assessing feasibility of implementation and validity of specifications. Not all functional validation tests and results carried out are reported in this deliverable for reasons of document length.
- Performance assessment experiments, aiming at assessing the behaviour of the aspect under test in a variety of network operation and environment set-ups and conditions. Behaviour is assessed in terms of scalability, stability, sensitivity and yielded benefits/incurred cost. Specifically:
    - Benefit/Cost assessment experiments aim at assessing the benefits/costs that the aspect under test yields/incurs in network performance, as measured through specific metrics in a representative set of network and traffic cases.
    - Scalability assessment experiments aim at calculating and verifying the resource requirements and/or computational performance of the aspect under test as a function of various uncontrollable variables, to see if it can be used in a large scale deployment.
    - Stability assessment tests verify that the aspect under test, given its specified dynamics/responsiveness, is operating in a way that drives the network to a stable state of operation, in a representative set of network and traffic cases.
    - Usability tests demonstrate that the aspect under test can operate as expected (according to its functional objectives) in terms of policy-based and/or tuning parameters upon which it may depend.

Obviously, experimentation objectives are restricted by the capabilities of the experimentation environment. As such, performance assessment experiments were primarily performed in a simulated network environment, static or dynamic, while functional validity experiments fit better in a testbed environment.

2.2 Experimentation Structure

The identified experimentation activities were specified in a clear and concise manner using a common structure/template, along the following headings:

Objectives: An answer to the question "What do we want to test?". The aspects under test (specified algorithm, protocol, mechanism) and the particular goals of experimentation are outlined. Specifically, the broad experimentation categories of functional validity and assessment of benefit/cost, scalability, stability and usability are qualified in terms of concrete objectives as appropriate to the functional aspect under test.

Performance Metrics: The metrics inherent to the particular functional aspect under test that quantify the experimentation objectives such as processing time, overhead, throughput, size of etc. are

described. How these metrics can be obtained, e.g. through probes in the entity under test or through test tools, is also discussed where appropriate.

Controlled Variables: The configuration parameters of the aspect under test. The defined performance metrics will be calculated as a function of these configuration parameters.

Uncontrolled Variables: The parameters of the external environment where the aspect under test is to operate are defined affecting its behaviour and/or its performance. Such parameters are network topology, volume and symmetry of traffic, number of peers, contracts etc. Generators or models for creating a realistic and representative set of their values are described where appropriate.

Experimentation Environment: The platform and the set-up upon which the envisaged experimentation is to be carried out are described in terms of: components of the functional architecture, experimentation platform and required test tools, their capabilities and interactions.

Test Campaigns: The tests to be carried out in achieving the specified objectives. Each of the tests aims at verifying/assessing a particular aspect of the behaviour/performance of the functional aspect under test (quantified by appropriate performance metrics) in a variety of test cases (quantified by appropriate combinations of uncontrolled variables) as a function of its configuration parameters (quantified by appropriate controlled variables). Tests are aggregated in test suites according to the general category they fall in.

3 OFFLINE TRAFFIC ENGINEERING TESTS AND RESULTS

3.1 Inter-domain Traffic Engineering Tests

3.1.1 Introduction

In this section we describe results from two Inter-domain TE test groups:

- Genetic Algorithm, implementing both delay and bandwidth as QoS metrics. This algorithm is decoupled from any intra-domain TE algorithms. In the presented results, the algorithm is compared with random assignment and brute force approaches;
- Heuristic Algorithms, implementing bandwidth only as a QoS metric, and including intra-domain route optimisation. This set of algorithms implements inter- and intra-domain TE in an integrated approach.

3.1.2 Genetic Algorithm for Decoupled Inter-domain TE

3.1.2.1 Overview

This Section describes the results of the Inter-domain traffic engineering tests specified in section 5 of [D3.1].

3.1.2.2 Experiment Set-up and Test Description

The functional tests were conducted using a Very Small Network Topology (Figure 1). This is a degenerate case where there is no intra-domain TE and hence the testing of the Inter-domain TE functions is separate from the behaviour of any Inter-domain TE software.

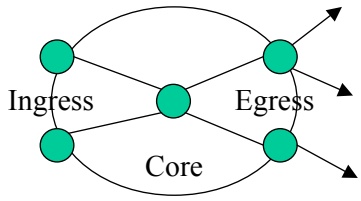


Figure 1: Very Small Topology for functional tests

The network topology for the simulations is shown in Figure 2, and focuses on the inter-domain connectivity. We assume a moderate sized AS with 20 adjacent ASs. The AS under test supports two l-QCs (25ms and 50 ms delays), and as a result of its service planning wishes to offer two e-QCs for its inter-domain flows (100ms and 175ms). Each adjacent AS is connected to the AS under test by either 1 or 2 links, giving a total of 27 inter-domain links, each of whose link bandwidth is set in the range 150-300 units. A number of destination prefixes are reachable through each adjacent AS (there may be other ASs en-route to the final destination prefix, but these are not relevant to our model). Each AS is able to reach between 30 and 60 of the prefixes. This reflects the observation that a small number of destination prefixes are responsible for a large fraction of an AS' outbound traffic volume [Feam03]. Although in reality the destination prefixes will in general overlap each other, for simplicity here we assume they are disjoint.

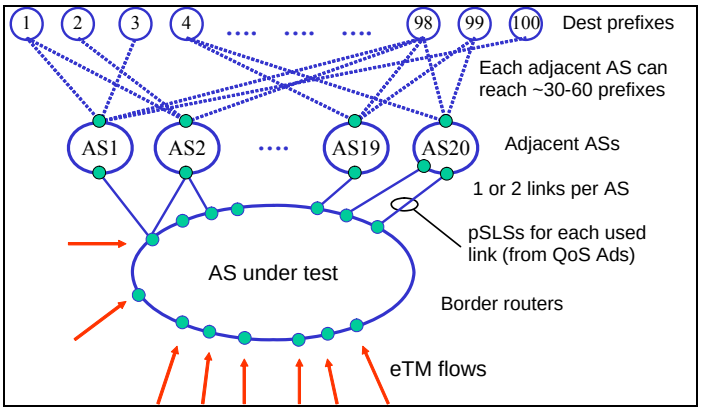


Figure 2: Simulation network topology for benefit / cost performance tests

Each adjacent AS is assumed to support a subset of three downstream o-QC delays. For simplicity, the set of supported delays is identical in each adjacent AS, being set to any of 50, 75 and 125ms (Figure 3). QoS advertisements for each link are generated based on a random combination of downstream o-QCs and random pSLS costs; for the QoS advertisements announced by any individual adjacent AS, the cost of a higher QoS class (i.e. lower delay) is set higher than the cost of a lower QoS class. Each pSLS has a bandwidth in the range 0 to 300, and the pSLS cost is set to a value between 1 and 10 per unit bandwidth. This results in overbooked pSLSs that support a total bandwidth that is 1.9 times the inter-domain link capacity. In the evaluation described here, each QoS advertisement is assumed to have resulted in the establishment of a pSLS, resulting in a total of 47 pSLSs being available to the 20 adjacent ASs. Finally the entire system is driven by a set of eTM flows randomly generated in such a way that the destination prefix in each eTM entry can be reached through one or more pSLSs supported by at least one adjacent AS. Each flow requires either a 100ms or 175ms e-QC to one of the 100 remote destination prefixes, and has a bandwidth requirement randomly selected in the range 1 to 40.

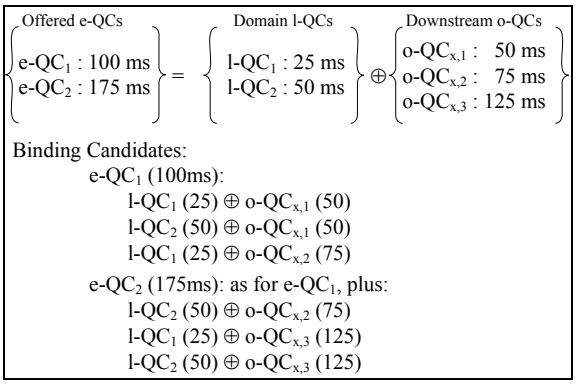


Figure 3: QC Mapping

Three cost functions were used in the Inter-domain tests, two of which represent inter-domain parameters and the third was used to represent Intra-domain costs. The cost functions were as follows:

- pSLS cost Ω, representing the cost per unit bandwidth of all pSLSs to which the domain is subscribed. We assume a subscription cost for a pSLS to be proportional to the bandwidth used.
- Inter-domain link utilisation cost function $\Theta = \sum_j \theta(x_j)$, where $\theta(x_j)$ is based on the Fortz and Thorup piecewise linear cost function [For02], and reflects the desirability of minimising the inter-domain link utilisation.
- Intra-domain TE cost Φ that reflects the cost of using Intra-domain resources. In order to decouple Intra-domain and Inter-domain effects, and to implement a decoupled algorithm (i.e.

inter-domain TE and intra-domain TE algorithms are independent) a simple illustrative model was used that reflects the higher cost of using low-delay l-QCs:

$$\Phi = K \sum_{flows} \frac{bandwidth}{delay}$$

In our tests we used a variety of cost function combinations; the Inter-domain cost defined in [D1.3] is to be read as either the pSLS cost Ω or the Inter-domain link utilisation cost function Θ or their sum, as appropriate.

3.1.2.3 Test Results

3.1.2.3.1 Functional Tests

The Binding Selection and Inter-domain Resource Optimisation function blocks were designed, coded and tested.

Test Id	Purpose	Result
InterTE/Funct/BSel	Binding Selection functions	Successfully completed.
InterTE/Funct/IDRO/1	General operation of Inter-domain Resource Optimisation	Successfully completed.
InterTE/Funct/IDRO/2	Random algorithm functions	Variation 1 successfully completed.
InterTE/Funct/IDRO/3	Brute force algorithm functions	Successfully completed.
InterTE/Funct/IDRO/4	Genetic algorithm functions	Successfully completed.
InterTE/Funct/System	System functional tests: interworking between Binding Selection / Binding Activation and Inter-domain Resource Optimisation	Successfully completed.

3.1.2.3.2 Algorithm Benefit/Cost Performance Tests

The behaviour of the algorithm was validated by considering a *simplified* set of QCs, in which only a single downstream o-QC is employed. The delay values in this validation were a single e-QC (150ms), three l-QCs (25, 30, 40ms), and a single downstream o-QC per adjacent AS (all o-QCs=100ms). We assume that the intra-domain links have sufficient capacity to carry all flows. We simplify the problem by assuming that all destination addresses in the eTM can be carried by the set of lowest cost pSLSs that have a total bandwidth equal to the total bandwidth in the eTM. Finally we relax the problem constraints by allowing a single eTM flow to be partially assigned to more than one pSLS. The result of these simplifications is to decouple the task of assigning an l-QC to a given eTM flow from the task of pSLS selection, and an analytically solvable approximation to the problem can be produced. In this case, all flows are optimally carried within the AS using the cheapest l-QC (i.e. the one with the highest delay), and for the Inter-domain link the flows are all assigned to the set of lowest cost pSLSs. By considering only the two cost functions pSLS cost and Intra-domain TE, we can calculate using a spreadsheet a lower bound cost, shown in Figure 4 by the solid line. This lower bound cost is better than the brute force solution. The motivation for calculating a lower bound is to observe how close the genetic algorithm approaches this simplified approximation. We see that the genetic algorithm produces results close to this lower bound.

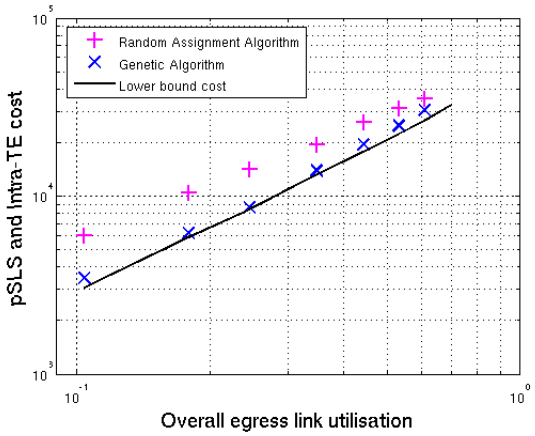


Figure 4: Validation of full-scale tests (single e-QC)

We now present results for the test scenario of Figure 2 with the full set of QoS classes shown in Figure 3. The Genetic Algorithm divides the population into three classes [D1.3]: we set the best to be the top 35%, the middle class to be the next 35% and the bottom class to be the bottom 30%. In producing each generation all members of the bottom class are discarded and replaced with child chromosomes one of whose parents is from the top class and the other parent from the middle class. We used a population size of N=250 chromosomes, crossover probability p_c=0.6 and mutation probability p_m=0.05.

We first consider only two cost functions: pSLS cost Ω and Intra-domain TE cost Φ. Figure 5 shows how the sum of these costs varies as the total eTM traffic increases. The x-axis is normalised by dividing the total eTM flow by the sum of the capacities of the Inter-domain links. The genetic algorithm has a lower cost than the random assignment algorithm at all values of utilisation. We note in passing that the brute force algorithm is only computationally feasible at very low utilisation, and that at this point, the genetic algorithm solution successfully matches the cost of the brute force solution.

In essence, the genetic algorithm identifies solutions where a flow can be assigned to a low-cost combination of l-QC and downstream o-QC. A destination prefix is in general reachable with a given downstream o-QC through a number of different pSLSs, and each of these pSLSs is offered by an adjacent AS at one of a number of different pSLS costs. The genetic algorithm identifies the pSLS with the lowest cost.

We can observe this behaviour by analysing the utilisation of each pSLS. In Figure 6 the 47 pSLSs are shown, arranged in ascending order of cost per unit bandwidth. For each pSLS, the assigned bandwidth is shown for the random assignment algorithm and for the genetic algorithm. We see that the random assignment algorithm has distributed the flows over all pSLSs approximately evenly. However, the genetic algorithm has weighted the flows towards the lower cost pSLSs. In fact, the random assignment algorithm has assigned only 18% of the traffic to the pSLSs with cost per unit bandwidth of 2.2 or less, whereas the genetic algorithm has assigned 85% of the traffic to these pSLSs.

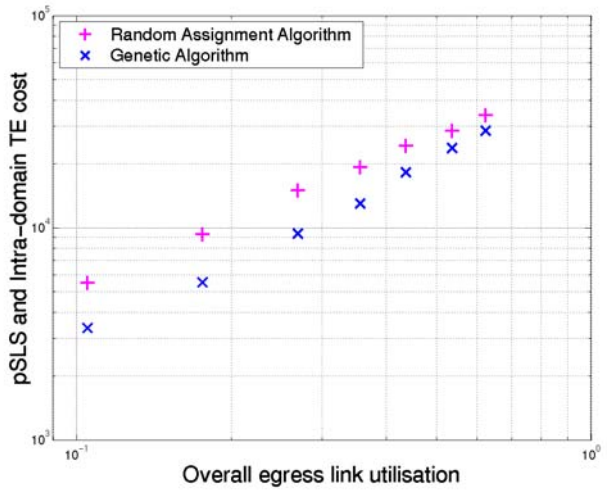


Figure 5: pSLS cost plus Intra-domain TE cost (Ω+Φ)

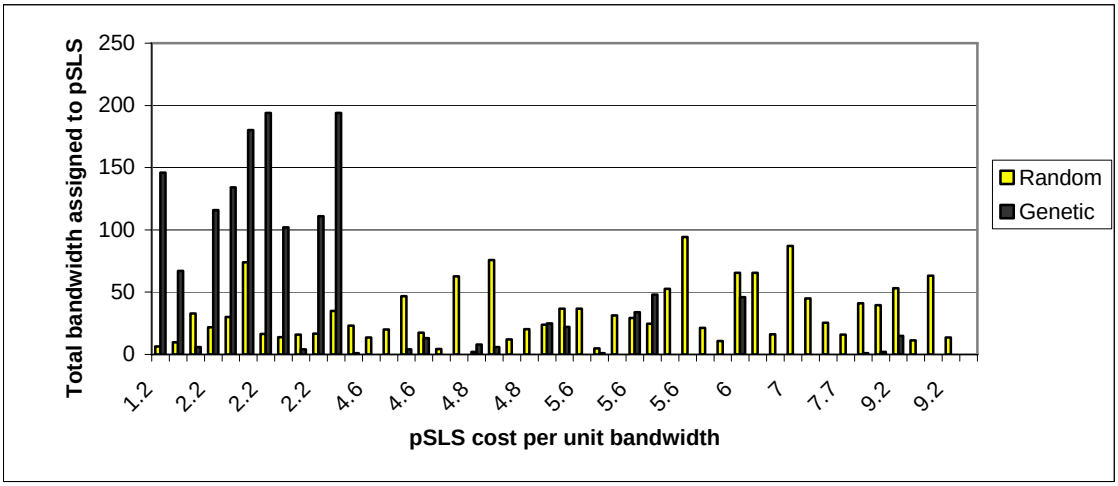


Figure 6: Comparison of pSLS utilisation in random and genetic algorithms (pSLS cost Ω and Intra-TE cost Φ only)

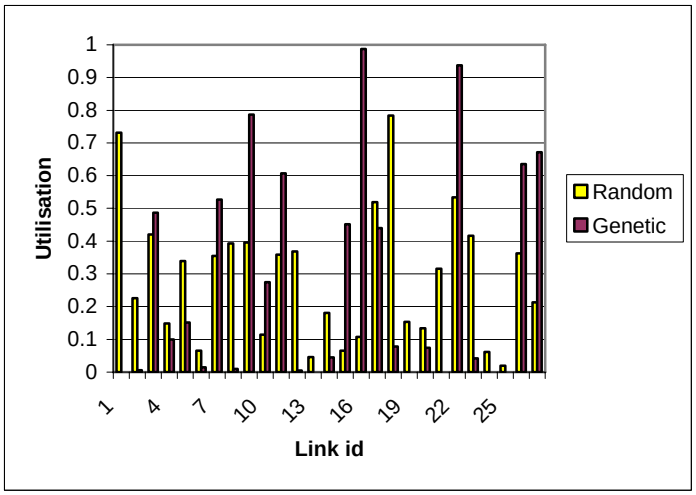


Figure 7: Link utilisations (pSLS cost Ω and Intra-TE cost Φ only)

However, the flow assignments are made without consideration of the maximum inter-domain link utilisation and have resulted in the genetic algorithm assigning flows such that some links are heavily utilised (Figure 7). This can be corrected by introducing the third cost function Θ so that the total cost function is the sum of the pSLS cost, Intra-domain TE cost, and Inter-domain link utilisation (Figure 8). The link utilisation cost function is scaled so that all three components are given approximately equal weight. By introducing the link utilisation function, the peak link utilisations are reduced (Figure 9), with the worst link utilisation from the genetic algorithm reduced from 99% to 69%.

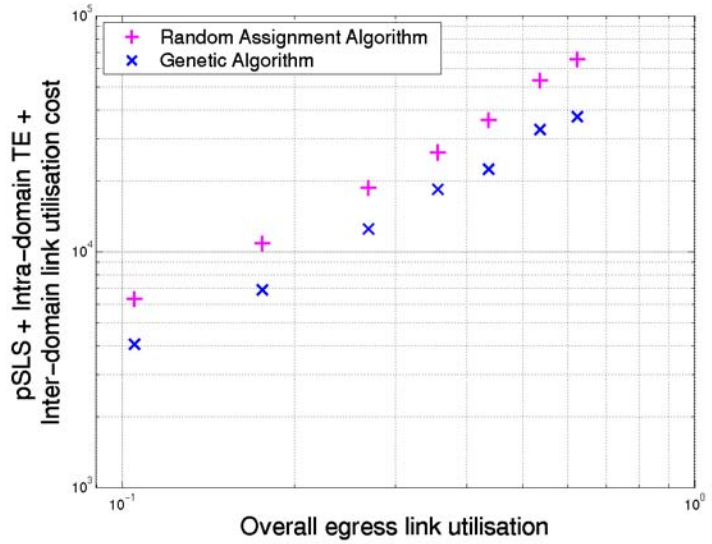


Figure 8: pSLS cost, Intra-domain TE, and Inter-domain link utilisation costs (Ω+Φ+Θ)

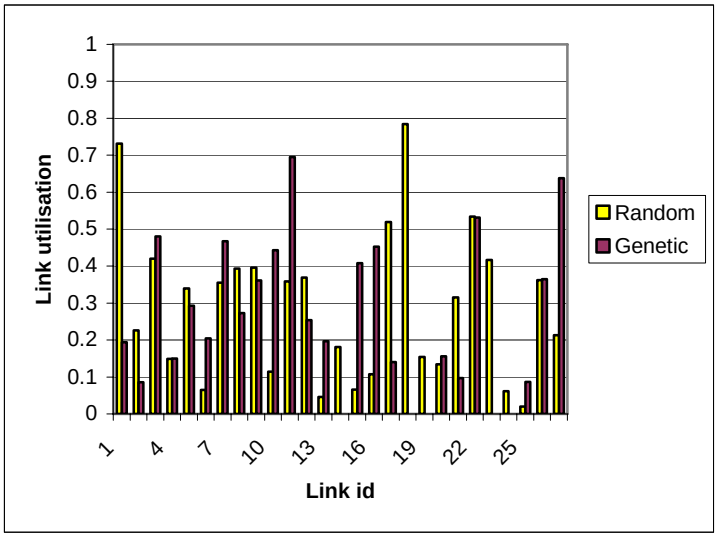


Figure 9: Link utilisations (based on Ω+Φ+Θ)

3.1.2.3.2.1 Impact of Controlled Variables on Performance of Genetic Algorithm

We assessed the performance of the Genetic Algorithm as a function of its controlled variables. We used the same test scenario as that described above for validation (Figure 4): i.e. a single e-QC (150ms), three l-QCs (25, 30, 40ms), and a single downstream o-QC value (100ms) per AS.

To assess the impact of the crossover probability p_c, three versions of the genetic algorithm were compared:

- Crossover probability p_c=0.6, child chromosomes based on one "best" parent and one "middle" parent (both randomly chosen);
- Crossover probability p_c=0.9, child chromosomes based on one "best" parent and one "middle" parent (both randomly chosen);
- Crossover probability p_c=1.0, child chromosome based on *single* parent chosen randomly from "best"/"middle" groups.

To enable fair comparison of these versions, all results were obtained for a fixed length run of 250 generations (and a population size of N=250 chromosomes). It should be noted that the GA results at high utilisation can be improved still further by increasing the number of generations.

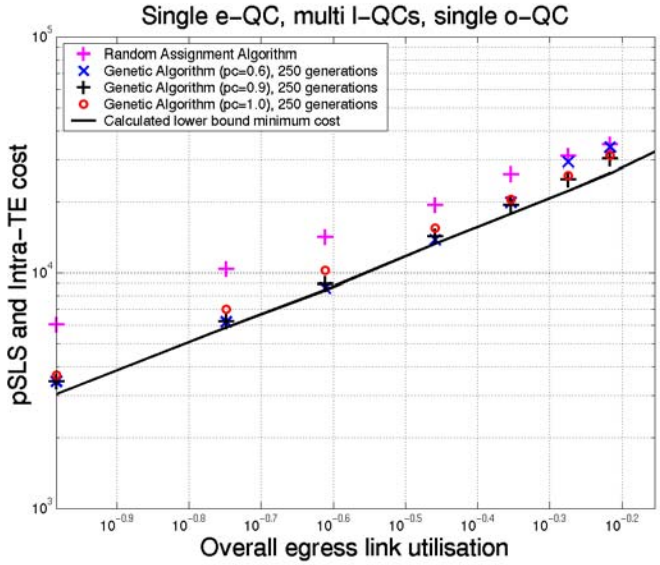


Figure 10: Impact of Genetic Algorithm parameters: variation of p_c

The results are compared in Figure 10. This graph shows that at low utilisation, selecting either crossover probability of p_c=0.6 or 0.9 makes very little difference. However, at high utilisation (>0.75) p_c=0.9 give much better results than p_c=0.6. We believe that this is because at low utilisation there are many valid solutions that give a low cost, and mixing two solutions results in a valid child solution. However, at high utilisation the solution space is much smaller: retaining the bulk of a consistent solution (by setting p_c=0.9) means that the algorithm is able to converge and reduce costs much faster than mixing two possible inconsistent solutions (p_c=0.6). It is further to be noted that p_c=0.9 always gives better results then p_c=1.0: in this final case, there is no evolutionary / genetic component, and the algorithm is relying purely on random mutations to develop improved solutions. However, the random mutations occur slowly (we set p_m=0.03 here) and so convergence is slow.

We also investigated values of mutation probability p_m in the range 0.01 to 0.05 and found no general difference in the algorithm convergence or final results.

Test Id	Purpose	Result
InterTE/Perf/1	Performance assessment of genetic algorithm	Successfully completed. Performance assessed for GA parameters: crossover and mutation probabilities, pc and pm. Performance assessed for impact of different cost functions pSLS cost, Intra-domain TE cost, and inter-domain link utilisation cost.
InterTE/Perf/3	Performance comparison of algorithms	Successfully completed.
InterTE/Perf/4	Performance assessment of interactions	Successfully completed.

3.1.2.3.3 Scalability Tests

We investigated the elapsed run time for each algorithm as a function of the number of flows in the eTM (Table 2). This shows that as expected the brute force approach is not scalable and is not applicable for any realistic configuration. The random assignment algorithm runtime increases with eTM size because it only assigns flows to pSLSs that have sufficient spare capacity; if a flow cannot be assigned the solution is discarded and a further attempt at randomly assigning flows is made [D1.3].

Number of eTM rows	3	4	5	30	50	75	100
Percentage utilisation	0.8%	0.9%	1.2%	11%	18%	28%	36%
Brute Force Assignment runtime	0.1 mins	5 mins	>24 hrs	-	-	-	-
Random Assignment runtime	-	-	-	2 secs	2 secs	3 secs	4 secs
Genetic Algorithm runtime	-	-	-	5 mins	15 mins	50 mins	180 mins

Table 2: Scalability: offline Inter-domain TE runtime as function of egress link utilisation

Given the observations that the Genetic Algorithm with p_c=1.0 gives reasonable (but not excellent) results (Figure 10) and that the random assignment algorithm runs very quickly (Table 2) we investigated the hypothesis that an algorithm that selects the lowest of several randomly chosen configurations might provide an approach that gives a good solution with a moderately fast run time and might provide a scalable solution.

For a GA with N=250 chromosomes, running for 250 generations, with the bottom 30% of the population being replaced in each generation, a total of 19 000 chromosomes are generated. We therefore ran an algorithm which selects the lowest cost solution from 19 000 randomly generated solutions. The results are compared with the GA and random assignment algorithms in Table 3. This shows that while the lowest of 19 000 random solutions is significantly better than the random assignment algorithm, the Genetic Algorithm produces the best solutions.

By assuming that the probability density function of the costs is a normal distribution, an estimate can be made of the number of random solutions required to approach the GA solution. The random solutions can be printed to a spreadsheet, where their mean and standard deviation are found to be 10 300 and 450 respectively. The solution space contains approximately (3*12)^50 = 10^{78} solutions (since each of the 50 rows of the eTM can be assigned to any of 3 l-QCs and on average any of 12 pSLSs). To find a solution that is 6 standard deviations from the mean (i.e. 10 300 - 6 * 450 = 7 600) requires ~ 10^9 random guesses. This algorithm is therefore not scalable, and we therefore conclude that the Genetic Algorithm provides the superior approach in a way that is scalable.

Algorithm	pSLS and Intra-TE cost (Ω+Φ)
Genetic Algorithm, p_c=0.6	6 200
Genetic Algorithm, p_c=0.9	6 200
Genetic Algorithm, p_c=1.0	7 000
Random Assignment	10 300
Lowest of 19 000 random solutions	8 800

Table 3: Comparison of approaches (overall egress utilisation=18%, single e-QC validation scenario)

3.1.2.4 Conclusions

We compared three algorithms for offline QoS-aware traffic engineering: a random assignment algorithm (effectively representing current day best-effort inter-domain traffic engineering practices applied to a QoS-aware environment); a brute force assignment algorithm; and an evolutionary Genetic Algorithm.

We have shown that in a simplified validation scenario the genetic algorithm obtains results that are close to an analytically obtainable lower bound solution. We have also demonstrated that in a more complex scenario the GA can be used to obtain offline QoS-aware traffic engineering solutions that are of significantly lower cost than a random approach; and that we can reduce the maximum inter-domain link utilisation by representing this utilisation in the cost function, minimising the total of the Inter-domain pSLS costs, Intra-domain TE costs and Inter-domain link utilisation costs.

3.1.3 Heuristic Algorithm for Integrated Inter-/Intra-domain TE

3.1.3.1 Overview

In this section we describe the performance tests of the offline inter-domain TE heuristic algorithms. These algorithms provide an integrated approach between inter-domain and intra-domain TE. They implement bandwidth as the single QoS parameter, and we consequently use the total bandwidth consumption as the performance metric. The total bandwidth consumption is defined as the sum of bandwidth needed on each link in order to accommodate the projected eTM.

3.1.3.2 Experiment Setup and Test Description

We evaluate the three proposed heuristic algorithms through simulation. The simulation results are based on 100-node transit domain topologies. The topologies are randomly generated by the method described by Waxman. The set of ingress and egress routers are disjoint. We set the number of ingress routers to 30, whereas the number of egress routers is a variable, as we will evaluate some effects by changing its value between 10 and 30. Each egress router is attached to a maximum of two inter-domain links. We assume that the inter-domain resource is less than that of intra-domain resource. The capacity of each link within a domain is randomly generated between 400 and 500, and the capacity of each inter-domain link is randomly generated between 250 and 300.

Feamster [Feam03] discovered that a typical default-free routing table may contain routes for more than 90,000 prefixes, but only a small fraction of prefixes are responsible for a large fraction of the traffic. Based on this finding, we consider 1000 routing prefixes. As these routing prefixes are usually popular destinations, we assume that each egress router can reach all of them. This set of routing prefixes is randomly distributed on the inter-domain link(s) of each egress router. Each routing prefix is advertised with available bandwidth randomly generated between 200 and 250.

For each customer traffic flow, the destination prefix and the ingress router are randomly generated and its bandwidth requirement is randomly generated between 10 and 40.

3.1.3.3 Test Results

Figure 11 presents the total bandwidth consumption as a function of the number of eTM customer traffic flows under the three proposed greedy-based heuristic algorithms. This simulation is based on the scenario of 30 egress routers. The Greedy-penalty heuristic consumes less bandwidth than the others because it considers the penalties of all unassigned customer traffic flows and determines which of these flows, if assigned in the first place, can avoid consuming additional bandwidth. On the contrary, the Greedy-cost heuristic does not take this into consideration and often results in a greater penalty in terms of consuming more bandwidth. As the Greedy-random heuristic randomly selects an egress router without considering any optimisation, any efficient egress router selection algorithms should always outperform it.

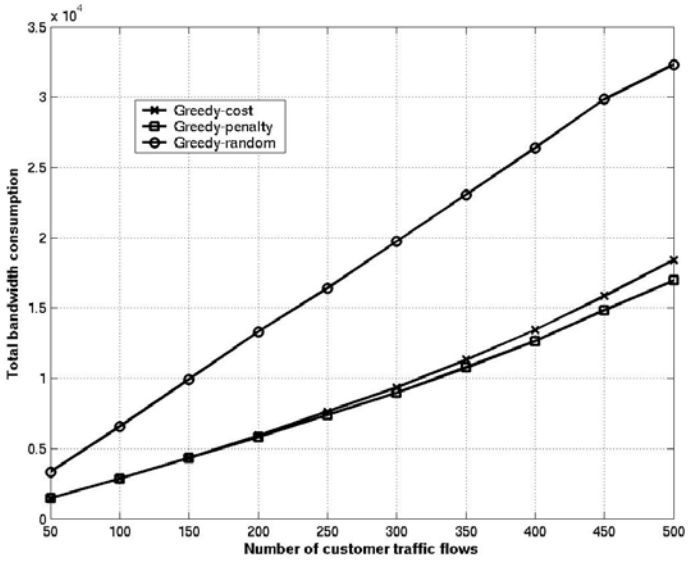


Figure 11: Total bandwidth consumption as function of traffic

In Figure 12, we show the difference of bandwidth consumption between the Greedy-cost and Greedy-penalty heuristics for a different number of egress routers. We study the bandwidth consumption difference under three traffic loads with 100% acceptance ratio at any considered number of egress routers: 50, 100 and 150 customer traffic flows. The bandwidth consumption difference is the total bandwidth consumption using the Greedy-cost heuristic minus the total bandwidth consumption using the Greedy-penalty heuristic. It is worthwhile to determine the improvement of bandwidth consumption when using the Greedy-penalty heuristic over the Greedy-cost heuristic.

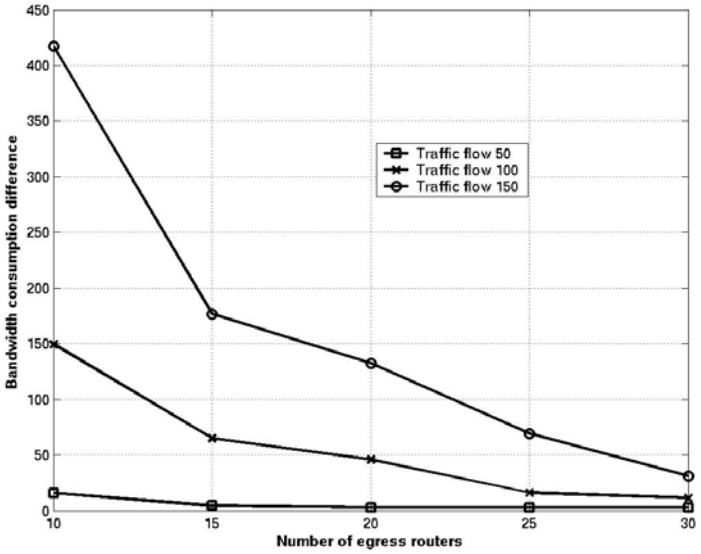


Figure 12: Bandwidth consumption difference between Greedy-cost and Greedy-penalty heuristics

When the number of traffic flows increases, the bandwidth consumption difference between the two heuristic algorithms increases. This can be explained by the case that, as traffic load to the egress routers increases, some egress routers do not have sufficient resource so that some customer traffic flows are directed to the "distance" egress router with possible great penalty in terms of consuming more bandwidth. It is the case where Greedy-penalty heuristic is used to avoid additional bandwidth consumption.

Something else that can be deduced from the figure is that as the number of egress routers increases, the bandwidth consumption difference decreases. This is the opposite effect to the previous one, with the aforementioned case occurs less frequently as more capacity is added. As a result, the two heuristic algorithms are likely to have same selection for traffic flows and the performance of them tends to become identical.

From the above, we conclude that the Greedy-penalty heuristic provides significant performance improvement over the Greedy-cost approach, under the situation where the network has a certain level of loading in order to take the advantage of penalty-based selection, and that no more than one egress router can preferentially accommodate most of the traffic flows while leaving the other egress routers barely selected. The latter situation is achievable due to the fact that resources are commonly distributed in the network for load balancing.

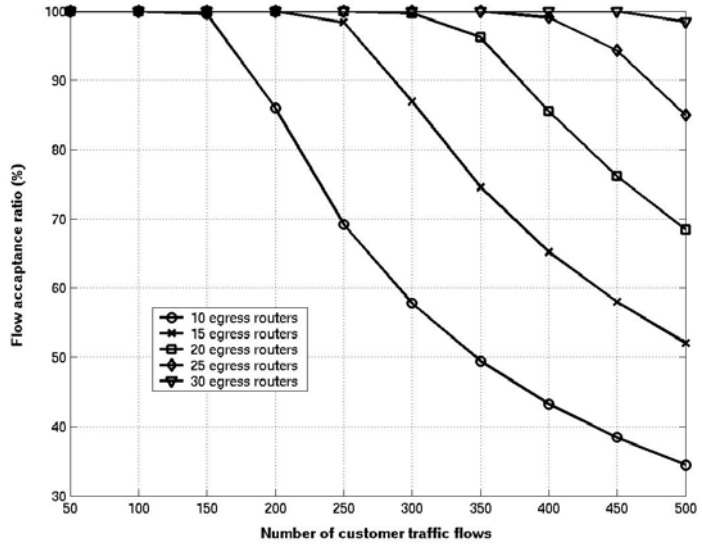


Figure 13: Bandwidth acceptance ratio for Greedy-penalty heuristic

For the rest of simulations, we continue to study the performance as the number of egress routers varies. As the Greedy-penalty heuristic outperforms the others, we only consider this one. Figure 13 shows the influence of the number of egress routers on the bandwidth acceptance ratio. The bandwidth acceptance ratio is the sum of bandwidths of accepted traffic flows over the sum of bandwidths of all the traffic flows. As the number of egress routers increases, the bandwidth acceptance ratio increases. This is due to the property that performance improves as more capacity, such as inter-domain link and advertised bandwidth capacity, is added by increasing the number of egress routers. It is also worthwhile to determine when the bandwidth acceptance ratio reaches a level of diminishing return.

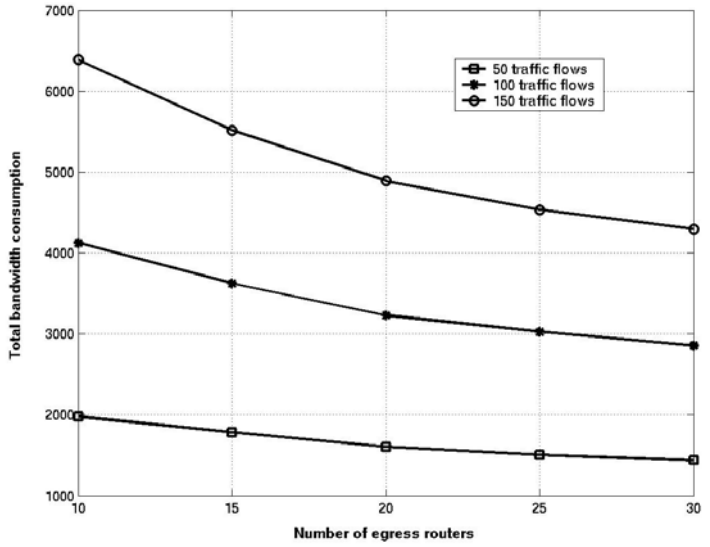


Figure 14: Total bandwidth consumption vs. number of egress routers for Greedy-penalty heuristic

To evaluate the influence of the number of egress routers on the total bandwidth consumption, we study the bandwidth consumption under three traffic loads as they were previously used: 50, 100 and 150 customer traffic flows. Figure 14 shows the total network bandwidth consumption with a different number of egress routers. For all the traffic flows, as the number of egress routers increases, the total bandwidth consumption decreases. This is because, as the number of

egress routers increases, the traffic flow can be directed to a "closer" router which results in reduced bandwidth consumption. This effect becomes more apparent when the number of traffic flows is large since the traffic load of each egress router is high, while adding additional egress routers can significantly improve the performance. On the contrary, this effect is less apparent when the number of traffic flows is small.

3.1.3.4 Conclusions

We have developed three heuristic algorithms to solve the integrated inter-domain / intra-domain TE problem. Simulation results show that the Greedy-penalty performs better than the other two algorithms in terms of total network bandwidth consumption. We have also evaluated the influence of the number of egress routers on the total bandwidth consumption and bandwidth acceptance ratio. We found that the total bandwidth consumption decreases and the bandwidth acceptance ratio increases as the number of egress routers increases.

3.2 Offline Traffic Engineering Interactions

3.2.1 Overview

In this section we describe the objectives, performance metrics and experimentation environment for the performance tests of the interactions between offline intra- and inter-domain TE.

In [D1.3], we proposed two approaches, namely the decoupled and integrated approaches, to combine offline intra- and inter-domain TE. The objective of the performance tests is to assess the performance of the decoupled and the integrated optimisation approaches.

3.2.1.1 Assumptions

For the proposed heuristic algorithms for the decoupled and integrated approaches, there are of course many possible algorithm or solution combinations for the two approaches. However, since this paper is not intended as a comparative study of these options, we will propose a classical greedy-based heuristic algorithm as the TE algorithm for the decoupled and integrated approaches. The proposed heuristic algorithms are similar to that proposed by Xiao [Xiao00] which has been deployed in a real network system. The proposed algorithms for both approaches are very similar in order to accurately compare their TE performance. Although it might be appealing to test some more complex algorithms, the approach presented here is sufficient to illustrate the point of interest. For simplicity, but without loss of generality, we make the following assumptions for our algorithm and evaluation:

- Only outbound and transit traffic are considered.
- Bandwidth is considered as the QoS metric.
- The inter-domain resource objective to optimise is the inter-domain link utilization, and the outbound provider SLA is used as capacity constraint.
- Explicit routing is assumed and bandwidth constrained minimum cost routing algorithm is used for intra-domain route selection, where the cost is dynamically calculated for each considered traffic flow by the piece-wise linear cost function proposed in [For02]. This not only minimizes resource consumption but also attempts to achieve load balancing within the network. The granularity of explicit paths is per-prefix.
- The AS under consideration has sufficient capacity to meet the end-to-end bandwidth requirements of all inter-domain traffic flows. Thus, the traffic-oriented TE objective can be negligible.

3.2.1.2 Performance Metrics

We use the following performance metrics as the optimisation criteria to evaluate the decoupled and the integrated approaches: (i) Total network cost (the sum of intra-domain and inter-domain cost), (ii)

total bandwidth consumption, and (iii) maximum intra-domain and inter-domain link utilization. The first metric captures the overall network cost. Overall intra-domain (respectively inter-domain) cost is defined as the sum of the cost of the intra-domain (inter-domain) links. Fortz and Thorup [For02] propose a piecewise linear increasing function of link utilization which imitates the response time of M/M/1 queue to access the cost of intra-domain links. By using the piecewise linear cost function, two objectives of bandwidth usage and resource load balancing are taken into account simultaneously. These two objectives are related to our second and third performance metrics. In other words, the overall network cost is a function of both bandwidth consumption and link utilization.

In this paper, we adopt the piecewise linear function to quantify the cost of intra-domain and inter-domain links. Since inter-domain links are the bottleneck in the Internet [Akam99], we assume that the cost of using them is a factor α times the cost of intra-domain links. We assume $\alpha=2$ as our initial evaluation in this paper. The impact of α on network performance will be evaluated in our future work.

The total bandwidth consumption is the amount of bandwidth needed to accommodate all traffic flows within an AS. It is calculated based on the bandwidth requirement of each traffic flow and the length of path on which the traffic flow has been assigned.

The utilization of a link is the amount of traffic on the link divided by its capacity. The maximum link utilization is the maximum utilization over all links in a network. Minimizing this objective ensures that traffic is moved away from congested to less utilized links and the distribution of traffic is balanced over the links [Wang99].

For all three metrics, the lower values are preferred.

3.2.2 Experiment Setup and Test Description

The simulation is based on 100-node topologies generated by BRITE with node degree of 4. The number of border routers is set to 30% of the total network nodes. Note that inter-domain links can be ingress or egress links, and we only consider egress links for outbound TE in this paper. Without loss of generality, we assume that each border router is attached to a maximum of three egress links and the capacity of each egress link is randomly generated between 150 and 300 units. The capacity of each intra-domain link is randomly generated between 80 and 200 units. As inter-domain links are usually the bottleneck in the Internet, the total capacity of all intra-domain links should be larger than that of all inter-domain links.

Due to the fact that only a small fraction of prefixes are responsible for a large fraction of the traffic [Feam03], we consider 100 popular remote destinations which are uniformly and randomly distributed over all the border routers. The number of remote destinations that each border router can reach, specified in outbound provider SLAs, is randomly generated between 30 and 60 units, and these remote destinations are randomly distributed among all the egress links. The contracted bandwidth for a remote destination is randomly generated between 30 and 60 units.

For each aggregated inter-domain traffic flow, the remote destination and the ingress router are randomly generated. The bandwidth demand of each aggregated inter-domain flow is randomly generated between 1 and 40 units.

To ensure confident results, each simulation point takes an average value based on 10 trial runs.

3.2.3 Test Results

Figure 15 and Figure 16 show the inter-domain and intra-domain cost as a function of number of inter-domain traffic flows achieved by the decoupled and integrated approaches respectively. The inter-cost achieved by the two approaches is nearly identical. This is because the cost of using inter- AS links is higher than that of intra-domain links, so the inter-domain link utilization becomes a dominant factor in the selection decision in both approaches. It is possible that there are several inter-domain links that have very similar utilization, but the intra-domain routes connected to them may have different costs. In this case, the integrated approach can select the best combination of inter-domain links and intra-

domain routes. We see in Figure 16 that the performance difference between the two approaches is primarily in their intra-domain cost.

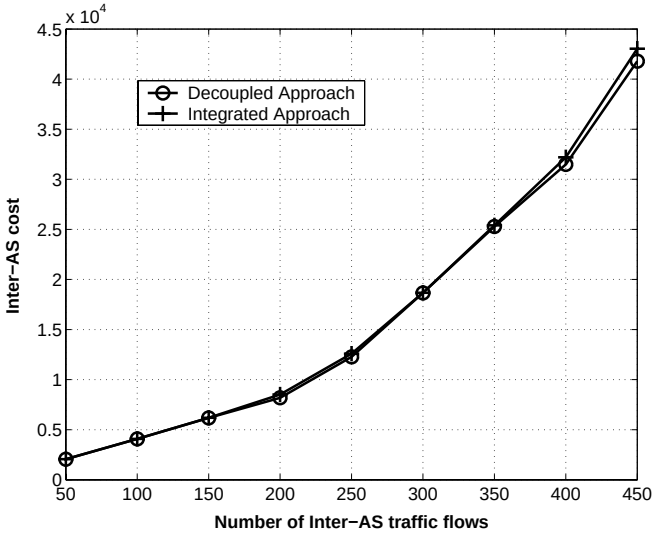


Figure 15: Evaluation of inter-domain cost

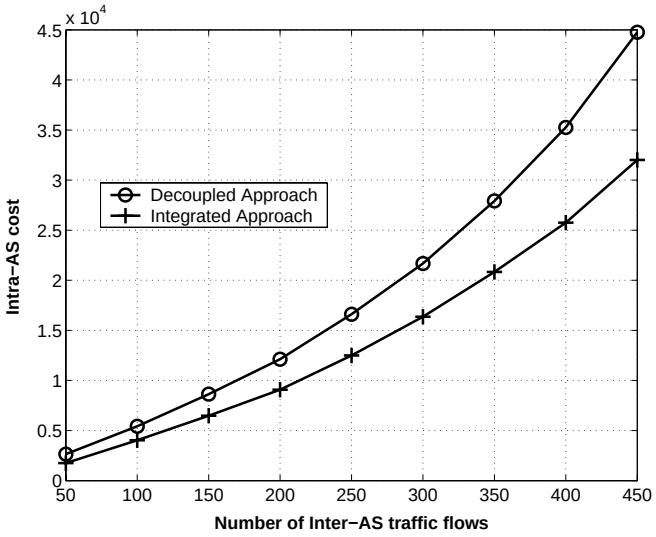


Figure 16: Evaluation of intra-domain cost

The total network cost is defined as the sum of intra-domain and inter-domain cost. Since the inter-domain cost achieved by both approaches are nearly identical, the total cost will mainly depend on the intra-domain cost. Hence, the total cost achieved by the integrated approach is much lower than that achieved by the decoupled approach. This resembles the performance shown in Figure 16.

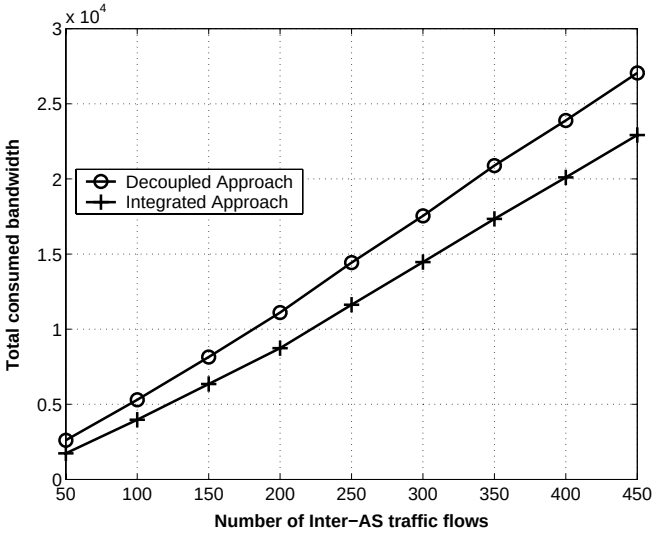


Figure 17: Evaluation of total bandwidth consumption

A major reason for the large intra-domain cost in the decoupled approach is due to the increase in bandwidth consumption and link utilization within an AS. Figure 17 shows that the integrated approach uses less bandwidth to accomplish its end-to-end QoS provisioning within the network than the decoupled approach. This is because, when choosing egress routers, the number of hops on the corresponding intra-domain routes has been considered as the selection criteria. The decoupled approach on the other hand may choose an egress router with the best inter-domain link utilization but at the expense of long intra-domain route towards the egress router, resulting in high bandwidth consumption.

Although Figure 18 shows that the integrated approach has a slightly higher maximum inter-domain link utilization than the decoupled approach, both approaches incur nearly identical inter-domain costs, as shown in Figure 15. This may result partially from the piecewise linear cost function, which gives the same penalty to links with utilizations in the same block, such as between 1/3 and 2/3. In this case, such links are considered as at the same level of congestion. Based on the fact that both approaches result in nearly identical inter-domain TE performance, Figure 19 shows that the integrated approach exhibits the advantage of significantly reducing the maximum intra-domain link utilization, compared to the decoupled approach.

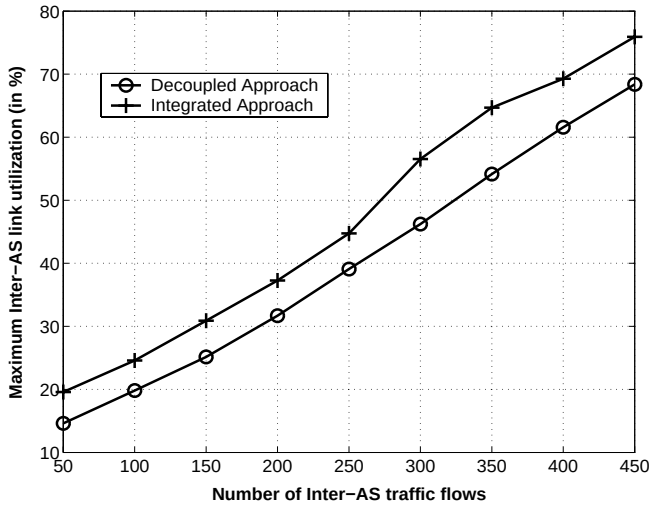


Figure 18: Evaluation of maximum inter-domain link utilization

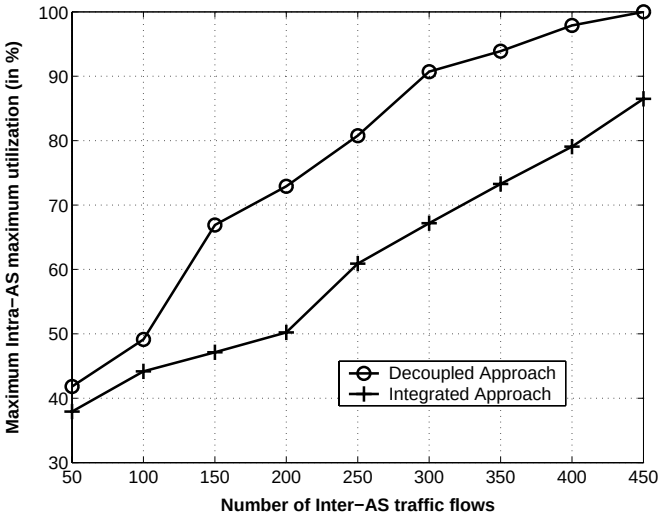


Figure 19: Evaluation of maximum intra-domain link utilisation

As the decoupled approach performs inter-domain TE prior to intra-domain TE, utilization performance on inter-domain link is good compared to that on intra-domain link. On the other hand, the integrated approach takes the balanced approach optimizing between intra-domain and inter-domain resource utilization, therefore the achieved inter-domain resource utilization may not be good as that achieved by the decoupled approach. Nevertheless, significant improvement in intra-domain utilization achieved by the integrated approach compared to the decoupled approach offsets this minor degradation in inter-domain resource utilization.

To compare the overall performance achieved by the decoupled and integrated approaches, our numerical experiments reveal that the integrated approach could save a significant amount of resource cost and achieve a good overall network resource performance, compared to the decoupled approach. Hence, we attempt to answer the question posed in the introduction section by introducing the integrated approach to achieve lower cost complete TE solution.

In fact, other factors can also affect the performance of the two approaches, such as the efficiency of algorithms, the definition of link cost function (linear, concave or discrete), network size and topology, etc. Further experiments are needed to understand their impact on traffic and resource utilization performance.

3.2.4 Conclusions

We have established a direct relationship between intra-domain and inter-domain TE, and explored the interaction between them by proposing and analysing both the decoupled and integrated approaches. We have shown through simulation how the integrated approach results in lower cost TE solutions with lower total consumed bandwidth.

3.3 Intra-domain Traffic Engineering Tests

3.3.1 Overview

MESCAL's intra-domain traffic engineering approach is based on layer 3 mechanisms (rather than MPLS-TE, for example). Its purpose is to compute a set of OSPF link weights to balance network load while honouring the QoS constraints of the traffic; and to provide answers to "what if" scenarios posed by Inter-domain Traffic Engineering in order to coordinate and optimise inter-domain and intra-domain traffic engineering decisions.

The IPTE approach is built on classical OSPF routing, but additionally introduces DSCP based routing to form multiple OSPF routing planes in the network. Each DSCP plane has individual link weights and can thus route traffic independently of the other planes. Each plane may be used to route traffic of

an equivalent QoS-class to meet the performance constraints of that class. Another benefit of this approach is that multiple routing planes – even for a single QoS-class – allow for better load balancing across an AS.

The IPTE algorithm runs off-line at Resource Provisioning Cycle epochs. Given a traffic demand matrix and the network topology, the algorithm computes a set of link weights using a search heuristic. The optimisation is cost function based, so that individual QoS class constraints as well as other optimisation goals can be taken into account by factoring them into the algorithms cost function. This allows for parallel existence of hop-count-constrained, bandwidth-constrained and best effort traffic classes.

Because the solution relies on IP routing, the IPTE approach is more lightweight than MPLS-TE in terms of state-information required to be maintained in the network and the associated management configuration overhead for establishing LSPs. Since QoS information remains at the management layer in the off-line algorithms, no QoS awareness is required at layer 3. Recent MT-OSPF Internet Drafts provide the required DSCP based routing support, so potentially no major changes at the router level are required for the approach to be implemented.

This section gives a detailed description of the simulations that have been carried out for the Intra-domain Traffic Engineering component described in the MESCAL functional architecture. The campaigns focus predominantly on the *Resource Optimisation* block contained within Intra-domain Traffic Engineering. *Resource Optimisation* contains the essential link weight based IP traffic engineering functionality. In contrast, functionality contained within the *Resource Reconfiguration Scheduler* block is concerned with the efficient implementation of results computed by *Resource Optimisation* and is therefore of secondary concern. The test campaigns have three overall objectives:

1. Functional Validation
2. Algorithm Performance Measurement and Optimisation
3. Algorithm Efficiency Measurement and Optimisation

3.3.2 Experiment setup and test description

This section contains a description of the simulation setup and the input data used for each test. The network shown in Figure 20 resembles the early NSFNet backbone. It was used as test topology for functional testing as well as for load spread simulations to determine the effectiveness of routing planes.

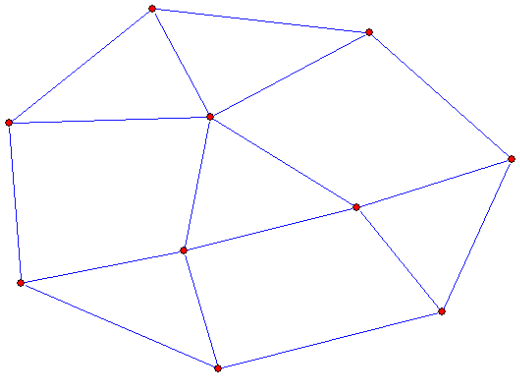


Figure 20: 10 Node Test Network

Larger topologies were also used ranging from 50 to 300 nodes. Topologies were generated using the BRITE [Brite] topology generator. A list of topologies used with more details is displayed in Table 4.

Number of Nodes	Number of Links	Topology Type	Min, Max Link Capacity	Link Capacity Distribution	Node Distribution	Average Degree
10	17	NSFNet	2, 5	exponential	heavy tail	**3.2**
50	100	Waxman	10, 1024	exponential	heavy tail	**4**
100	200	Waxman	10, 1024	exponential	heavy tail	**4**
300	400	Waxman	10, 1024	exponential	heavy tail	**4**
300	**600**	**Waxman**	**10, 1024**	**exponential**	**heavy tail**	**4**

Table 4: Topologies used for Simulations

As in [Fortz00] demands were generated according to

$$\alpha O_u D_v C_{uv} e^{-\frac{\lambda(u,v)}{2\Delta}}$$

Where O and D $\in [0,1]$ are random numbers chosen for each node. Similarly, C $\in [0,1]$ is chosen for each pair of nodes u, v. The parameter $\lambda(u,v)$ denotes the Euclidian distance between u and v and Δ is the maximum Euclidian distance between two nodes. This ensures that demands are greater between nodes with shorter Euclidian distance. Also, since there are three random numbers multiplied, the variation between demands is large. For the simulations, three demand matrices were generated for each topology size with small, medium and large numbers of individual demands; they are detailed in Table 5.

Nodes in Topology	Number of Demands			**Demand**
	Small (#)	Medium (#)	Large (#)	**α factor**
10	N/A	35	N/A	**N/A**
50	900	1300	3000	**1**
100	900	1300	3000	**5**
200	900	1300	3000	**10**
300	**900**	**1300**	**3000**	**15**

Table 5: Demands used for Simulations

The demand set for the 10 node topology was created manually; not using the method described above, but rather by applying individual demands across some of the network edges.

3.3.3 Test Results

3.3.3.1 Algorithm Performance and Optimisation

3.3.3.1.1 (Perf1) Load Balancing Performance

The load balancing performance of the IPTE system is important to give the network more flexibility towards changes in the demand pattern. The more evenly balanced the network, the less traffic engineering changes have to be made over time in order to achieve operational goals of the network operator. The plots in Figure 21 show the results of an IPTE optimisation cycle run on a 100 node topology, utilising 5 routing planes. Both mean utilisation and standard deviation of utilisation are shown. As comparison, values for inverse capacity link weights are also plotted. Inverse capacity link weights are the Cisco recommended configuration for OSPF networks.

Plotting the standard deviation of the link utilisation gives a measure of how well loads are balanced across the network. The mean utilisation is the mean of individual link utilisation values and thus gives a measure the overall network utilisation.

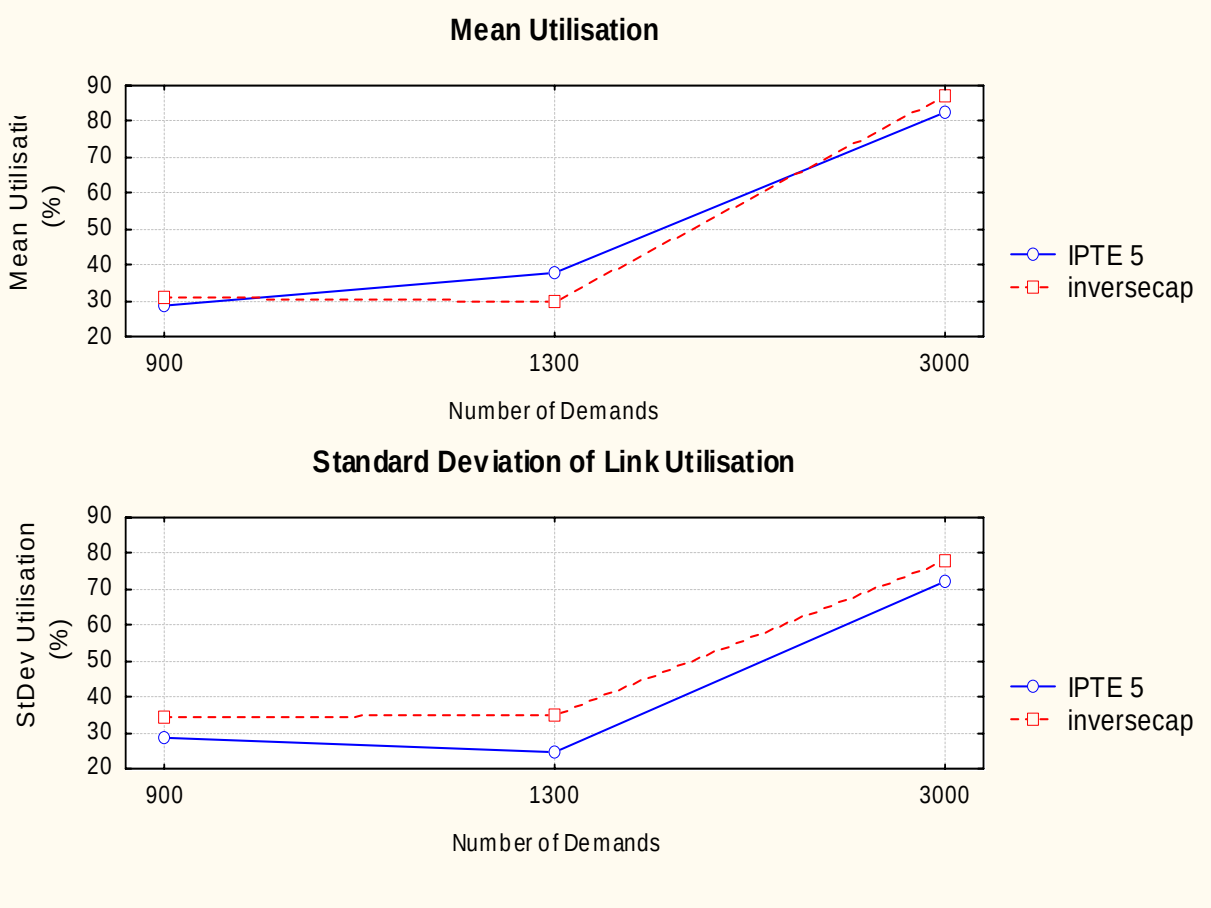


Figure 21: Load balancing improvement on a 100 node network, after 500 iterations

Whereas the mean utilisation increases from 30% to 80% for the IPTE case, the standard deviation decreases slightly from 30% to 40% utilisation and then increases as the mean utilisation is increased to 80%. The IPTE solution stays about 10% below the inverse capacity link weight settings. This shows that better load balancing is achieved.

3.3.3.1.2 (Perf4) Routing Plane Effectiveness

Since there are 64 routing planes available, not all of which are necessary for MESCAL inter-domain QoS, spare planes may be used to split traffic of the same class for the purpose of intra-domain load balancing. The tests in this category were thus designed to investigate how the individual routing planes can be employed for the purposes of load balancing. Figure 22 shows the results simulated on the 10 node topology.

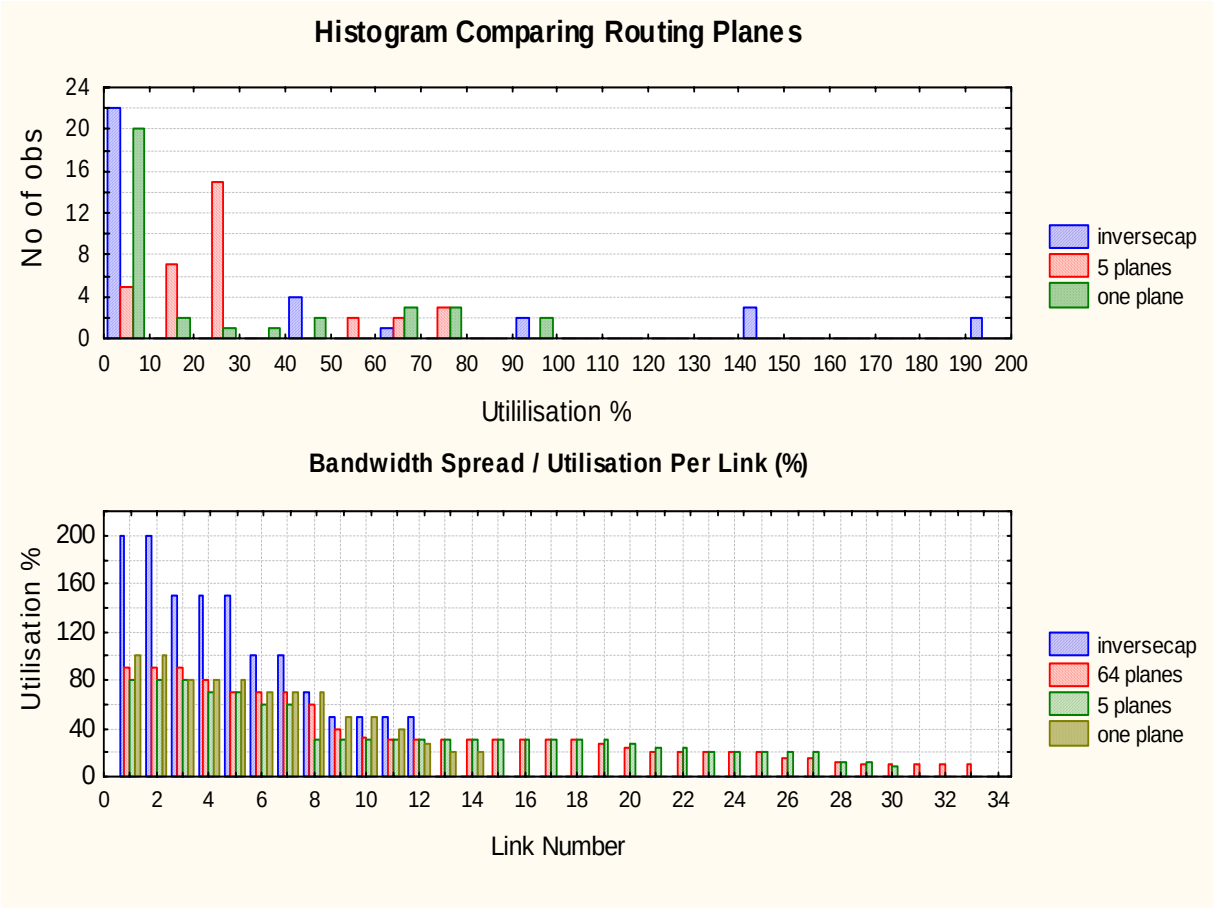
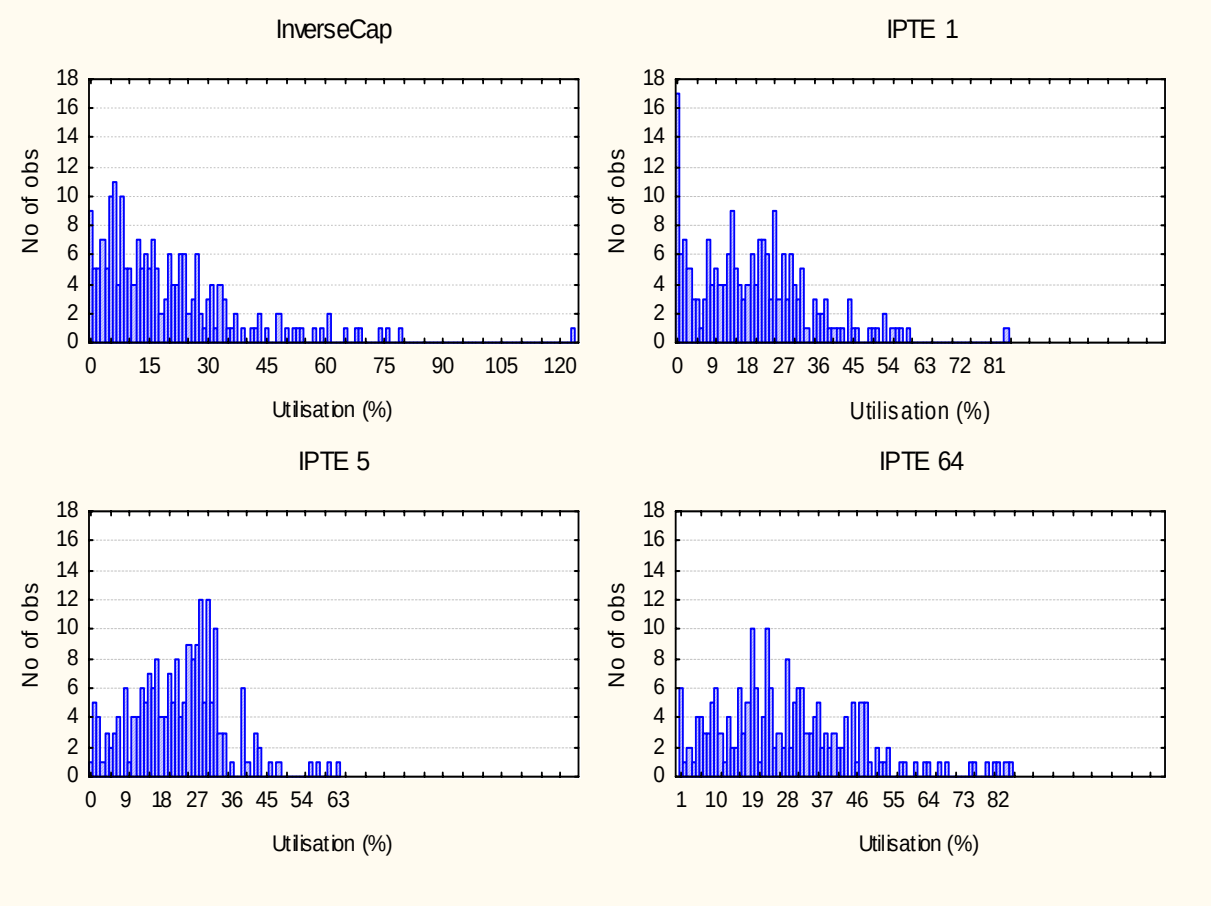


Figure 22: routing plane effectiveness on a 10 node network, 500 iterations

From the graph it can be seen that even a single routing plane based IPTE link weight optimisation can provide large gains in terms of load balancing in this case. Overloaded links from the Inverse Capacity routing disappear almost completely. However, several links remain underutilised with IPTE 1 (single routing plane) and some links still remain at near 100% utilisation. Performing the optimisation on 5 routing planes shows that the demands are now more effectively balanced, with many of the available links utilised and maximum link loads of less than 80% in this example. Interestingly, increasing the number of routing planes to 64 does not bring further improvement (second graph). Whereas only the 64 plane IPTE utilises all links, maximum link utilisation increases slightly, compared to the IPTE solution with 5 planes. Considering that a 10 node 17 link network does not offer many alternative paths, this result seems intuitive. In fact, even larger networks with 50+ nodes of the type generated for these simulations do not appear to benefit from more than 5 routing planes as can be seen on the graphs in Figure 24 which show the results for the 50 node topology.

Each of the four graphs shows a histogram of the utilisation of individual links on the network for Inverse Capacity and IPTE with 1, 5 and 64 routing planes with additional statistics displayed in the table below the plot. Again IPTE 1 shows significant improvement over inverse capacity weights, removing the overloaded link and significantly decreasing the utilisation Standard Deviation. The IPTE with 5 routing planes shows the best performance, with a further decrease in standard deviation accompanied by an increase in utilisation mean. This shows that IPTE 5 indeed performs load balancing more effectively than a single routing plane optimisation is able to. More links are utilised and the average utilisation increases, yet the maximum utilisation decreases (to 64% compared to 84% for IPTE 1 and 123% for InverseCap). IPTE 5 also removes the peak of underutilised links that featured in both InverseCap and IPTE 1.



Utilisation statistics				
Type	**Mean (%)**	**StdDev (%)**	**Max (%)**	**Min (%)**
InverseCap	19.9789	18.0273	123.2366	**0**
IPTE 1	19.9368	14.715	84.8083	**0**
IPTE 5	23.021	11.6104	63.9776	**0.8214**
IPTE 64	**28.9738**	**17.9128**	**85.3099**	**1.145**

Figure 23: Effect of Routing Planes on Utilisation for a 50 Node Network, 1300 demands

IPTE 64 does improve on the result of InverseCap, but not as significantly as either IPTE 1 or IPTE 5, which appears to be counterintuitive. However, one possible cause for this behaviour could be that IPTE 64 has a larger probability for arriving at local minimums than solutions with less routing planes. While iterations with 5 routing planes have an effect on the routing of up to 1/5 of the traffic, with 64 routing planes it is only up to 1/64. The optimisation algorithm may therefore track towards a near local minimum with marginal per-iteration improvements, from which it cannot escape without non improving moves (Perturbations are available as non-improving moves. However, these may be too coarse for the purpose). The Convergence performance of the IPTE 64 scenario supports this theory, by quickly arriving at the final solution with no further improvement for several 1000 iterations after that. IPTE 1 on the other hand shows the longest improvement time, with improvements still being occurring after 4500 iterations without plateau effect (see section 3.3.3.2.1 for details on plateau effect). In order to get maximum benefit from more routing planes, it thus seems necessary to modify the heuristic approach with increasing numbers of planes to compensate for the effect described.

For comparison, the cost function values range from 35358.0 for InverseCap to 4014 for IPTE 1, 2955 for IPTE 5 and 9264 for IPTE 64.

3.3.3.1.3 (Perf5/Perf6) QoS Constrained Performance

This section shows how individual routing planes can be optimised with different performance goals using the cost function. Two types of this individual treatment are shown: hop count constraint and utilisation constraint. The hop count constrained optimisation can be used for delay sensitive traffic classes, whereas the utilisation constrained optimisation can be used for bandwidth constrained classes. More elaborate QoS based optimisations could be devised, the results in this section are a demonstration of the feasibility of the approach.

Hop Count Constrained Optimisation

Before performing the hop count constrained optimisation it is important to realise that whereas the optimisation algorithm is based on identification of high cost links, hop count is an end-to-end feature. Thus, in order to factor hop count constraints into the cost function it is necessary to calculate the end-to-end path of each demand on a link. Once the hop-count has been determined, a per-link cost can be calculated based on routing plane membership, which is then added to the equivalent utilisation of the link as the sum of d_h over all (delay constraint) routing planes. (For more details on the build up of the basic cost function, see [D1.3], section 10.6.2.3.9)

$$\Phi = \sum_{l \in E} \Phi_l \left(\frac{\sum_{h \in H_l} (f_{l,h}(x_{l,h}))}{c(l)} + \sum_{h \in H_l} d_h \right)$$

Since the starting link weights are unit weights, the shortest path is already configured for each demand. It is thus important to ensure that the algorithm does not make these paths longer for the purposes of load balancing and so the cost calculated above helps the optimisation heuristics to identify when a link weight modification has caused a hop count limit to be exceeded. The resulting increase in cost should lead to a discarding of this modification. This ensures that hop counts are honoured while load is balanced.

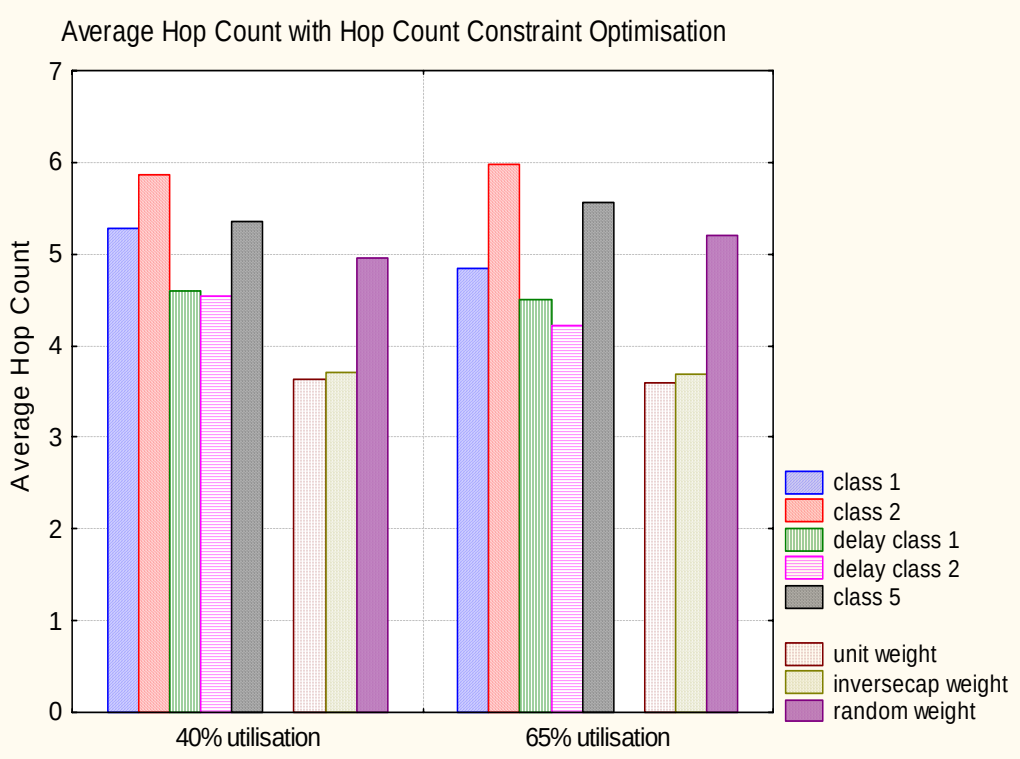


Figure 24: Average hop count for 50 node 100 link network, 500 iterations, 1300, 3000 demands

The graph in Figure 24 shows the effect of hop count constrained optimisation for a 50 node 200 link network. The two delay classes were hop count constrained, using a sharp maximum hop count cut-off of 7 for class 2 (i.e. a sudden increase in cost from 0 to 5000 between 6 and 7) and a gradual cost increase from 5 to 7 for class 1. The maximum shortest path distance across the network is 8 hops. Classes 1-5 (including delay classes 1 and 2) are operated in parallel on the same network, whereas unit weight, inverse capacity weight and random weight are computed separately for the same network topology and demand matrix. Several observations may be made from this graph. Firstly, the hop count for all classes remains approximately the same for different network load. This should be expected, for all reasonable loads, as long as the IPTE algorithm has sufficient free capacity to operate. Secondly, it can be seen from the graph that non-delay constrained classes have high average hop counts. This is also as expected, since longer average paths are an effect of the load balancing on these classes. Finally, the graph shows that the two delay classes have lower average hop counts than the non hop-count constrained classes. The delay classes perform worse than the both inverse capacity and unit weights, as a result of the costs specified in the cost function which is based on the absolute hop-count limitation of 6-7, rather than limitation based on relative path length. Maximum hop counts on both delay classes are equal to that of the shortest path on unit weights (because of the large cost applied for demands exceeding 7 hops), whereas all other optimised classes (1,2 and 5) have larger maximums (10-14).

Spare bandwidth constrained optimisation

Optimisation for higher average spare bandwidth on a routing plane is accomplished by increasing the equivalent bandwidth factor of the class. Higher equivalent bandwidth causes the utilisation to appear greater to the cost function than it is. As a result, the optimisation algorithm reduces the load on links occupied by the class more than average, leading to higher average available bandwidth.

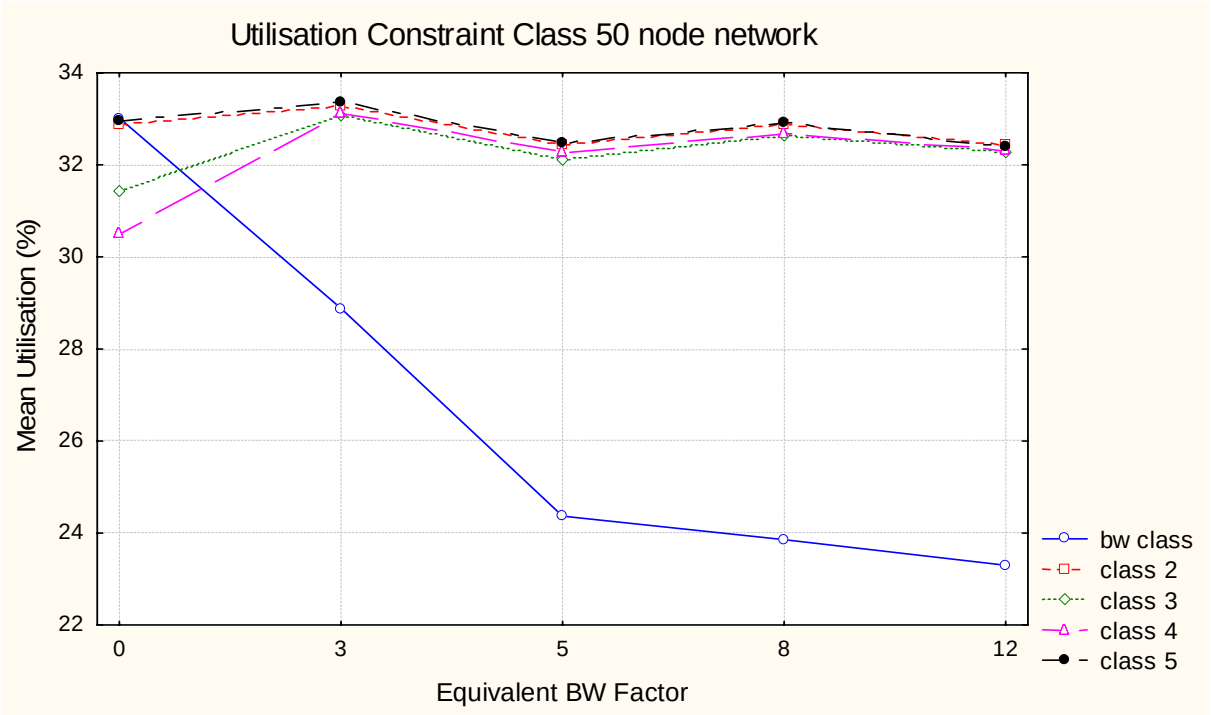


Figure 25: average utilisation for bandwidth constrained class, 1300 demands, 500 iterations

Figure 25 shows the effect of this method. The graph shows the mean link utilisation as seen by individual classes, i.e. the mean of utilisation (of all traffic) on all links that the class utilises. With all classes co-existing on the same network, Class 1 has lower mean utilisation than all other classes. The equivalent bandwidth factor is shown on the x axis. It demonstrates how the effect first increases rapidly and then levels off when the factor reaches 5. At this point, the utilisation constrained classes'

bandwidth requirements can no longer be met as efficiently through link weight optimisation, as bandwidth requirements begin to reach 100% link bandwidth.

It is also worth noting that the average link utilisation for all other classes stays approximately constant, which might be explained through the increase in both higher utilised links as well as lower utilised links on the network. Since some of the traffic on the non delay constrained classes also travels on the links with lower utilisation, the average of link utilisation remains approximately constant. Evidence for this is the increase in link utilisation standard deviation from 15.77 to 18.54 between equivalent bandwidth factors 0 to 8.

3.3.3.2 Algorithm Efficiency and Optimisation

3.3.3.2.1 (Effic5/Mixt) Convergence Properties

Measuring the algorithms performance is important in order to improve its convergence properties. The heuristic algorithm used for the IPTE system has to search a very large solution space that is a function of the number of links and routing planes. Thus, in order to achieve improvement over the inverse capacity/ unit link weight settings, a lot of work has to be invested into tweaking the heuristics. The plots in Figure 26 provide information on convergence. The top plot shows the improvement of cost function over iteration number plotted on a log scale, whereas the second and third plots track the mean and standard deviation of network utilisation. All graphs feature two plots, the best and the running value. The running value is the value computed for the current iteration and may be worse than the best value. If the value is worse, the new solution will be discarded at the beginning of the next iteration. The running cost value features large bumps caused by the “perturbation” feature of the algorithm, which perturbs the link weight set if no improvement is found to the best solution for some time. It is meant to enable the algorithm to escape from suboptimal local minimum solutions. After a perturbation begins, a new temporary best cost value is determined which causes the plateau-likeness of the large bums. Once a perturbation has completed it is discarded if unsuccessful or kept if an overall improvement was achieved.

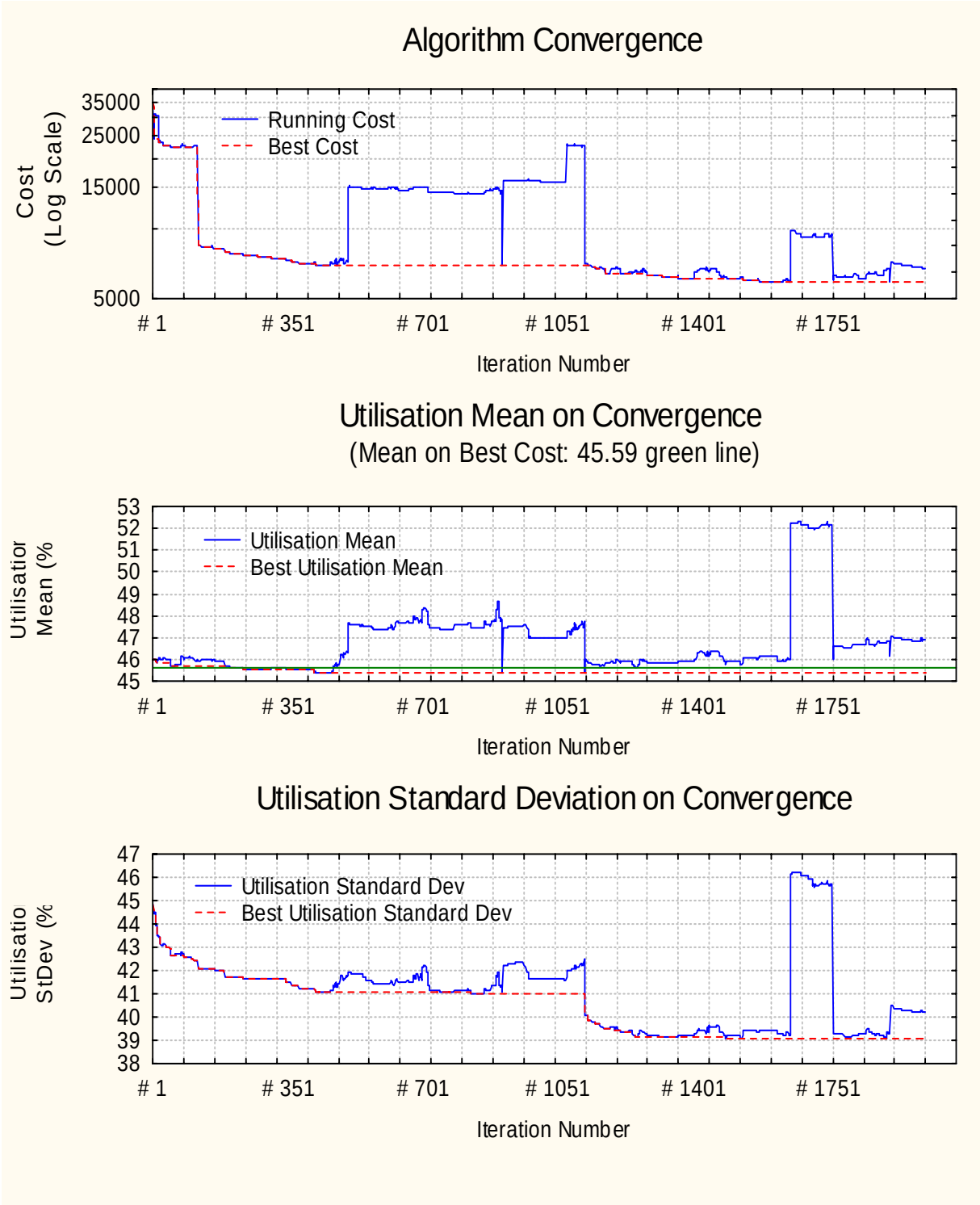


Figure 26: Convergence Efficiency for the 50 node topology, 1600 demands, 5 routing planes

The cost improvements are largest at the beginning of the optimisation cycle, but further improvement continues until the 1600th iteration in the depicted case. The standard deviation plot shows that these small improvements in cost have an equally large effect on load balancing than the first large drop in cost at the beginning of the optimisation. The reason for this becomes clear when recalling that the cost is based on an exponential function that assigns large values to overloaded links. These links are addressed by the optimisation in its first iterations causing large improvements on the cost plot. However, later iterations with less cost improvement nevertheless have high importance to load-balancing as can be seen on the standard deviation plot. The mean utilisation stays approximately

constant with a slight decrease in the beginning and then a slow increase over a long period. The drop is caused by reducing load on the most overloaded high cost links at the beginning of the optimisation. These links are most distant in utilisation value to the mean and thus removing them has a visible impact on the average. Load balancing over the course of the optimisation causes the slow increase in the mean, while more links are utilised and the paths that traffic takes become longer. An increase in mean utilisation should be expected when the load balancing is functioning effectively.

Further analysis of the algorithms convergence properties was performed on varying topology sizes, the results of which are shown in Figure 27. Plotted is the percentage improvement of link utilisation standard deviation from the starting value unit weight value.

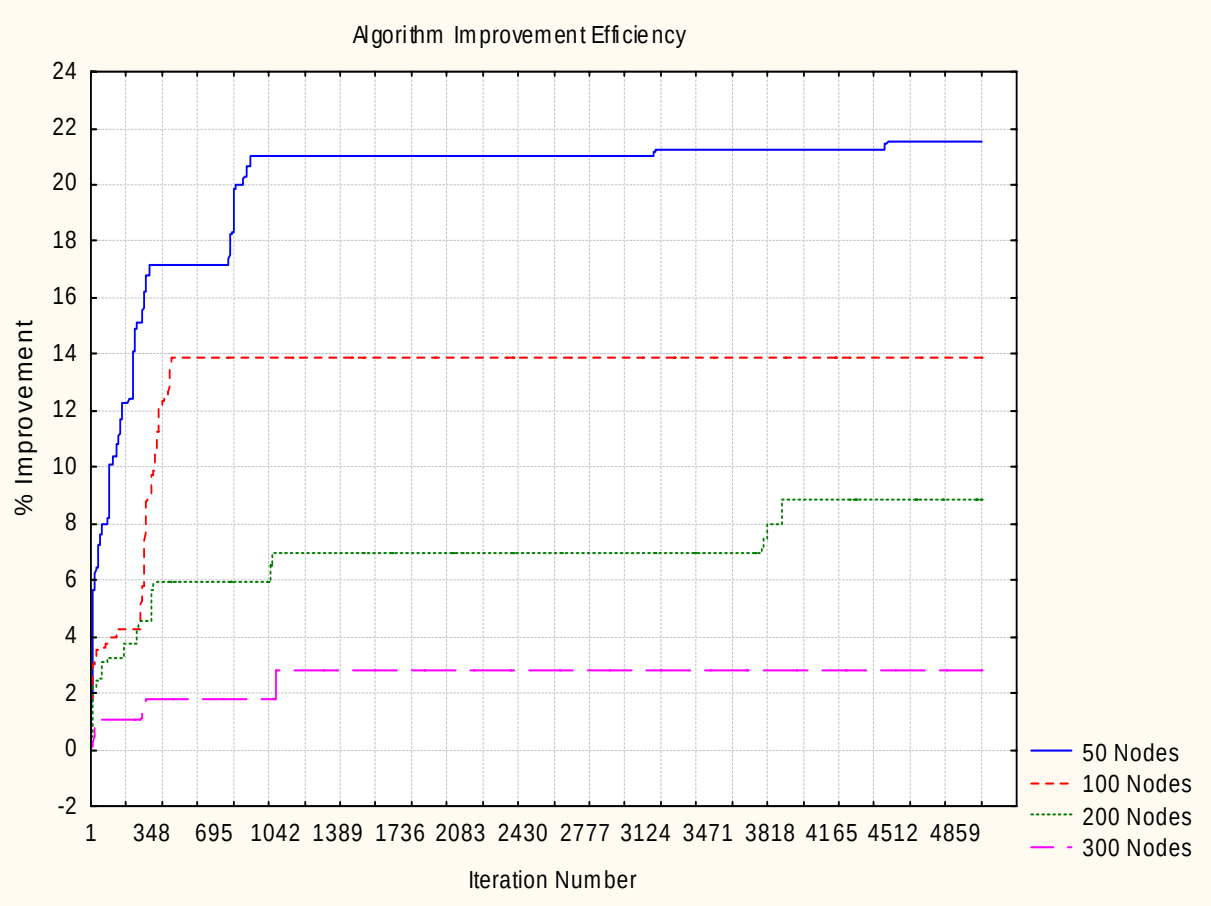


Figure 27: Algorithm efficiency measured on utilisation StDev, 60% average utilisation

As before, the results show that for all topology sizes there is a sharp improvement in the first few 300 iterations. However, the effect lessens with topology size with the 50 and 100 node networks benefiting most. With increasing topology size, the overall percentage improvement per iteration becomes smaller approximately halving per doubling of number of nodes. Improvements range from almost 22% for 50 nodes to only 3% for 300 nodes. As can be seen for the 200 node case, the optimisation for larger networks takes longer and significant improvement occurs at around 3800 iterations even after a long plateau of non-improvement. However, the heuristics devised for this study were not optimised for run times longer than a few thousand iterations required for the smaller topologies and runs longer than 5000 iterations are relatively inefficient. It would be more effective to adapt the heuristic for larger networks. Another likely cause for the lack of improvement on the two larger topologies is that only 1300 demands were in the demand matrix, thus causing a lack of improvement opportunities for the heuristic.

3.3.3.2.2 (Effic1/Effic2) Execution Time

In order to measure how long it takes for a result to be computed, the algorithm execution time has to be plotted against an improvement factor, such as the cost value or the load balancing measure of utilisation standard deviation. The plot in Figure 28 shows how the standard deviation of link utilisation improves over the run time of the optimisation cycle.

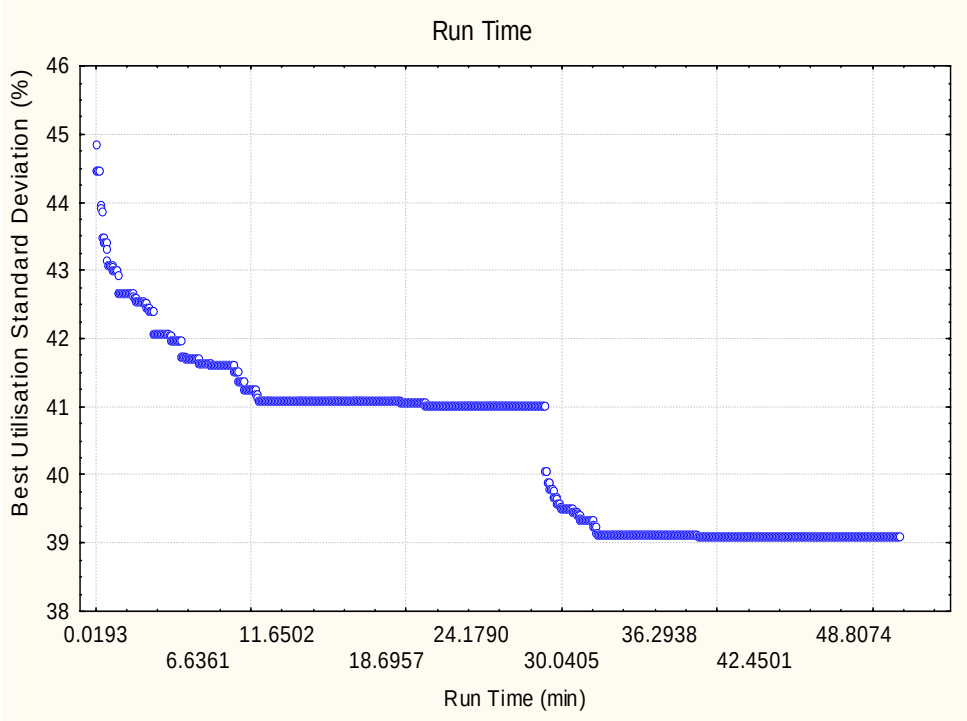


Figure 28: Convergence Time 50 Node Topology

For this case, the last improvement can be observed at around 32 minutes. However, since a long plateau was traversed before the improvement at around 28 minutes run time, it is not certain that no further improvement can be achieved beyond the 50 minutes of total run time. Execution time is significant for IPTE link weight optimisation. Whereas for this plot, 1800 iterations were computed on 1GHz CPU, larger topologies require 5000 and more iterations for large improvements to take place. However, results of this type are indicative as several factors that lead to the long execution time can be remedied. On more powerful hardware, such as a 3 GHz Pentium4, 5000 iterations for the same network configuration merely take 30 minutes. Additionally, the efficiency of the software implementation can be improved to reduce the algorithm run time. Finally, since offline intra-domain TE is an "offline" process, abundant time is available for processing.

3.3.4 Conclusions

IPTE has been tested in various networks ranging from 10 – 300 nodes with up to 600 bi-directional links. For each topology, three demand matrices were generated ranging from 900 to 3000 individual source destination demands.

Link weight optimisation on a single routing plane achieves better performance in terms of load balancing and avoidance of overloaded links compared to the inverse capacity link weight rule of thumb. It can be concluded that while inverse capacity is a good rule of thumb if the demand matrix is unknown, overloaded links and packet losses may occur when the traffic demand is high. With average utilisation figures of 30-40% on a 10-100 node network, off-line link weight optimisation can distribute traffic so that overloaded links do not occur while the same demand pattern on the same physical network will cause packet losses when random or inverse capacity link weight assignment

policies are used. Use of IP-TE can therefore relieve congestion without resorting to a potentially expensive reconfiguration of the physical network, e.g. installation of additional links. IP-TE implements its traffic redistribution policy through a soft configuration of the existing network, which can be achieved more easily and furthermore can be done periodically as traffic demand matrices change significantly, also accommodating temporary demand fluctuations.

While improvements over random and inverse capacity link weight assignment policies are achieved with the IP-TE link-weight optimisation heuristic, the use of multiple routing planes on the same physical infrastructure results in further gains in load balancing. Up to 64 independent routing planes are available with DSCP-aware routing and forwarding mechanisms deployed in the routers. Simulation results comparing the utilisation and load balancing performance of assigning demands to a number of parallel routing planes show that the use of 64 link-weight-optimised planes exhibits a significant improvement in load balancing compared to a single link-weight-optimised routing plane. However it has been demonstrated that using 5 parallel link-weight-optimised routing planes achieves a comparable performance.

The conclusions so far relate to a single QoS-class, e.g. the current best effort Internet. For QoS-enabled IP networks assumed by MESCAL an AS is required to implement multiple QoS-classes on the same physical infrastructure. We have demonstrated that the cost function of the link-weight optimisation heuristic can accommodate QoS-classes with different performance goals on different routing planes. Both hop-count- and bandwidth-constrained classes were considered and deployed on the same simulated network. The results show that link-weights for classes with different QoS targets can be derived and implemented on the same network through multiple routing planes. The resulting traffic distribution results show that delay-constrained classes take shorter paths and that bandwidth-constrained classes are routed over lower-utilised paths. Further investigations could study the impact of multiple equivalent routing planes for each QoS-class to determine whether the improvements in network load-balancing seen for a single QoS-class still hold. As it was seen that only 5 routing planes were required to achieve significant improvement over a single routing plane it can be seen that up to 64/5 parallel QoS-classes could be deployed within an AS, however, since traffic of all classes is shared on the same links it is unlikely that each QoS class requires 5 load balancing classes. Rather the different QoS classes should provide balancing between them if their QoS constraints are not too stringent.

With an iterative heuristic-based optimisation it is difficult to determine whether the algorithm has converged. Experimental results have shown that significant improvements in cost, utilisation and load-balancing are achieved after relatively few iterations of the algorithm (order of 100) for networks in the order of 10 – 100 nodes with 5 routing planes. Larger networks require more iterations (order of 1000) to reach significant improvement. Similarly if only a single routing plane is deployed, improvements may be observed after 1000s of iterations. However, even for smaller networks with 50-100 nodes and 5 routing planes, further significant improvements are occasionally observed at higher iterations (order of 1000) when a random perturbation of the current best solution finds an alternative set of solutions. It was seen overloaded links were removed early in the solution (order of 100 iterations), improvements in load-balancing were the main reason behind the smaller reductions in cost-function values at iterations in the order of 1000s. The reason for this is due to the high-cost associated with high link utilisation that causes the algorithm to quickly reroute traffic to remove overloaded links.

Algorithm run time was shown to scale approximately linearly with network size. Typical run times on a 3GHz Pentium 4 processor were around 30 minutes for 5000 iterations on a 50 node network with 1300 demands, however run time increases with topology size and the same run with a 300 node network can take up to 2 hours. The IP-TE optimisation heuristic is not intended as an on-line traffic control algorithm and therefore the execution time is acceptable for periodic off-line traffic engineering purposes.

3.4 Multicast Traffic Engineering Tests

3.3.1 Offline Dimensioned Test

3.4.1.1 Overview

We provide in this section the test results from the simulation software for the Offline Multicast Traffic Engineering (OMTE) specified in section 7 in [D1.3]. The objectives and controlled/uncontrolled variable settings are same as those included in [D3.1].

3.4.1.2 Experiment Setup and Test Description

We adopt the Waxman's model in GT-ITM topology generator for constructing our network models. This approach distributes the nodes randomly on the rectangular grid and nodes are connected with the probability function:

$$P(u,v) = \lambda \exp(\frac{-d(u,v)}{\rho L})$$

where $d(u,v)$ is the distance between node u and v and L is the maximum possible distance between any pair of nodes in the network. The parameters λ and ρ ranging (0, 1) can be modified to create the desired network model. A larger value of λ gives a node with a high average degree, and a small value of ρ increases the density of shorter links in comparison to longer ones. In our simulation we set the values of λ and ρ to be 0.2 respectively, and generate a random network of 100 nodes, out of which 50 are configured as Designated Routers (DRs) with attached group sources or receivers. The scaled bandwidth capacity of each link is set to 10^5 units. Apart from the GA approach, we also implemented two non-TE based hop-by-hop routing approaches and one explicit routing approach: (1) shortest path routing with random link weight setting (Random), (2) shortest path routing in terms of hop-counts (SPH), and (3) Steiner tree approach using the TM heuristic. For this TM Steiner tree algorithm, we use hop count as the link weight, and the resulting trees are group specific, i.e., one Steiner tree is specifically constructed for each multicast group.

3.4.1.3 Test Results

3.4.1.3.1 Functional Tests

The software is functioning correctly.

3.4.1.3.2 McastTE/Perf/OMTE-GA

Figure 29 illustrates the feature of overall bandwidth conservation capability of individual schemes with the variation of maximum group traffic demand D_g. As it is expected, explicit routing with the TM heuristic achieves the lowest overall network loading while random link weight assignment results in the poorest performance. We can also see in the figure that the GA approach exhibits the best capability in conserving bandwidth among all the hop-by-hop routing schemes. Typically, when the network is under-utilised, our proposed GA approach exhibits significantly higher performance than the conventional IP based solutions without explicit routing. For example when $D_g = 3000$, the overall bandwidth consumption of the Random and SPH solutions are higher than that of GA by 19.3% and 14.9% respectively. Compared with the TM heuristic that needs support from MPLS overlaying, the gap from GA is below 8%. However, when the external traffic demand grows, the performance of GA converges to that of the SPH approach. On the other hand, although the TM algorithm exhibits significant higher capability in bandwidth conservation when the external traffic demand grows ($D_g > 4000$), this does not mean what have been obtained are feasible solutions without introducing overloaded links.

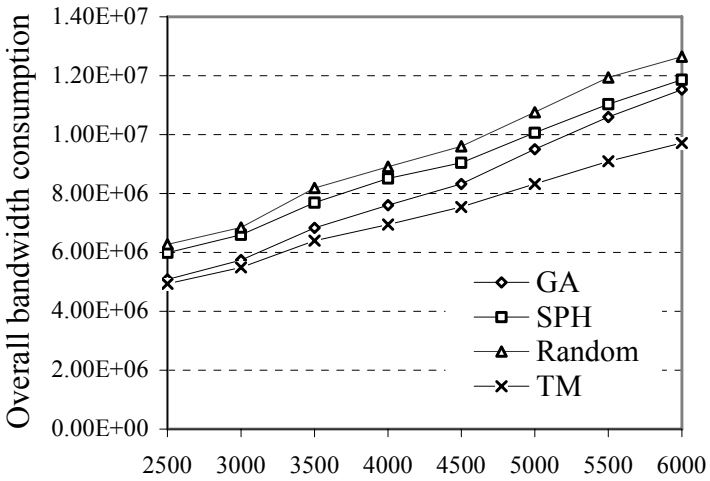


Figure 29: Total bandwidth consumption vs. Max D_g

Figure 30 shows the relationship between the proportion of overloaded links and the maximum group traffic demand D_g in time of network congestions. From the figure we can see that there are more overloaded links as D_g increases. The most interesting result is that, through our GA optimisation, the percentage of overloaded links is significantly lower than all the other routing schemes. In the most congested situation (D_g =6000), the average rate of overloaded links computed by GA is only 1.4%, in contrast to 12.6% by random link weight setting, 8.6% by the TM heuristic, and 4.4% by SPH respectively. On the other hand, the amount of overloaded bandwidth occurred on the most congested links is another important parameter an INP is interested in. We define the Maximum Link Overload Rate (MLOR) as follows:

$$MLOR = \max_{(i,j)\in E}\left(\frac{\sum_{g=1}^{G} D_g \times y_{ij}^{g} - C_{ij}}{C_{ij}}\right)$$

From this definition we can see that MLOR reflects the overloading scale of the most congested link (if any, i.e., MLOR>0). An INP should avoid configuring the network resulting in hot spots with high MLOR. Through our simulations, we also find that the proposed GA approach achieves the lowest MLOR performance. In Figure 31, the overloading scale is 45% of the bandwidth capacity on the most congested link in the GA approach with D_g equal to 6000, while this value reaches 110% and 59% in random link weight setting and SPH respectively. Even by using explicit routing TM heuristic, the overloaded bandwidth is 78% of the original link capacity.

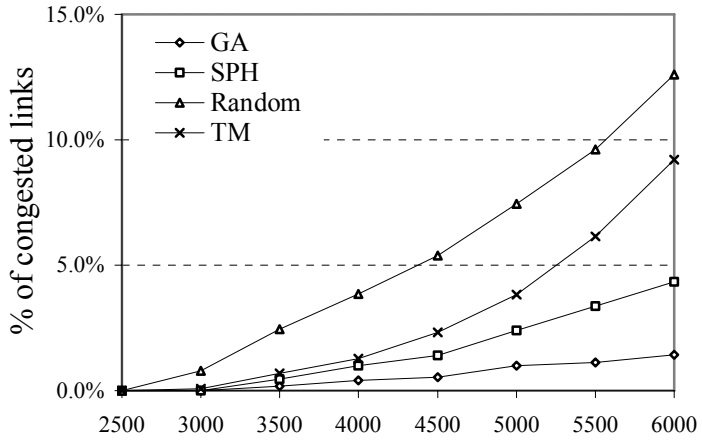


Figure 30: Overloaded link rate vs. Max D_g

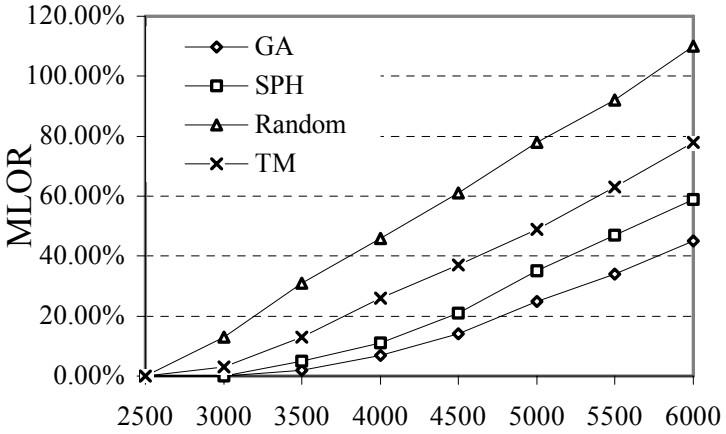


Figure 31: MLOR vs. Max D_g

From Figure 30 and Figure 31 we find that shortest path routing with hop-counts (SPH) has higher capability in finding feasible solutions (i.e., no overloaded links incurred) than random link weight setting approaches. Hence, we will start from the comparison between GA and SPH in the capability of exploring feasible solutions. Figure 32 presents the ratio of successful instances obtained by GA but failed to be found in SPH. In the figure, when the value of MLOR computed by SPH is in range of (0%, 5%], GA can obtain feasible solutions (i.e. $MLOR_{GA} \leq 0$) for 65% of these instances. We can also see that, with the increase of external bandwidth demands, the capability of GA in finding feasible solutions is decreasing. When the MLOR value of SPH grows up to 25% due to the higher external traffic demand, the success rate of GA drops to 5%. From this figure, it can be inferred that, when the external group traffic demand is at the brink of causing network congestion, GA has higher capability of avoiding link overloading compared to other approaches. Obviously, it may be the case that no feasible solution exists at all, if external traffic demand exceeds a certain threshold.

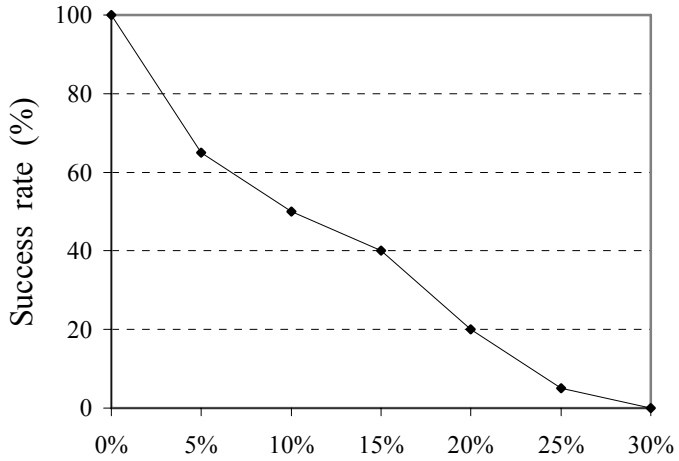


Figure 32: GA Success rate vs. $MLOR_{SPH}$

3.4.1.3.3 McastTE/Scal/OMTE-GA

The scalability test aims at the computing time required by the proposed GA based solution, particularly when large sized network topology and a large number of subscribed groups are considered.

The related GA configuration parameters for the following test are:

(1) $Population\_size = 100$

(2) $Maximum\_generation = 300$

Topology size	10	50	100	200
Running time (s)	36	196	400	828

Table 6: Running time vs. topology size (100 groups)

Number of groups	50	100	150	200
Running time (s)	201	400	602	801

Table 7: Running time vs. number of groups (100 nodes)

3.4.1.3.4 Inter-domain McastTE

The following are some preliminary test results for inter-domain multicast traffic engineering. We evaluate the three algorithms that are specified in [D1.3], namely (1) greedy single ingress router selection (GSIRS), (2) hop-count based Hot Potato routing (HC-HPR) and (3) GA based Hot Potato Routing (GA-HPR). We evaluate both intra-domain bandwidth consumption performances and inter-domain load balancing performances.

The configuration of Inter-domain Multicast TE is described as follows: The total number of multicast groups is set to 50 with altogether 20 DRs. We consider 100 sources and each of them can be reached via half of the border routers on average. The rest configuration is the same as the intra-domain scenario.

Figure 33 and Figure 34 shows respectively the overall intra-domain bandwidth consumption and the highest inter-domain link utilisation with the variation of D_g. From the perspective of bandwidth conservation, we can find that the two hot potato based routing approaches achieve significantly stronger capability than the single ingress router selection algorithm (GSIRS). However, from Figure

34 we can find that GSIRS has the best performance in terms of inter-domain link utilization among all the three solutions. On the other hand, the proposed GA based hot potato routing has resulted in higher link utilization by up to 10% compared to GSIRS, but it exhibits the best performance in intra-domain bandwidth consumption. Typically it only consumes 67% of bandwidth resources of GSIRS. From this point of view, we can regard the proposed GA based approach as a good trade-off example between intra- and inter-domain scenarios.

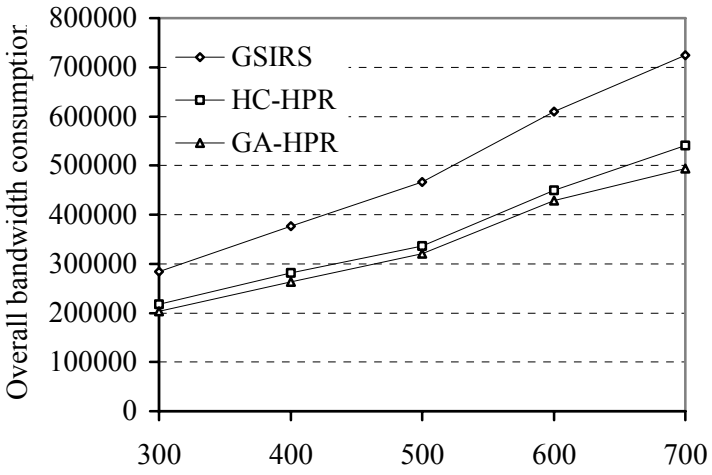


Figure 33: Total bandwidth consumption vs. Max D_g

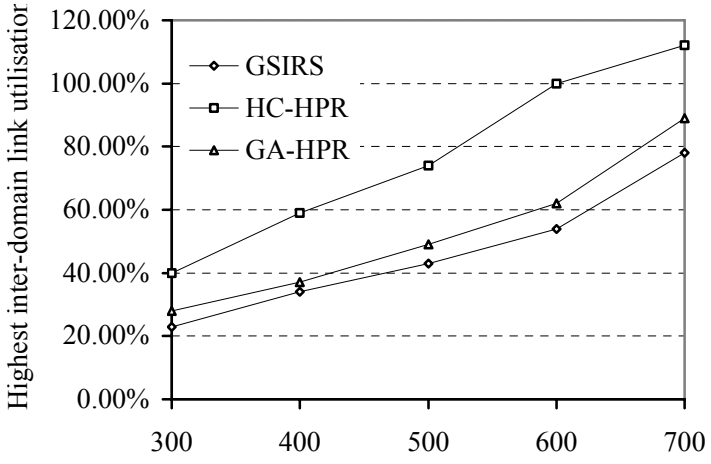


Figure 34: Highest inter-domain link utilization vs. D_g

3.4.1.4 Conclusions

From the above simulation results, we can find out that, compared to existing solutions, the proposed GA based OMTE algorithm can conserve significantly network bandwidth and is also able to guarantee higher success rate in finding feasible solutions with the bandwidth constraints. The running time for large sized networks and group numbers is within several minutes, which is feasible for offline TE computations.

3.4.2 Real-Time Test

3.4.2.1 Overview

In the real-time simulation tests, we mainly study the metric of blocking rate of group join requests based on the originally established mSLSes. The first objective (McastTE/Perf/DMR/1) is to study whether the proposed GA based OMTE algorithm is able to increase the service capability when individual mSLSes have been invoked. In addition, we also investigate the scenario when the proposed solution is applied to the DiffServ environment (McastTE/Perf/DMR/2), and see if effective service differentiation can be achieved without incurring any fairness issue between different l-QCs.

3.4.2.2 Experiment Setup and Test Description

Two different simulation scenarios have been applied in the real-time testing campaign. The first scenario (McastTE/Perf/DMR/1) is based on the flow level, which can be regarded as the continuation of the McastTE/Perf/OMTE-GA tests. The second (McastTE/Perf/DMR/2) is based on the packet level where the simulation is run on top of ns-2.

In the first scenario, we apply the same simulation model (topology, group membership information) as the one that is used in McastTE/Perf/OMTE-GA. Apart from that, We emulate a sequence of events for group membership updates based on the static scenario, and we evaluate the real-time traffic condition with the group dynamics derived from the original static multicast traffic matrix. For each event, we first randomly select one group $g \in G$, and then use the following probability function to decide whether this event is a group join or leave:

$$P_g = \frac{\omega(|V_g| - m_g)}{\omega(|V_g| - m_g) + (1-\omega)m_g}$$

In the function, m_g indicates the instant number of active members while $|V_g|$ identifies the maximum size of group g (i.e. total number of subscribers). ω ranging [0, 1] is known as the invocation ratio that controls the density of each group. For example, ω=0 means that no group joins are invoked, while ω =1 indicates full group membership invocation. In our simulation we use this function for creating a series of events of group join/leave based on the static multicast traffic matrix. When a join request is issued for group g ($P_g >$ a randomly created float number ranging from 0.0 to 1.0), a node $v \in V_g$ but not yet on the multicast tree T_g is selected to join the group. Likewise, in case of a leave request for group g, an on-tree node is randomly selected for pruning from T_g. By introducing this group dynamics generator, we can also investigate the stability of the proposed solutions in time of inaccurate mSLS invocations (i.e., not all receivers activate their contracts by sending group join requests).

In the ns-2 based simulation test, our configuration is as follows. The network shown in Figure 35 comprises two ingress routers (S1, S2), three egress routers (R1, R2, R3) and two core routers (C1, C2). The bandwidth capacity of each link is 10Mbps. The metric of each link is set to 1 so that the join request always follows the path with the minimum number of hops back to the source. We assume that the INP is providing 4 l-QCs, i.e. l-QC1, l-QC2, l-QC3 and l-QC4. The scheduling mechanism for individual l-QC queues is based on Weighted Round Robin (WRR), and the weight for each l-QC queue is set as follows:

$$\begin{bmatrix} l-QC1 \\ l-QC2 \\ l-QC3 \\ l-QC4 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

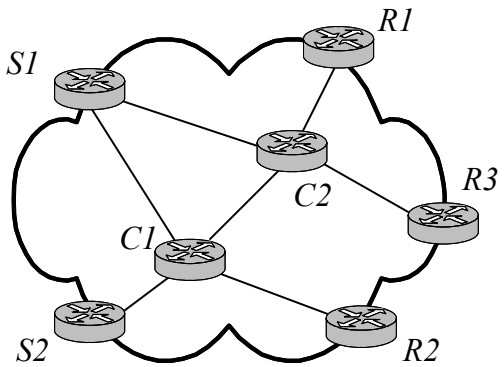


Figure 35: ns-2 based simulation topology

3.4.2.3 Test Results

3.4.2.3.1 Functional Tests

The software is functioning correctly.

3.4.2.3.2 McastTE/Perf/DMR/1

In the following simulation tests, we assume that new group join requests will be blocked once network congestion (i.e., an overloaded link) has been detected. Figure 36 and Figure 37 show respectively one typical instance of the real-time performance (5000 events in group dynamics) in terms of overall network load and maximum link utilisation respectively, with D_g equal to 3000 and ω equal to 1.0. In this condition the network is lightly loaded with no link congestions (over-provisioning). From Figure 36 we can see that when the group dynamics converge to a steady state, the network load resulting from random link weight setting is the highest, while using the TM algorithm for MPLS explicit routing achieves the lowest resource consumption. We also find that the proposed link weight optimisation using the GA approach results in very low network load compared to other IP based approaches, and its performance is even very close to the TM explicit routing scheme. This result is consistent with the static simulation scenario shown in Figure 29. As shown in Figure 37, the GA optimisation approach results in very high utilisation of the most heavily loaded link, which is only next to the Random link weight solution. On the other hand, both the SPH and TM algorithms exhibit good performance in load balancing. Nevertheless, it should be noted that although the performance in maximum link utilisation by the GA approach is not as good as these two schemes, there is still no network congestion as all the links are under-utilised and the overall bandwidth resources are significantly conserved.

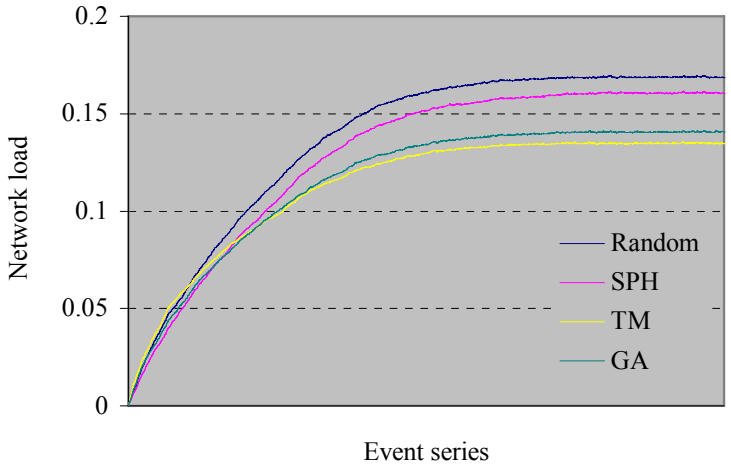


Figure 36: Real-time performance in average network load (Max D_g =3000, ω=1)

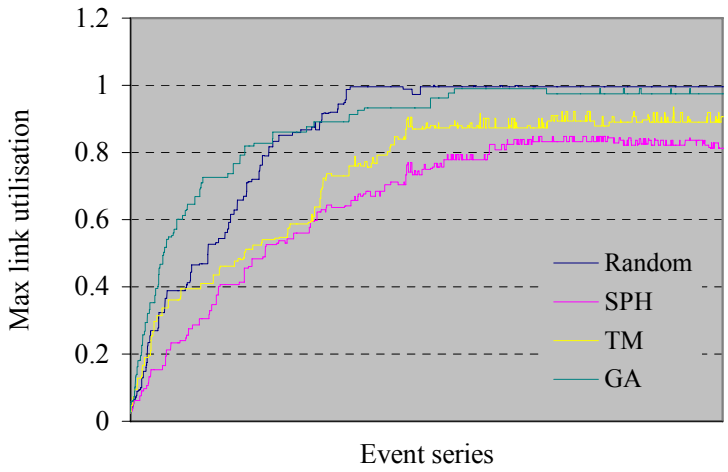


Figure 37: Real-time performance in maximum link utilisation (Max D_g =3000, ω=1)

From Figure 38 and Figure 39 (typical instances for over-subscription scenarios) we can see that the performance of the four approaches changes significantly in time of overwhelming traffic demand when Max D_g is increased to 6000. First, both the GA and SPH approaches converge to the highest overall network load. On the other hand, explicit routing with the TM algorithm still achieves the lowest resource consumption, which remains the same with the scenario in Figure 38. From Figure 39 we see that all four schemes result in 100% utilisation in the highest loaded link due to the overwhelming traffic demand, and thus some new group joins are blocked due to the overloaded links. We can also see from this figure that the random approach first converges to the congested state while our proposed GA optimisation is the last to reach this phase. This implies that more group joins are likely to be rejected in the former while the least join requests will be blocked in the latter. In effect, group join blocks prevent the underlying multicast trees from consuming more network resources, and this explicitly explains why the overall network load of SPH and GA is higher than the random link weight approach in Figure 38, where a large number of group joins have failed due to overloaded links. Our subsequent simulation study will continue to focus on the statistics of group join blocks for the four approaches in different scenarios (e.g., with variations of ω).

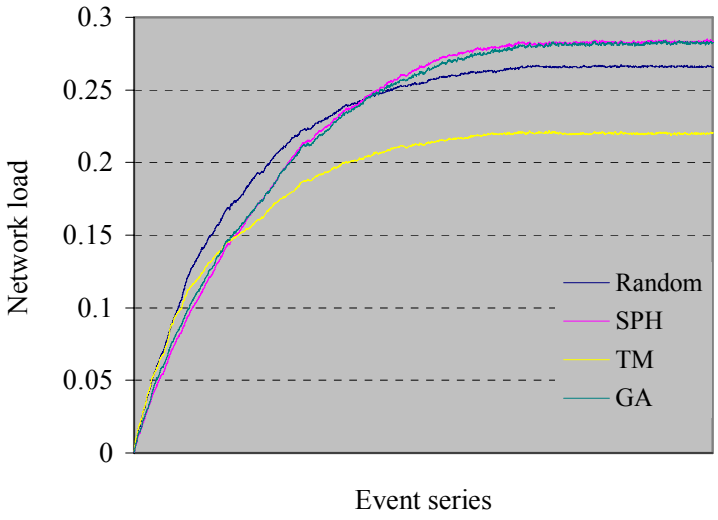


Figure 38: Real-time performance in average network load (Max D_g =6000, ω=1)

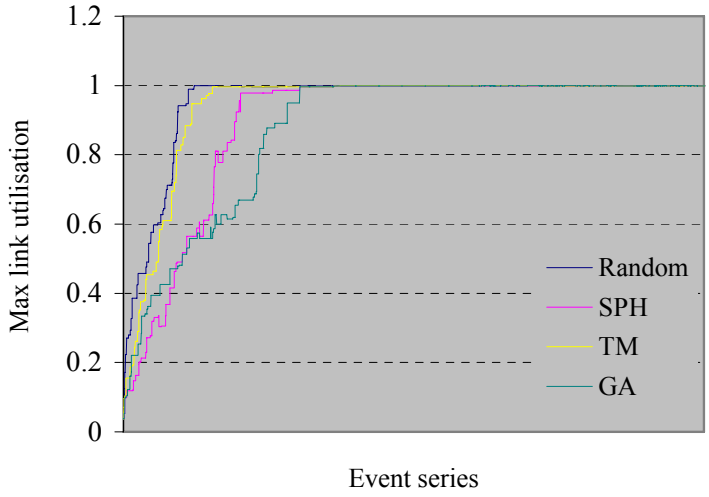


Figure 39: Real-time performance in maximum link utilisation (Max D_g =6000, ω=1)

3.4.2.3.3 McastTE/Stab/OMTE-GA/1

In this test campaign, we investigate the stability of the proposed algorithm in terms of inaccurate mSLS invocations. Figure 40 illustrates the overall block rate with the variation of the invocation ratio ω with respect to the 5000 group updates, while maximum D_g is set to 6000. From the figure we can see that more group joins are rejected as the invocation ratio grows. The reason for this is that, bandwidth consumption increases when there are more active members in each group. Once the consumed bandwidth on any link reaches its capacity, new group joins are blocked due to the detected congestion. On the other hand, we notice that through sophisticated network dimensioning using the proposed MT-IGP link weight optimisation, group join blocks are significantly lower than in the other approaches. When ω increases from 0.5 to 1.0, the total number of blocks grows very slowly with our proposed GA solution, which is in contrast to all the other conventional methods. One interesting thing is that, compared to Figure 31 in the static scenario, although the provisioning performance of the GA approach results in 45% MLOR, the actual number of blocked join requests is quite low (2.1%) even in case of full group invocation. When ω < 0.7, there are no blocked group join requests at all. The reason for this is that while there are overwhelming group joins, group leaves also take place at the

same time, with used bandwidth resources returned to the network. Finally, it is also worth mentioning that the MPLS based Steiner tree approach does not exhibit strong capability in reducing the blocking rate, as the TM algorithm is solely greedy in bandwidth conservation and not in eliminating congested links.

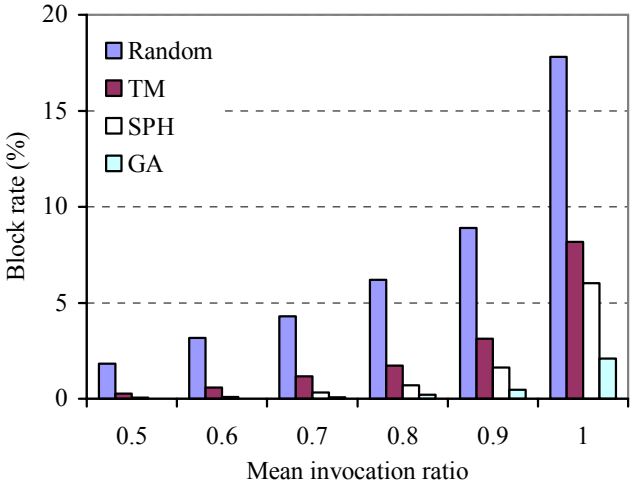


Figure 40: Join block rate vs. invocation ratio ω

Figure 41 shows the overall network load versus invocation ratio ω with respect to the 5000 group updates. From the figure we can see that higher invocation ratio results in higher network load. On the other hand, the TM heuristic using MPLS explicit routing always achieves the lowest network load, which is in line with Figure 36. Moreover, we also notice that the network load of the GA optimisation is very close to that of the TM approach when ω is relatively small, and this again indicates that the proposed solution exhibits strong capability in bandwidth conservation in time of light traffic loading. However, with the growth of ω, the network load by the GA approach increases more sharply than all the other approaches, and this is because more group joins are able to be accommodated successfully, while in the other approaches, especially the random link weight one, a large number of join requests are blocked due to network congestion so that the total bandwidth consumption is relatively lower.

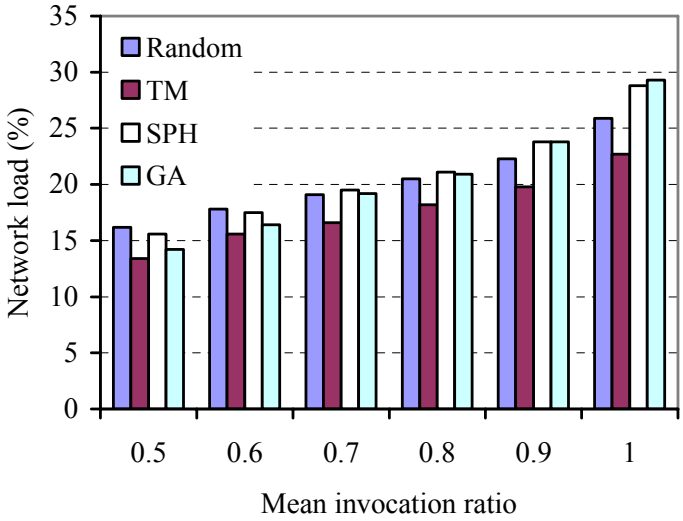


Figure 41: Network load vs. invocation ratio ω

3.4.2.3.4 McastTE/Perf/DMR/2

In our first experiment in this test campaign, there are two active groups whose sources send data via the ingress routers S1 and S2 respectively. For simplicity we only consider two classes of service in this experiment, i.e. l-QC1 and l-QC2. The source rate from S1 is 2Mbps and that from S2 is 1Mbps (both for l-QC1 and l-QC2). We also set 3Mbps background traffic (both l-QC1 and l-QC2) from each ingress router to all the egress routers. We consider the situation that each egress router joins both channels with l-QC1 and l-QC2 simultaneously, resulting in 4 distinct multicast trees: (S1, l-QC1), (S1, l-QC2), (S2, l-QC1) and (S2, l-QC2). We define the Transmission Ratio (TR) as the number of packets received by each group member over the total number of packets sent by the source. Figure 42 illustrates the TR performance of each source/receiver pair. We can see that in most cases the TR performance of l-QC1 is significantly better than that of l-QC2 (except R2). By examining the traffic load of each link, we find that all the links between S1 and R2 (i.e., S1→ C1 and C1→R2) are under-loaded, resulting in 100% transmission ratio for both l-QC1 and l-QC2. On the other hand, the performance of transmission ratio also depends on the location of the egress router through which group members are attached to the distribution tree. For example, both egress routers R1 and R3 have l-QC2 group members for S1. Our simulation results show that the TR value for R1 is 59.4% while that for R2 is significantly higher (77.6%). This is caused by the more overloaded link C2→R1.

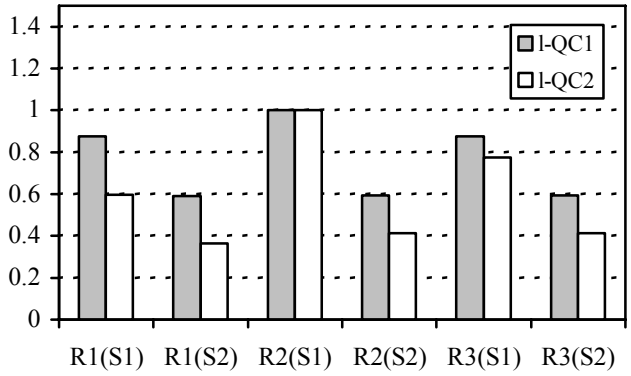


Figure 42: Transmission ratio of 2 groups

Next we investigate the performance of individual group members attached to the same egress router. The objective of this experiment is to examine the inter-class fairness in more detail without considering the receivers' physical location. The simulation scenario is described as follows: The source rate of S2 is fixed at 1Mbps and 4 receivers attached to egress router R1 join the session by subscribing to 4 different channels, i.e. (S2, l-QC1), (S2, l-QC2), (S2, l-QC3) and (S2, l-QC4). The grey column of Figure 43 indicates that if none of the links on the tree branch S2→C1→C2→R1 are congested, the transmission rate of all the four classes is 100%. In order to evaluate the performance in time of congestion, we impose 3Mbps background traffic for each of the four l-QCs. From the figure we can observe the significant differentiation of the four l-QCs when the network cannot handle all the traffic. The group member subscribing to the l-QC1 channel achieves virtually no packet loss, whereas the one subscribing to l-QC4 channel only receives 24.6% of the packets from S2. The "Good Neighbour Effect" does not happen if we construct this type of QoS specific trees for each channel. This scenario demonstrates the benefit of building per QC trees for end users with different QoS requirements.

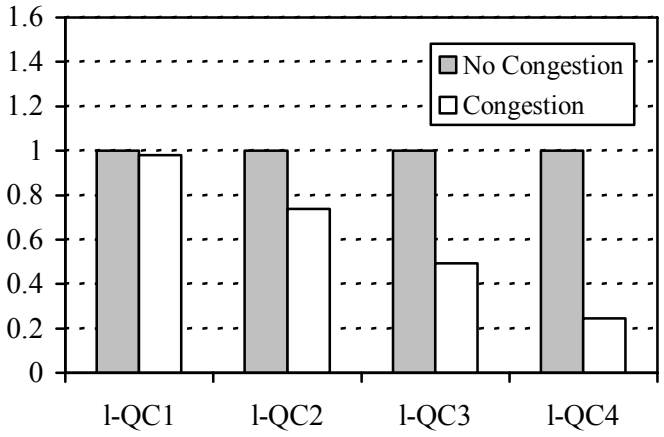


Figure 43: Simultaneous l-QC joins

Figure 44 illustrates the scenario when a particular group member dynamically upgrades its service level by joining higher QoS channels. Let's assume that a receiver attached to R1 subscribes to the (S2, l-QC4) channel in time t_0. Due to its capability to upgrade, this group member upgrades to the next higher QoS channel a number of times, i.e., at t_1 to l-QC3, t_2 to l-QC2 and t_3 to l-QC1. From the figure we can see that this upgrading makes no improvement when there is no congestion along the tree from S2 to R1; in fact, the group member can achieve 100% packet transmission ratio at time t_0 by subscribing to (S2, l-QC4). On the other hand, when we impose additional background traffic (in the same fashion to the last experiment), the performance differentiation of individual channels becomes obvious: the transmission ratio at t_1 goes up to 58.0%, to 73.3% at t_2, and finally achieves 85.1% at t_3 when the user finally subscribes to the (S2, l-QC1) channel.

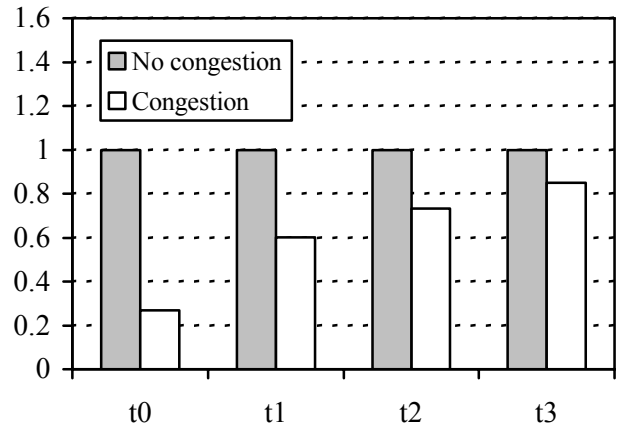


Figure 44: Dynamic l-QC upgrading

3.4.2.4 Conclusions

From the real-time performance tests we found that both bandwidth conservation capability and service capacity achieved by the proposed GA approach is constantly higher compared with existing paradigms even when the mSLSes are implicitly invoked. These results have proved the high stability of the proposed algorithm. Moreover, the service capacity in terms of admitting group join requests has also been drastically increased compared with conventional approaches. Finally, we also indicated in our simulation that inter-l-QC fairness problems are avoided by applying per l-QC trees.

4 DYNAMIC TRAFFIC ENGINEERING TESTS AND RESULTS

4.1 q-BGP Simulation Tests

This section focuses on the results of a macro-scale simulation of inter-AS topologies aimed at evaluating q-BGP's large-scale behaviour – tests that would be unfeasible on the relatively small-scale MESCAL testbed. The simulation models aggregate flows rather than individual packet behaviour as it implies an enormous simulation overhead when considering large inter-AS topologies and furthermore that level of simulation detail is unnecessary for the macroscopic behaviour under investigation.

It should be noted that these experiments are aimed at evaluating the use of q-BGP in the context of MESCAL's loose guarantees solution option (LGSO) and that since each meta-QoS-class runs a separate instance of q-BGP, will limit our simulations to a single QoS-class plane. This can be justified by the assumption that there is a partitioning between meta-QoS-classes at the pSLS level and therefore no interaction or interference between meta-QoS-class planes. The experiments and results obtained cover the following aspects:

- Scalability, which examines how the number of q-BGP messages depends on variables such as network size, topology, and traffic demand patterns.
- Stability, which considers 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.
- Efficacy, which considers 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).

4.1.1 Simulation Scenarios

When simulating an inter-domain network with inspection of QoS performance, a number of experimental variables play a large role in the resulting performance. These were discussed in Deliverable D3.1 and include:

- **Inter-domain Topology:** A representative inter-AS topology is required which we obtain from the BRITE topology generator. This creates power-law compliant topologies [bu02] when its preferential attachment option is used, and it has been shown that the Internet is also a power-law compliant topology at the AS level [fal99]. Parameters include network size (number of ASs) and average connectivity (the number of inter-domain links per AS).
- **Demand Matrix:** This is the traffic to be applied to our network. The demand matrix comprises a full mesh of demands between all AS pairs and whose offered bandwidth is uniformly randomly distributed across all demands. The parameter for the demand matrix generator is total network demand, so therefore the average demand bandwidth is this total bandwidth divided by ½.N.(N-1), where N is the network size in number of ASs. The absolute values chosen for the total demand bandwidth aren't important (as long as they are high enough to prevent rounding errors) as the pSLS capacities are directly derived from this matrix.
- **pSLS Capacity Matrix:** This is the scarce resource for which q-BGP is to optimise routing for. The pSLS generator described in Deliverable D3.1 provides a "base" pSLS capacity matrix which is capable of satisfying the input demand matrix but only with a single very specific routing configuration. So as not to favour shortest-path routing strategies the available capacity is placed on paths away from the shortest path. It would be very difficult for any routing policy to find the exact correct routing configuration that was used to generate the pSLS capacities so a pSLS scaling co-efficient is used to scale the capacities. Scaling the

capacities and therefore over-provisioning pSLSs, the number of alternative paths with available resources increases and it becomes easier for the routing policy to find a suitable routing configuration. This pSLS over-provisioning co-efficient is therefore a parameter to influence the number of suitable routing configurations. q-BGP strategies which require less over-provisioning to achieve good end-to-end QoS characteristics for demands are therefore the better solutions. As the pSLS capacities are in "useful" locations in the network an examination of pSLS utilisation is a meaningful measure of network resource utilisation. For the 100 AS experiments here the average shortest path between every AS pair was 2.8, however pSLS capacities were allocated on paths which had an average length of 4.6.

- **Aggregate flow treatment model:** as part of the calculation of end-to-end delay and delivered bandwidth we must emulate the effect of network congestion on packet flows through the network. To this end we use a simple M/M/1 queue to approximate queuing delay at pSLSs. Since router buffers are finite the delay experienced is capped at 100 ms. Demands have been implemented to perform as if they were inelastic and if along the path of the demand there is not enough capacity available then the demand will experience a degradation in throughput for the successive hops. The division of available pSLS capacity between demands is performed to the ratio of incoming offered flow bandwidth.
- **l-QC Generator:** For simplicity it is assumed that within each AS there are pre-defined l-QCs between all ASBRs (AS Border Routers). The ASs are assumed to have sufficient bandwidth for accommodating demands for the offered l-QCs (since the scarce resource which we are optimising for is the pSLS capacity and l-QCs should be matched anyway to pSLSs), and a fixed QoS transfer characteristic (i.e. a fixed delay). For these experiments the delay parameters are generated with a uniformly random number generator between the bounds of 5 and 50 ms and remains constant for all the l-QCs within a single AS, but differ between ASs.

4.1.2 q-BGP Policies under test

4.1.2.1 QoS_NLRI QoS Attributes

In these experiments we'll be concentrating on two QoS Attributes (QAs):

4.1.2.1.1 One-Way Delay (OWD) QA:

This is the expected time for a packet to reach the prefix advertised. When traversing pSLSs and ASs this value is formed through the concatenation of the various delay contributors:

```
Advertised OWD QA = incoming advertisement OWD QA + l-QC delay + pSLS queuing
delay;
```

When calculating the actual *delivered* end-to-end delay the value calculated from the aggregate flow treatment model is used instead.

4.1.2.1.2 Bandwidth (BW) QA:

This is a value for available bandwidth to the prefix specified in the NLRI field. l-QCs are assumed to have sufficient bandwidth so the only restriction is the pSLS capacities, thus the value advertised becomes:

```
Advertised BW QA = min (incoming advertisement BW QA, offered pSLS capacity);
```

As there is no injection of dynamically monitored QoS attributes in this set of experiments the "offered pSLS capacity" is specified as the capacity of the pSLS on which the q-BGP advertisement came in, divided by the number of neighbours to which this message is to be re-advertised.

4.1.2.2 Route Selection Policies

Throughout the experiments we examine a number of route selection policies which make use of various combinations of the 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. This approach is similar, but different to the precision parameter described in section 10.5.1.5.5.2.2.2.4 of deliverable D1.3.

In the following simulations QA equivalence is calculated by:

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

then the messages are equivalent and the decision must be performed on the next metric.

The route selection processes examined here are:

4.1.2.2.1 Meta-QoS-Class Identifier Only (MCID-only)

Routing decisions are based purely on AS Path length, and use ASN (AS number) as a tie-breaker.

4.1.2.2.2 Delay QoS Attribute only (DELAYQA-only)

The routing decision is performed based on a One Way Delay (OWD) QoS attribute first, and then on AS path length and ASN. The value of the OWD is static throughout the simulation and calculated as described in section 4.1.2.1.1. A range of equivalence margins, pOWD, for OWD are also examined.

4.1.2.2.3 Bandwidth QoS Attribute only (BWQA-only)

The routing decision is performed based on the bandwidth QoS attribute. The advertised BW QA is as described in section 4.1.2.1.2. i.e. if an AS has eight neighbours and one of these sends an incoming message advertising 49 bandwidth units to the prefix in the NLRI, and the pSLS capacity is 100 units then the AS will send seven messages, each advertising 7 bandwidth units to the prefix in the NLRI. A range of equivalence margins, pBW, for BW are also examined.

4.1.2.2.4 Delay and Bandwidth Priority scheme (DELAYBWPRIO)

A two level priority scheme where depending on the priorities specified in the policy either one of OWD QA or BW QA is checked first, and then if found equivalent (depending on the pBW and pOWD parameters) the other QA is checked. If that too is equivalent the decision is the based on AS path length and the ASN.

4.1.3 Experimental overview

We examine three aspects of q-BGP policies, as described in the introduction to section 4.1: Scalability, Stability and Efficacy. To limit the range of parameters for the various policies under test we must first find regions of parameter space which perform well in efficacy tests to continue are analysis, otherwise the graphs become cluttered and have little additional value. Any additional parameters are described as part of the experiment groups below. As mentioned before all experiments are for a single meta-QoS-class in MESCAL LGSO (loose guarantee solution option).

4.1.4 Experimental results: efficacy

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 pSLS utilisation are the standard deviation of the mean pSLS utilisation for each network and not the mean of the standard deviations for all pSLS within each network.

Our first inspection of efficacy is to examine the effect of QA equivalence margins and to find a range of useful values for pBW and pOWD.

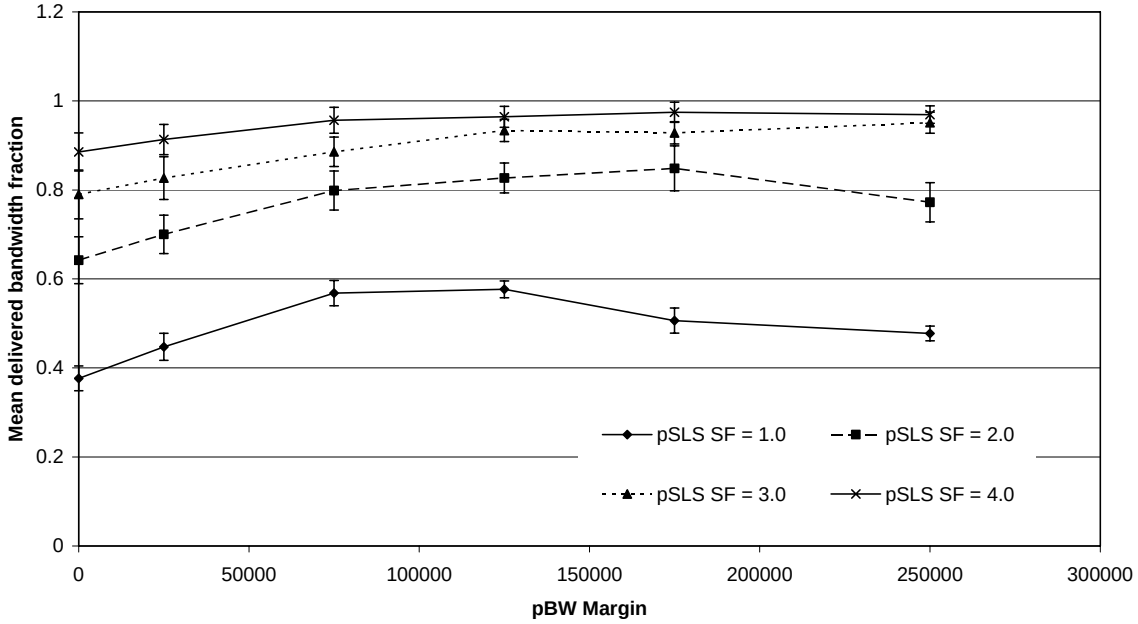


Figure 45 Mean delivered bandwidth fraction (delivered/offered) for a range of pBW under the BWQA-only policy

In Figure 45 we can see that when resources are scarce (pSLS SF = 1.0) the delivered bandwidth is low for no margin (pBW = 1.0) 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. For reference the delivered BW fraction (and for later reference the mean delivered end-to-end delay) for MCID-only is:

pSLS SF	Mean Delivered BW fraction	Mean delivered end-to-end delay (ms)
1	0.4713	201.7
1.5	0.5446	193.6
1.75	0.5733	190.2
2	0.5986	189.7
2.25	0.6216	187.9
2.5	0.6422	183.6
3	0.6777	177.0
4	0.7347	171.4

Table 8 Mean delivered BW fraction and delivered end-to-end delay for the MCID-only

We hypothesise that the cause of the poor initial performance of BWQA-only with pBW = 1 is the convergence of routing paths towards the areas of high capacity and therefore the saturation of those links. As the QA values are static and administratively set they won't change to reflect this saturation

and the overall throughput for demands suffers. We refer to this phenomenon as the "QA rush". The effect can be also seen in the average utilisation of pSLSs in Figure 46:

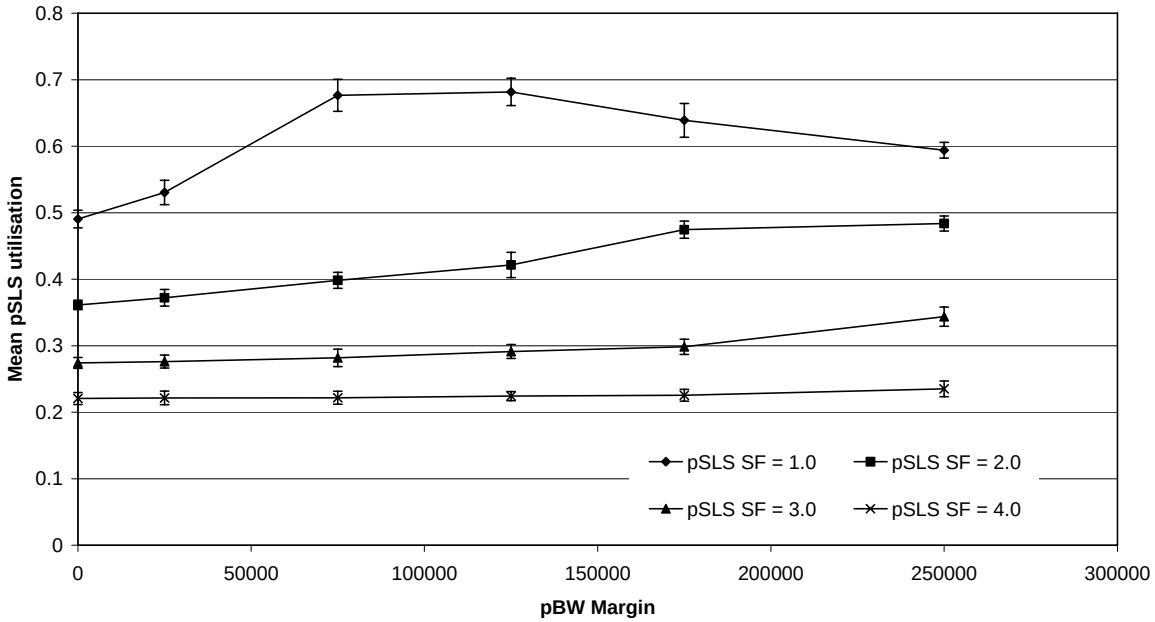


Figure 46 Mean pSLS utilisation for a range of pBW equivalence margins for the BWQA-only policy

As the pBW margin increases route selection is no longer performed purely on the BW QA, but also on AS path length, then there is less of a rush towards the high capacity links. This can be seen here as an increase in the average pSLS utilisation as more of the demand gets through the bottlenecks and the network load is better distributed across the network.

A second performance metric is mean delivered end-to-end delay. In our simulations the per-domain delay is chosen at random from the range 5 to 50 ms, while the administratively set pSLS queuing delay contribution is considered to be 7 ms. The figure of 7 ms is really a forecast of the very worst case scenario and corresponds to a pSLS utilisation of 0.875, assuming the queue behaves as an M/M/1 process. The sum of these values is the OWD QA metric advertised in the qBGP messages. However, when calculating the delivered end-to-end delay the per-domain delay is used alongside the *value calculated by the M/M/1 queue equation* (D3.1) (which is capped to 100 ms). This will provide a feedback into delivered delay from highly utilised pSLSs. This is so as to model the benefit to delivered delay from BW QA based policies.

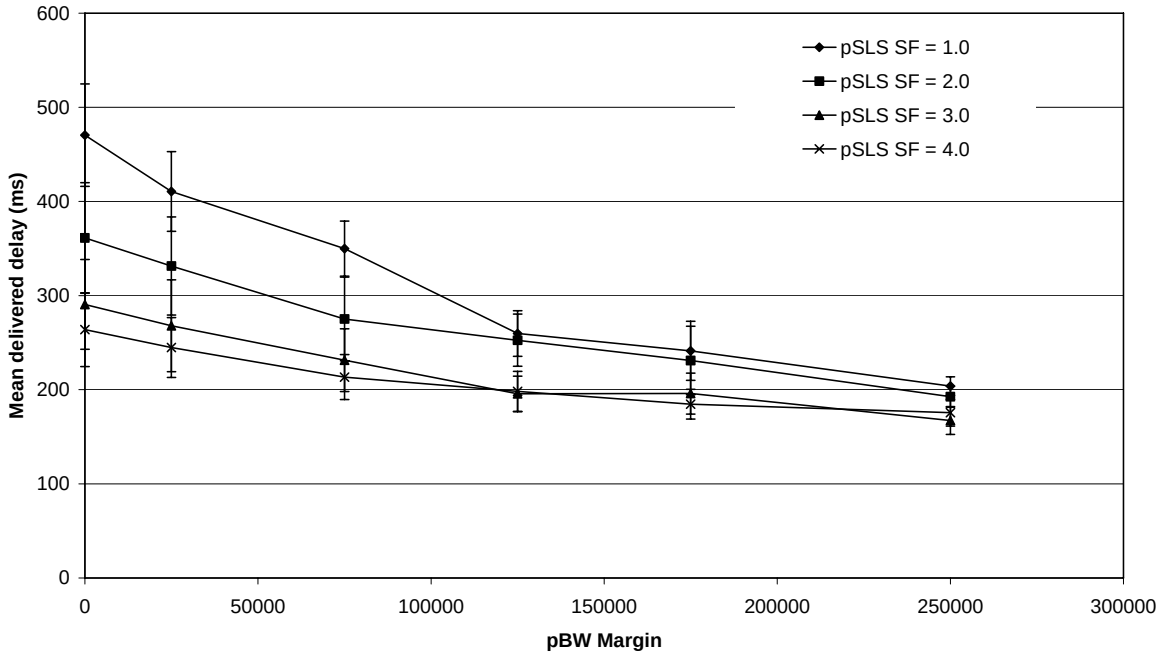


Figure 47 Mean delivered delay for various BW QA equivalence margins for the BWQA-only policy

In Figure 47 we see that for all over-provisioning factors pBW margin values below 125000 results in a much worse delivered delay, and higher values yield a much less significant improvement. This would suggest that a pBW value of 125000 is beneficial to end-to-end delivered delay. This improvement in delivered delay is the result of better load distribution, more available capacity, and therefore faster queuing service rates and a lower queuing delay.

The graph in Figure 48 examines the effect of the OWD equivalence margin on the actual delivered delay. When inter-domain resources are scarce (pSLS SF = 1.0) and there is no equivalence, all decisions are made purely on OWD QA and any advertisements of low delay routes cause "QA rush" causing poorer delivered performance. As the equivalence margin increases a number of alternative routes appear across which load is distributed, leading to less congestion and better delivered delay. As the margin increases further the delay gets worse (more significantly for the high pSLS SF cases) and the routes approach the shortest-path and the resulting delivered delay becomes more like the MCID only case (see Table 8 for comparison). A general purpose value for pOWD margin is therefore 50 ms as this is approximately a point where performance changed for the entire range of pSLS SFs.

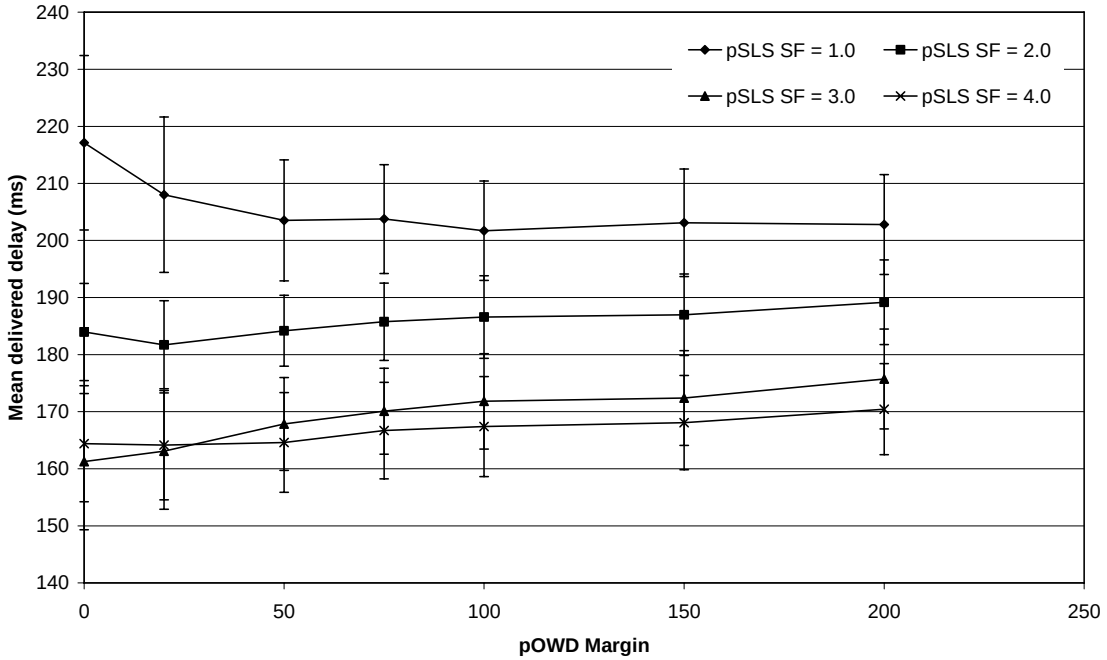


Figure 48 Mean delivered delay for various OWD QA equivalence values for the DELAYQA-only policy

In Figure 49 we can see the fraction of the offered load that is actually delivered for a range of route selection policies. This time we compare a range of policies against the amount of over-provisioning. The pBW chosen (125000) for the BWQA-only policy is seen as one of the better values from the previous graph, similarly a pOWD (the OWD equivalence margin) of 50 is found to perform better than other values, e.g. Figure 48.

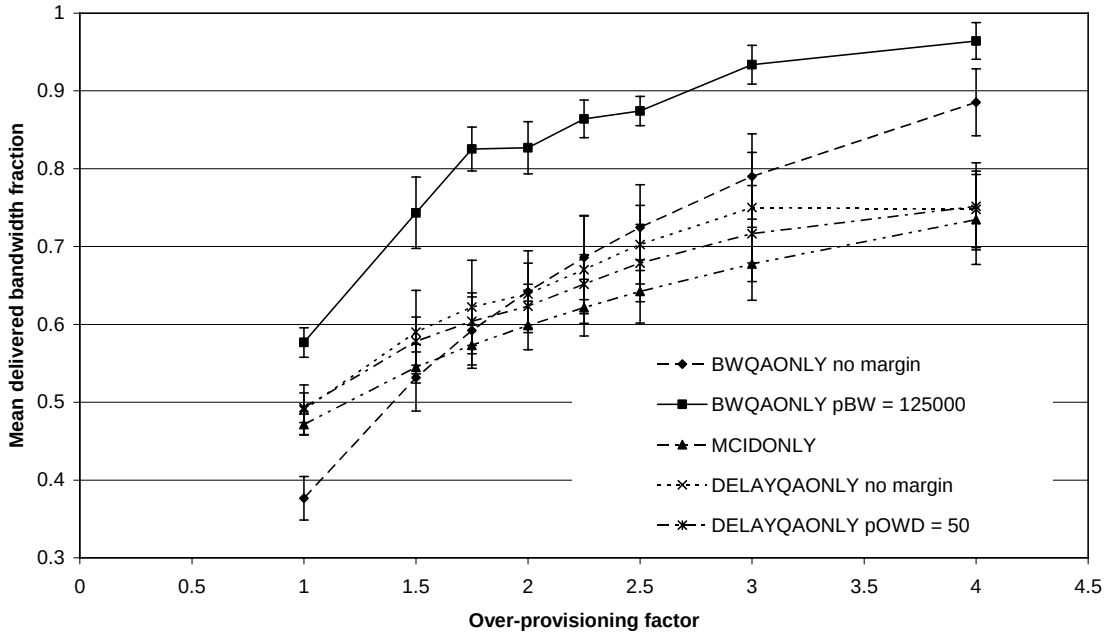


Figure 49 The mean delivered bandwidth fraction over a range of over-provisioning coefficients for the various q-BGP policies

For the delivered bandwidth fraction performance metric MCID initially outperforms BWQA-only with no equivalence margin (pBW = 1), but with an increase in over-provisioning the BW QA based policies outperform all other policies. MCID is outperformed by all policies, including the policies based on OWD QA which follow a non-shortest path dictated by the per-domain delays, adding heterogeneity and a certain level of load balancing.

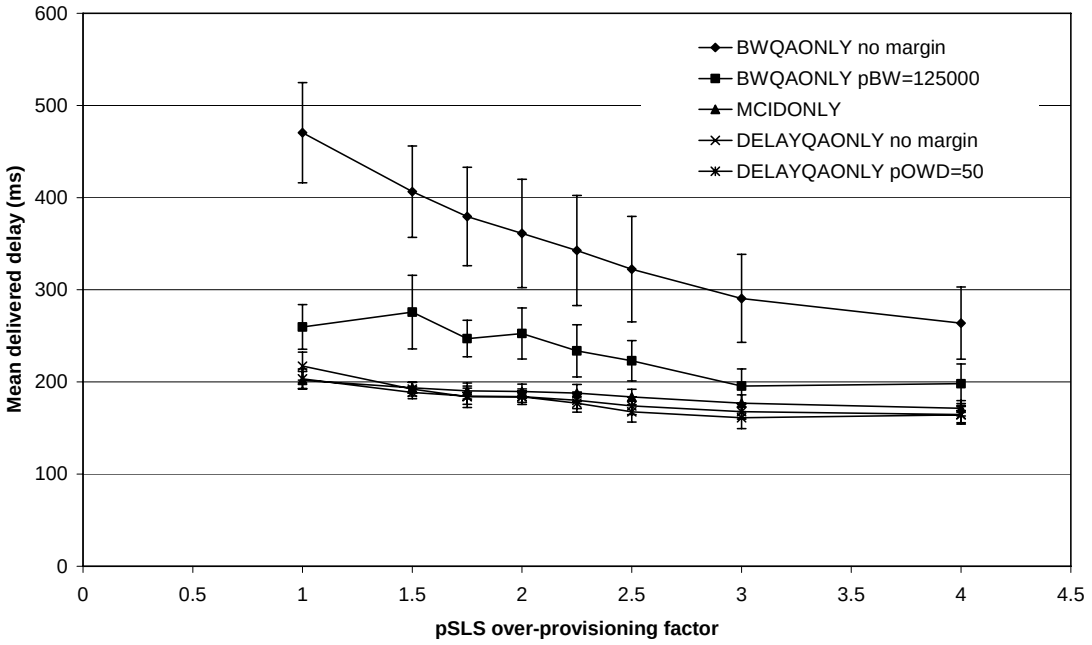


Figure 50 Mean delivered delay for a select range of policies against the over-provisioning co-efficient

In Figure 50 we can see the mean delivered delay as experienced by all demands. The plot demonstrates that BWQA-only with no equivalence margin results in very poor delays in the network. This is caused again by the "QA rush", and as over provisioning is increased the delays drop as the bottlenecks are lessened. With an equivalence margin the BWQA-only actually performs a bit better because of the load distribution afforded by the shortest path decisions. What is interesting also from this graph is that DELAYQA-only doesn't perform significantly better than MCID-only. This is probably caused by the random distribution of l-QCs and because of the homogeneity of pSLS administratively set delay contributions (the 7 ms) effectively denature the path into a shortest-path equivalent. Any benefit from choosing a lower delay path by DELAYQA-only may also be eroded by a "QA rush" effect on that low delay path and forcing queuing delays higher until the result isn't much better than shortest-path.

4.1.5 Experimental results: comparison of q-BGP selection policies

This section analyses the relative performance of a range of different q-BGP selection policies in terms of their impact on the delivered delay and delivered bandwidth of end-to-end flows. The q-BGP route selection policies subject to test and comparison are:

- MCID-only. Given that the simulations focus on a single meta-QoS-class plane these tests are effectively without any additional QoS information injected into q-BGP and are therefore equivalent to classical BGP.
- Single QoS attributes of DELAYQA-only and BWQA-only. The performance of the q-BGP selection process where there is no margin of equivalence is compared to the delivered QoS when a margin is used. The values selected for the equivalence margin under test are those that exhibited good performance in the tests described in the previous subsection.
- Priority-based selection on both DELAYQA and BWQA attributes comparing the performance of DELAYQA having priority over BWQA and vice versa. A range of equivalence margin values were used for these tests based on the results obtained for single QoS attribute tests as described in the previous subsection.

Figure 51 shows a scatter plot of mean delivered delay against mean delivered bandwidth for a range of q-BGP selection policies. The results are shown for three pSLS over provisioning factors – 1.5, 2.0 and 2.5 – shown from left to right on each of the curves. Although results were obtained for many cases of equivalence margin value, for clarity the comparison graph concentrates on results from a more limited set of values, selected to highlight the major differences between the selection policies.

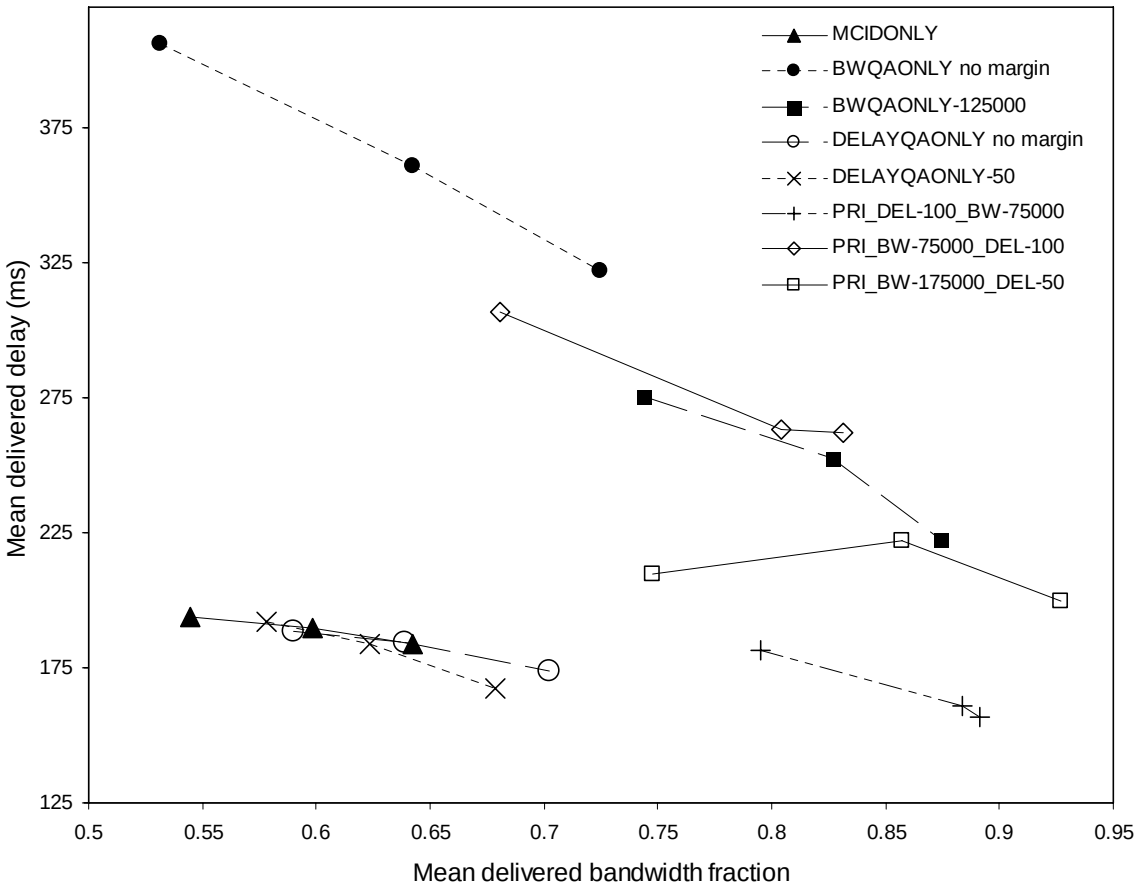


Figure 51 Effect of q-BGP selection policy on delivered delay and bandwidth

The policy of selection based on BWQA-only with no equivalence margin delivers higher bandwidth fractions than MCID-only for higher pSLS over provisioning factors, but performs worse than MCID-only in congested networks. The reason for the latter is due to the phenomenon of QA-rush as described earlier in the section. In all cases the adoption of the BWQA-only policy shows worse delivered delay than MCID-only, due to it selecting the largest capacity route at any cost. By adding a margin of equivalence, e.g. of 125000 bandwidth units as shown for the BWQAONLY-125000 curve, the performance is improved in terms of delivered delay and bandwidth when compared to selection based on the absolute widest path. This also beats MCID-only in terms of delivered bandwidth fraction but not one way delay. The policy of using an equivalence margin improves performance because the QA-rush has been avoided 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.

The policy of selection based on DELAYQA-only shows some improvement over selection based on shortest AS path (MCID-only) in terms of both delay and delivered bandwidth. However the improvement is marginal. 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 is

selected carefully so that the ASs along shortest path routes have large l-QC delays then a more marked improvement in performance of the DELAYQA-only selection policy may be observed.

The best performing route selection policies are those that select paths according to advertised delay *and* bandwidth. PRI_ DEL-100_ BW-75000 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 75000 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 pSLS over provisioning factors. It is interesting to compare this to PRI_BW-75000_DEL-100 – i.e. the same bandwidth and delay margins, but with the priority reversed (first select based on bandwidth then on delay). In the latter case delivered bandwidth and delay is worse than the former and worse than selection based on BWQA-only with a wider margin of equivalence. However 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-75000_DEL-100 with PRI_BW-175000_DEL-50. It appears, therefore, that it is better for the path selection process not 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.

One notable conclusion to be drawn from these comparisons is that a range of different performances can be achieved through applying different q-BGP selection policies. This means that different meta-QoS-classes may require different selection policies to implement desired end-to-end behaviour. 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 it also indicates that end-to-end QoS differentiation is achievable even with homogenous forwarding behaviour for all traffic classes, e.g. BE only as in the current Internet.

4.1.6 Experimental results: scalability

The improvement in delay can be seen in Figure 52 as a function of AS topology size. It can be seen that the benefit of additional QoS info (delay QA only with no margin in the shown tests) in q-BGP messages is increasing with topology size. This is due to an increased number of alternative AS paths between a given source-destination pair (other than the default shortest AS-path length) as the topology grows, and therefore the chances of finding an improved path on one-way delay grounds is increased.

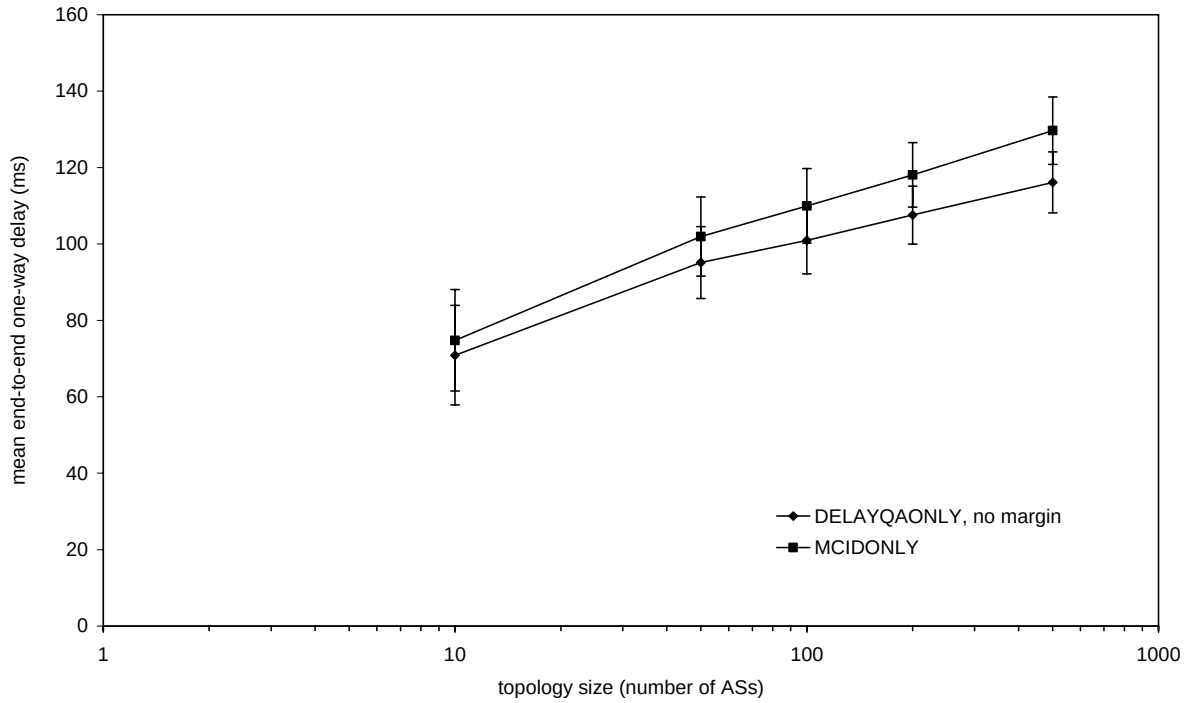


Figure 52 Q-BGP scalability: mean one way delay versus number of ASs

The use of additional QoS info in q-BGP brings an additional overhead in terms of an increased number of q-BGP UPDATE messages. Figure 53 shows the total number of q-BGP messages sent from the first set of bootstrap messages through to a stable routing configuration comparing MCID-only (equivalent to classical BGP) with DELAYQA-only selection policies. It should be noted that no equivalence margin was set for the DELAYQA-only test, and that this represents the worst case in terms of quantities of UPDATE messages generated for this class of q-BGP selection policy. Further, it should be noted that no message aggregation is performed in these simulations, either on network prefixes or QoS attributes.

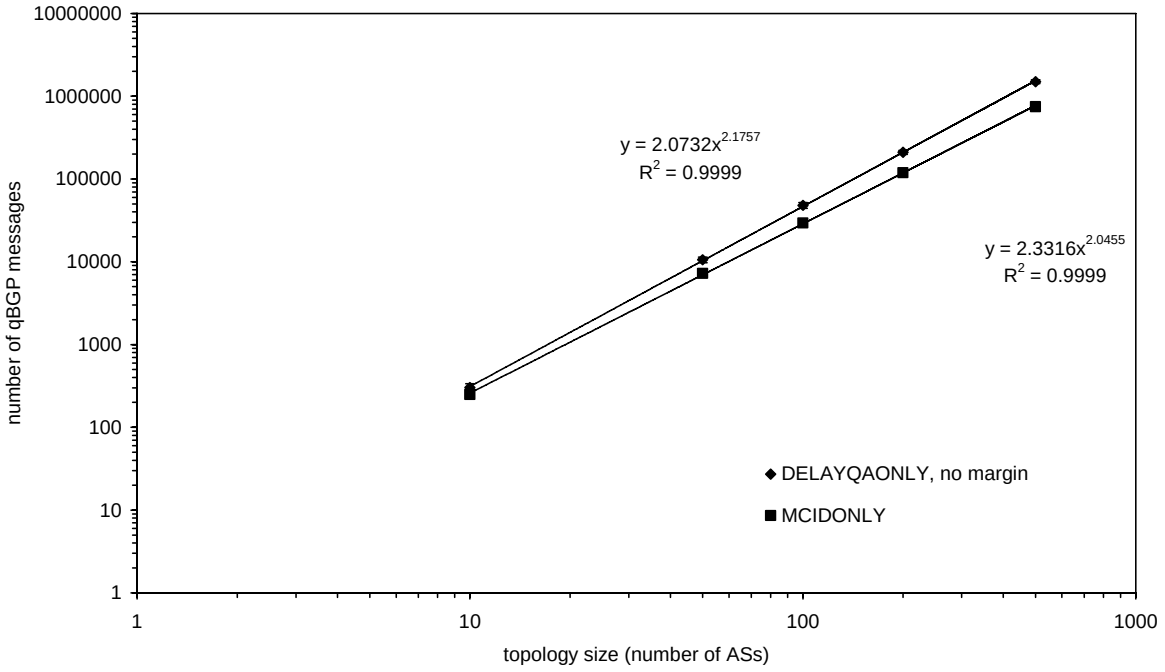


Figure 53 Q-BGP scalability: number of q-BGP messages sent from initialisation until it settles in a stable state with a full mesh of demands applied

When the two plots are extrapolated to a topology size of 18,000 ASs the q-BGP category two routing scheme produces only approximately three times as many messages as q-BGP messages conveying MCID only. The inclusion of additional QoS info 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 MCID 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, on QoS grounds, the preferred AS path 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 could imagine a scheme where an AS would not make immediate decisions, but wait for some period to be sure that it has received all likely updates. This would improve the transient stability of the solution but at the cost of longer convergence times.

4.1.7 Experimental results: stability

Table 9 shows convergence time for a range of q-BGP path selection policies for a topology of 100 ASs with a pSLS provisioning 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	Average number of simulator epochs until convergence
MCID-only	9.5
DELAYQA-only (pOWD=50)	10.4
DELAYQA-only (no margin)	10.8
PRI_DEL_BW (pOWD=100, pBW=75000)	13.2
BWQA-only (pBW = 125000)	16.1
PRI_BW_DEL (pBW=175000, pOWD=50)	16.4
BWQA-only (no margin)	17.6

Table 9 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.

4.1.8 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 meta-QoS-classes – 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.

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.

4.2 Data Plane Testbed Tests

4.2.1 Overview

The data plane testbed experiments were carried out in order to verify that the network is set-up and operates correctly for conducting the tests in further phases. Especially, the objectives of experimentations conducted within this phase are to verify that routing and QoS configuration detailed in [D3.1] are correctly deployed and implemented. In addition, these experiments aim at verifying that policing and shaping policies are correctly configured in all testbed ASs.

4.2.2 Experiment Setup and Test Description

The environment to execute these tests is the MESCAL testbed deployed in FTR&D premises. The testbed is composed of eight ASs and ten Linux-based routers. All ASs are composed of a single Linux-based router except AS4 which is composed of three routers. Four local QoS Classes are configured in each AS. Shaping and policing are configured in testbed routers. BGP is configured to run between two neighbouring ASs. For more detailed information about the configuration of the testbed for this phase refer to section 9. This configuration will be used as it is for executing these tests except when there are explicit recommendations in the procedure tag.

4.2.3 Test Results

In this section, we provide a list of tests that have been carried for this phase. Detailed results are provided in section 10. The experiments carried out during this phase are composed of several test groups (referred to as TB_P1_FUNCT group) that contain the following test suites:

Test Suite Id	Objective
TB_P1_FUNCT/ROUT	This group of test aims at verifying the routing features, especially the activation of BGP and reachability aspects.
TB_P1_FUNCT/DSSW	This group of tests aims at verifying the DSCP swapping operations in ingress and egress of a domain.
TB_P1_FUNCT/SHAP	This group test aims at verifying shaping operation.
TB_P1_FUNCT/POLI	This group of test aims at verifying policing issues.
TB_P1_FUNCT/BWMA	This group of test aims at examining the bandwidth management.

Table 10: Phase 1 Test Suites

The table Table 11 hereafter gives the status of sub group test results:

Test Id	Purpose	Status
TB_P1_FUNCT/ROUT/1	Validate inter-domain link connectivity.	SUCCESSFULLY PASSED
TB_P1_FUNCT/ROUT/2-12	Validate connectivity between two neighbours when BGP process is activated.	SUCCESSFULLY PASSED
TB_P1_FUNCT/ROUT/13	Check the route propagation in a simple scenario.	SUCCESSFULLY PASSED
TB_P1_FUNCT/ROUT/14	Check the reachability of all interfaces.	SUCCESSFULLY PASSED
TB_P1_FUNCT/ROUT/15	Verify reachability status when link failure occurs.	SUCCESSFULLY PASSED
TB_P1_FUNCT/ROUT/16	Verify reachability status when a link failure is re-established.	SUCCESSFULLY PASSED
TB_P1_FUNCT/ROUT/17	Verify intra-domain routing in AS4 domain.	SUCCESSFULLY PASSED
TB_P1_FUNCT/DSSW/1-10	Verify DSCP swapping at egress of MESCAL11, MESCAL21, MESCAL31, MESCAL41, MESCAL42, MESCAL43, MESCAL51, MESCAL61, MESCAL71 and MESCAL81.	SUCCESSFULLY PASSED
TB_P1_FUNCT/DSSW/11-20	Verify DSCP swapping at ingress of MESCAL11, MESCAL21, MESCAL31, MESCAL41, MESCAL42, MESCAL43, MESCAL51, MESCAL61, MESCAL71 and MESCAL81.	SUCCESSFULLY PASSED
TB_P1_FUNCT/DSSW/21	Verify QoS configuration of the whole testbed.	SUCCESSFULLY PASSED
TB_P1_FUNCT/SHAP/1-11	Verify shaping configuration in MESCAL11, MESCAL71, MESCAL81, MESCAL51, MESCAL43, MESCAL41, MESCAL42, MESCAL21, MESCAL31 and MESCAL61.	SUCCESSFULLY PASSED
TB_P1_FUNCT/POLI/1-11	Verify policing configuration in MESCAL11, MESCAL71, MESCAL81, MESCAL51, MESCAL43, MESCAL41, MESCAL42, MESCAL21, MESCAL31 and MESCAL61.	SUCCESSFULLY PASSED
TB_P1_FUNCT/BWMA/1-11	Verify bandwidth management configuration in MESCAL11, MESCAL71, MESCAL81, MESCAL51, MESCAL43, MESCAL41, MESCAL42, MESCAL21, MESCAL31 and MESCAL61.	SUCCESSFULLY PASSED

Table 11: Phase 1 Tests results

4.2.4 Conclusions

The results obtained during this phase certify that the configuration of the testbed is aligned with its objectives especially the following features:

- Configuration of local QoS classes: DSCP, prioritisation, bandwidth pre-emption
- DSCP marking and remarking at ingress and egress interfaces
- Shaping and policing at the boundary of domains
- Bandwidth pre-emption between a meta-QoS-class and BE configured in a given inter domain link
- Routing aspects.

4.3 q-BGP Testbed Tests

4.3.1 Overview

The objectives of experimentations of this phase are mainly: to (1) test the q-BGP messages conformance specifications enclosed in [D1.2], (2) to validate QoS computation as implemented by q-BGP machinery, (3) to validate the route selection process and finally to (4) validate DSCP swapping operations as implemented in q-BGP, especially validate the QoS route-map introduced in ZeboS.

4.3.2 Experiment Setup and Test Description

The environment to execute these tests is the MESCAL testbed that is deployed in FTR&D premises. The configuration of the testbed for this phase is detailed in section 9. This configuration will be used as it is for executing these tests except when there are explicit recommendations in the procedure of the test.

4.3.3 Test Results

In this section, we provide a list of tests that have been carried for this phase. Detailed results are provided in section 10. The experiments carried out during this phase are composed of several test groups (referred to as TB_P2_FUNCT group) that contain the following test suites:

Test Suite Id	Objective
TB_P2_FUNCT/CMES	This group of test aims at verifying the conformance of q-BGP messages such as defined in [D1.2].
TB_P2_FUNCT/DSCP	This group of tests aims at verifying DSCP swapping operations for ingress and egress.
TB_P2_FUNCT/QCMP	This group tests aims at verifying basic computation of QoS information between two peering ASs.
TB_P2_FUNCT/RSEL	This group of tests aims at verifying the behaviour of q-BGP route selection algorithm such as defined in [D1.2]. It also verifies more complex computation of QoS information.
TB_P2_FUNCT/QFIB	This group of tests aims at verifying that QoS-enabled entries are correctly installed in FIB via q-BGP.
TB_P2_FUNCT/INT	This group of tests aims at verifying the interoperability of q-BGP and BGP.

Table 12: Phase 2 Validation Test Suites

TB_P2_FUNCT group contains the tests in Table 13.

Test Id	Purpose	Status
TB_P2_FUNCT/CMES/1	Verify the capability length.	SUCCESSFULLY PASSED
TB_P2_FUNCT/CMES/2	Verify the QoS service capability field length.	SUCCESSFULLY PASSED
TB_P2_FUNCT/CMES/3	Verify that Group 1 QoS service capability is supported.	SUCCESSFULLY PASSED
TB_P2_FUNCT/CMES/4	Verify that Group 2 QoS service capability is supported.	SUCCESSFULLY PASSED
TB_P2_FUNCT/CMES/5	Validate the conformance of QoS information length.	SUCCESSFULLY PASSED

TB_P2_FUNCT/CMES/6	Verify that "Packet Rate QoS Code" and its associated Sub-codes are supported.	SUCCESSFULLY PASSED
TB_P2_FUNCT/CMES/7	Validate that "One Way Delay QoS Code" and its associated Sub-codes are supported.	SUCCESSFULLY PASSED
TB_P2_FUNCT/CMES/8	Validate that "Inter-Packet Delay Variation QoS Code" and its associated Sub-codes are supported.	SUCCESSFULLY PASSED
TB_P2_FUNCT/CMES/9	Validate the QoS information value.	SUCCESSFULLY PASSED
TB_P2_FUNCT/CMES/10	Verify that QoS class identifier can be set to a value that is between 0 and 63.	SUCCESSFULLY PASSED
TB_P2_FUNCT/CMES/11	Validate the QoS Origin field.	SUCCESSFULLY PASSED
TB_P2_FUNCT/CMES/12	Check the validity of Address Family Identifier (AFI).	SUCCESSFULLY PASSED
TB_P2_FUNCT/CMES/13	Check the validity of Subsequent Address Family Identifier (SAFI).	SUCCESSFULLY PASSED
TB_P2_FUNCT/CMES/14	Check the validity of Network Address of Next Hop.	SUCCESSFULLY PASSED
TB_P2_FUNCT/CMES/15	Verify the conformance of NLRI field.	SUCCESSFULLY PASSED
TB_P2_FUNCT/DSCP/1-2	Validate that egress DSCP swapping operation is correctly achieved when receiving BGP UPDATE messages.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/1	Check that the *reserved-rate* QoS parameter is correctly computed by the receiving ASBR.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/2	Check that invalid *reserved-rate* values are rejected by the command-line interface.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/3	Check that the *available-rate* QoS parameter is correctly computed by the receiving ASBR.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/4	Check that invalid *available-rate* values are rejected by the command-line interface.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/5	Check that the *min-owd* (minimum one-way-delay) QoS parameter is correctly computed by the receiving ASBR.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/6	Check that invalid *min-owd* values are rejected but the command-line interface.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/7	Check that the *max-owd* (maximum one-way-delay) QoS parameter is correctly computed by the receiving ASBR.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/8	Check that invalid *max-owd* values are rejected by the command-line interface.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/9	Check that the *average-owd* (average one-way-delay) QoS parameter is correctly computed by the receiving ASBR	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/10	Check that invalid *average-owd* values are rejected by the command-line interface.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/11	Check that the *loss-rate* QoS parameter is correctly computed by the receiving ASBR.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/12	Check that invalid *loss-rate* values are rejected by the command-line interface.	SUCCESSFULLY PASSED

TB_P2_FUNCT/QCMP/13	Check that the *jitter* QoS parameter is correctly computed by the receiving ASBR.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/14	Check that invalid *jitter* values are rejected by the command-line interface.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/15	Check that the receiving ASBR is able to compute multiple QoS parameters contained in an announcement.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QCMP/16	Check that the receiving ASBR is able to compute multiple QoS parameters for a same prefix announced within different meta-QoS-planes.	SUCCESSFULLY PASSED
TB_P2_FUNCT/RSEL/1	Check that several ASs involved in the loose service option are able to exchange route updates containing correctly computed QoS information.	SUCCESSFULLY PASSED
TB_P2_FUNCT/RSEL/2	Check, in simple Scenarios, that the route selection process takes into account the priority level of each QoS attribute.	SUCCESSFULLY PASSED
TB_P2_FUNCT/RSEL/3	Check that the route selection process takes into account the QoS attributes which have a lower priority when the previous attributes (with higher priority) are equivalent.	SUCCESSFULLY PASSED
TB_P2_FUNCT/RSEL/4	Check that the precision command line parameter is correctly handled for the reserved-rate QoS attribute.	SUCCESSFULLY PASSED
TB_P2_FUNCT/RSEL/5	Check that the precision command line parameter is correctly handled for the available-rate QoS attribute.	SUCCESSFULLY PASSED
TB_P2_FUNCT/RSEL/6	Check that the precision command line parameter is correctly handled for the loss-rate QoS attribute.	SUCCESSFULLY PASSED
TB_P2_FUNCT/RSEL/7	Check that the precision command line parameter is correctly handled for the min-owd QoS attribute.	SUCCESSFULLY PASSED
TB_P2_FUNCT/RSEL/8	Check that the precision command line parameter is correctly handled for the max-owd QoS attribute.	SUCCESSFULLY PASSED
TB_P2_FUNCT/RSEL/9	Check that the precision command line parameter is correctly handled for the average-owd QoS attribute.	SUCCESSFULLY PASSED
TB_P2_FUNCT/RSEL/10	Check that the precision command line parameter is correctly handled for the jitter QoS attribute.	SUCCESSFULLY PASSED
TB_P2_FUNCT/RSEL/11-17	Validate the behaviour of q-BGP when mandatory parameters aren't received.	SUCCESSFULLY PASSED
TB_P2_FUNCT/RSEL/18-24	Validate the behaviour of q-BGP when optional parameters aren't received.	SUCCESSFULLY PASSED
TB_P2_FUNCT/QFIB	A series of elementary tests will be carried out in order to verify the correct installation of QoS-based routes in the q-FIB table.	SUCCESSFULLY PASSED
TB_P2_FUNCT/INT/1	Validate the behaviour of a BGP speaker when receiving unrecognised optional parameters.	SUCCESSFULLY PASSED
TB_P2_FUNCT/INT/2	Validate the behaviour of a q-BGP speaker when receiving notification set to unsupported capabilities from BGP speaker.	SUCCESSFULLY PASSED
TB_P2_FUNCT/INT/3	Validate the q-BGP router installs routes received from BGP speaker in best effort plane.	SUCCESSFULLY PASSED
TB_P2_FUNCT/INT/4	Validate the BGP router installs routes received from q-BGP speaker.	SUCCESSFULLY PASSED

Table 13: Phase 2 Validation Tests results

4.3.4 Conclusions

The obtained results of this phase experiments certify that:

- q-BGP implementation is aligned with specifications
- q-BGP QC-Id swapping operation are correctly implemented
- QoS computation as achieved by q-BGP conforms with specification
- q-BGP route selection process conforms to what has been specified in [D1.3]
- q-BGP interoperates with classical BGP
- QoS-enabled routes could be installed in q-FIB and that q-BGP can successfully install QoS-enabled entries in q-FIB.

4.4 PCS Testbed Tests

4.4.1 Overview

The objectives of the experimentations of this phase are as follows:

- Test the PCP message conformance with what has been specified in [D1.2]
- Validate the QoS computation as implemented by PCS machinery
- Validate the route selection process
- Validate the interface between q-BGP and PCS
- Validate resource reservation and release

4.4.2 Experiment setup and test description

The environment to execute these tests is the MESCAL testbed that is deployed in FTR&D premises. The configuration of the testbed for this phase is detailed in section 9. This configuration will be used as it is for executing these tests except when there are explicit recommendations in the procedure of the test.

4.4.3 Test Results

TB_P3_FUNCT group contains the following test suites:

Test Suite Id	Objective
TB_P3_FUNCT/CMES	This group of test aims at verifying message conformance of PCP to what is specified in [D1.2].
TB_P3_FUNCT/QAGG	This group of tests aims at verifying QoS aggregation operations as achieved by PCE entities.
TB_P3_FUNCT/RESAV	This group if tests aims at verifying the reservation operations when a path has been computed by a PCE

Table 14: Phase 3 Validation Test Suites

TB_P3_FUNCT group contains the tests in Table 15.

Test Id	Purpose	Status
TB_P3_FUNCT/CMES/1	Check the format of OPEN, CLOSE and ACCEPT messages.	SUCCESSFULLY PASSED
TB_P3_FUNCT/CMES/2	Check the format of REQUEST, RESPONSE PATH-ERROR and ACKNOWLEDGE messages.	SUCCESSFULLY PASSED
TB_P3_FUNCT/CMES/3	Validate the REQ-REFERENCE-ID and PATH-COMPUTATION-ID.	SUCCESSFULLY PASSED
TB_P3_FUNCT/CMES/4	Validate QoS information contained in REQUEST-PATH message.	SUCCESSFULLY PASSED
TB_P3_FUNCT/CMES/5	Validate QoS information contained in RESPONSE-PATH message.	SUCCESSFULLY PASSED
TB_P3_FUNCT/CMES/6	Check the format of PATH-ERROR and messages.	SUCCESSFULLY PASSED
TB_P3_FUNCT/CMES/7	Check the format of CANCEL and messages.	SUCCESSFULLY PASSED
TB_P3_FUNCT/CMES/8	Check operational behaviours when receiving REQUEST messages.	SUCCESSFULLY PASSED
TB_P3_FUNCT/QAGG/1	Check QoS aggregation operation.	SUCCESSFULLY PASSED
TB_P3_FUNCT/RESAV/1	Check resource reservation	SUCCESSFULLY PASSED
TB_P3_FUNCT/RESAV/2-3	Check resource release when the order is cancelled by user and when the validity date expires.	SUCCESSFULLY PASSED

Table 15: Phase 3 Validation Tests results

4.4.4 Conclusions

In this phase, we have tested the MESCAL PathCompSys implementation, especially the following features:

- Configuration of path computation orders;
- Interface between PCEs and routing;
- Computation of inter-domain paths satisfying a set of QoS performance characteristics;
- The conformance of the PCP implementation.

These features are aligned with the specification.

5 SERVICE MANAGEMENT TESTS AND RESULTS

5.1 pSLS Ordering Tests

The role of the *pSLS Ordering* functional block (see section 4.5 of [D1.3]) is to establish the set of pSLS agreements, the most advantageous to the AS with respect to Traffic Engineering and business objectives. The high-level experimentation objectives for the *pSLS Ordering* functional block are:

- Functional validation of the prototype implementation;
- Verification of the convergence of the collective agreement optimisation logic;
- Assessment of the impact of environment complexity upon the scalability of the approach;
- Gaining insight on inherent benefit/cost tradeoffs of the collective agreement optimisation.

The performance metrics, the controlled and uncontrolled variables considered for the *pSLS Ordering* experimentation are listed in Table 16 and Table 17.

Performance Metrics	
negotiation logic processing time (ProcT.NLogic)	The processing time of the pSLS Ordering negotiation logic per round.
SrNP processing time (ProcT.SrNP)	The processing time of the SrNP engine for handling send message requests.
order execution time (OrdExecT)	The time elapsed between the reception of a *Negotiation Plan* and the completion of negotiations.
agreement optimality (AgrOptimality)	The confirmed cost at a round of the established order over the minimum possible cost, assuming a full knowledge of the providers' cost function.

Table 16: pSLS Ordering Performance Metrics

Controlled Variables	
maximum number of rounds (MaxRounds)	The maximum number of negotiation rounds the negotiation logic is permitted to undertake before concluding the execution of an order.
Uncontrolled Variables	
target limit (TrgL)	The limit each pSLS is bound to contribute to the order, takes values in (0,1]. A target limit of 1/3 for instance, signifies to the logic that each pSLS cannot exceed 1/3 of the total bandwidth required for a particular order; hence only combinations of three or more pSLSs may implement this order. We consider the distinct values of 1/3, 2/3 and 3/3.
number of cost areas (CAreas)	Assuming a linear step-wise decreasing cost function, the number of discontinuities on the cost function of a provider for a service type. In other words, the number of different values of cost per unit of a provider for a service type. We consider just one area, 4 areas and 20 areas.
delta between cost areas (CDAreas)	The difference factor of the cost per unit between adjacent cost areas; we consider 1.5 and 3 as values.
delta between providers (CDProviders)	The maximum difference factor of the cost per unit between providers; we consider 2 and 10 as values.

Table 17: pSLS Ordering Variables

5.1.1.1 Experimentation Environment

The test platform is composed by the *pSLS Ordering* prototype, a reduced version of the *SLS Order Handling* prototype, the *Negotiation Plan Generator* acting on behalf of *Binding Selection* block and the *SLS Order Handling Response Generator* testing tools (see Figure 54).

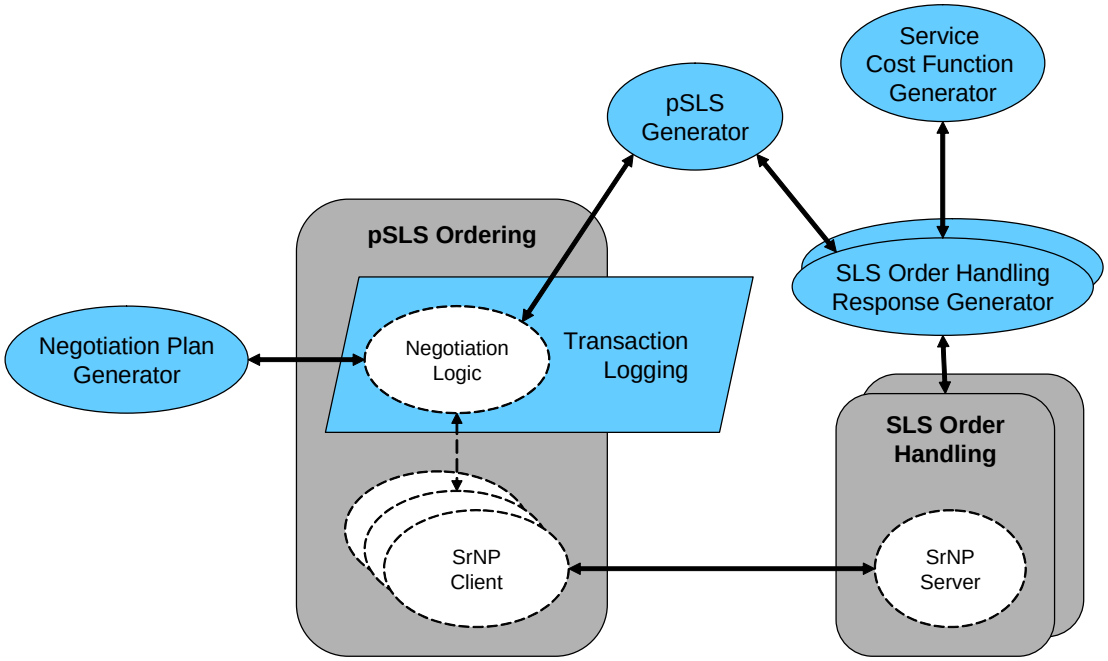


Figure 54: pSLS Ordering Experimentation Environment

An experiment corresponds to the execution of one order, expressed in the *Negotiation Plan*. However, multiple dynamic interactions with a number of *SLS Order Handling* servers may take place for the completion of the order execution and the conclusion of the experiment.

The ***Negotiation Plan Generator*** produces the negotiation plans to be fed to the *pSLS Ordering*. It can be configured to produce negotiation plans targeting a configurable number of pSLSs and providers. The acceptance criteria restrict the total bandwidth to be a fixed value (100 units) and the total cost less than a maximum value, aligned with the problem definition of *pSLS Ordering* negotiation logic (see [D1.3]). Target tolerance criteria automatically restrict the bandwidth per pSLS to be at maximum the *Target Limit* (see Table 17) times the order bandwidth of 100 units.

The pSLSs to be negotiated at each negotiation round are generated by the ***pSLS Generator*** function and can be parameterised based 1) on the pSLS type (Provider Loose QoS and Provider Loose QoS Tunnels, Peer Loose QoS and Peer Loose QoS Tunnels, Proxy Statistical QoS and Proxy Statistical QoS Tunnels), 2) the boundary link, 3) the QoS-class, 4) the destination labels or IP prefixes, 5) the bandwidth and 6) the cost. A simplified version used in the bulk of the experimentation activities generates pSLSs of Provider Loose QoS type for premium meta-QoS-class based on the provider identifier, the bandwidth and the cost. *pSLS Generator* is also used in functional validation of the *SLS Order Handling* function (see section 5.2).

The *SLS Order Handling* prototype is reduced and contains only its *SrNP server* engine which, instead of the *Admission Logic*, it is now controlled by the ***SLS Order Handling Response Generator***. The *SLS Order Handling Response Generator* operates on the basis of the service cost function generated by the ***Service Cost Function Generator***. A cost function for i-th service is of the form

$$fc_i(x) = \begin{cases} c_{i1} & 0 < x \le x_{i1} \\ c_{i2} & x_{i1} < x \le x_{i2} \\ \vdots & \vdots \\ c_{ik} & x_{i(k-1)} < x \le x_{ik} \end{cases}$$

where x is the bandwidth and $fc_i(x)$ the cost per bandwidth unit. The *Service Cost Function Generator* generates cost functions with definite number of cost function areas k equal to the value of the *CAreas* variable with $c_{ij} / c_{i(j+1)}$ always equal to *CDAreas* (see Table 17). Note that, as long as *CDAreas* > 1, the lesser the bandwidth the bigger the cost per unit, i.e. the function of cost per bandwidth unit monotonically decreases. Finally, the base cost c_{i1} for the cost function of each of the configured *pSLSs* is determined randomly based on a uniform distribution in the range of [1, *CDProviders* * C_{base}].

The *SLS Order Handling Response Generator* will reply to any Proposal received from the *pSLS Ordering* for bandwidth in (x_{ij}, $x_{i(j+1)}$] and cost unspecified or other than $c_{i(j+1)}$, with a Revision SrNP message containing a pSLS generated by the *pSLS Generator* with the requested bandwidth and cost set to $c_{i(j+1)}$. On a Proposal or BindProposal SrNP message with bandwidth in (x_{ij}, $x_{i(j+1)}$] and cost set to $c_{i(j+1)}$, it will reply with SrNP Accept or AgreeProposal message respectively. An *SLS Order Handling Response Generator* runs per provider among the configured *Providers*.

In order to test the negotiation logic, appropriate ***Transaction Logging*** is added, so that the calculated costs and the processing time can be tracked down per negotiation round.

5.1.2 Experiment Setup and Test Description

To facilitate experimentation we focus to a representative set of test configuration options (see Table 18). The maximum number of rounds *MaxRounds* is set to infinite. The number of pSLSs to concurrently pursue is fixed to three.

The undertaken tests are described in detail in Table 18. Controlled or uncontrolled variables left unspecified in a test description are set to appropriate fixed values so that they have no impact upon the subject under testing.

<table>
<tr><th></th><th>id</th><th>provider delta</th><th>cost areas</th><th>cost area delta</th><th>target limit</th></tr>
<tr><td>TestSetup#1</td><td>2A[1,2,3]</td><td>2</td><td rowspan="2">1</td><td rowspan="2"></td><td rowspan="8">{1/3, 2/3, 3/3}</td></tr>
<tr><td>TestSetup#2</td><td>10A[1,2,3]</td><td>10</td></tr>
<tr><td>TestSetup#3</td><td>2B[1,2,3]</td><td>2</td><td rowspan="4">4</td><td rowspan="2">1.5</td></tr>
<tr><td>TestSetup#4</td><td>10B[1,2,3]</td><td>10</td></tr>
<tr><td>TestSetup#5</td><td>2C[1,2,3]</td><td>2</td><td rowspan="2">3</td></tr>
<tr><td>TestSetup#6</td><td>10C[1,2,3]</td><td>10</td></tr>
<tr><td>TestSetup#7</td><td>2D[1,2,3]</td><td>2</td><td rowspan="2">20</td><td rowspan="2">3</td></tr>
<tr><td>TestSetup#8</td><td>10D[1,2,3]</td><td>10</td></tr>
</table>

Table 18: Test Configurations

5.1.3 Test Results

First, we assess the evolution of optimality and processing time through the negotiation rounds.

For test 10C2, Figure 55 presents per negotiation round the estimated and the confirmed cost, which are normalised over the actual minimum, theoretically calculated, cost. The estimated cost is the cost that the negotiation logic computes at each round based on the knowledge of the cost gained from the providers from the requests made at previous rounds; note that the logic builds on the assumption that the cost-rate function of the providers is decreasing. The confirmed cost at a negotiation round is the cost already agreed with the providers –the estimated cost accepted by the providers at a previous round.

As we can see from Figure 55, and actually is the case in all other conducted tests, the confirmed cost decreases, while the estimated cost increases as we progress the negotiation procedure, until they both converge to the same value, which is the optimum, minimum possible, cost value. The decrease of the confirmed cost is logical as this constitutes the mere objective of the logic. The increase of the estimated cost can be explained because of the optimistic nature of the logic and the fact the providers' cost-rate functions are decreasing. Departing from an initial valid solution, the logic asks for more bandwidth from what thought to be the cheaper providers, however this might not be the case –the providers might respond with higher than the estimated values. Should no estimates on feasible combinations yielding less than the confirmed cost can be found, the logic concludes with the last confirmed cost, which proves to be the minimum possible cost –assuming decreasing linear step-wise cost functions per provider. This proves the validity of the specified algorithm as well its stability - convergence.

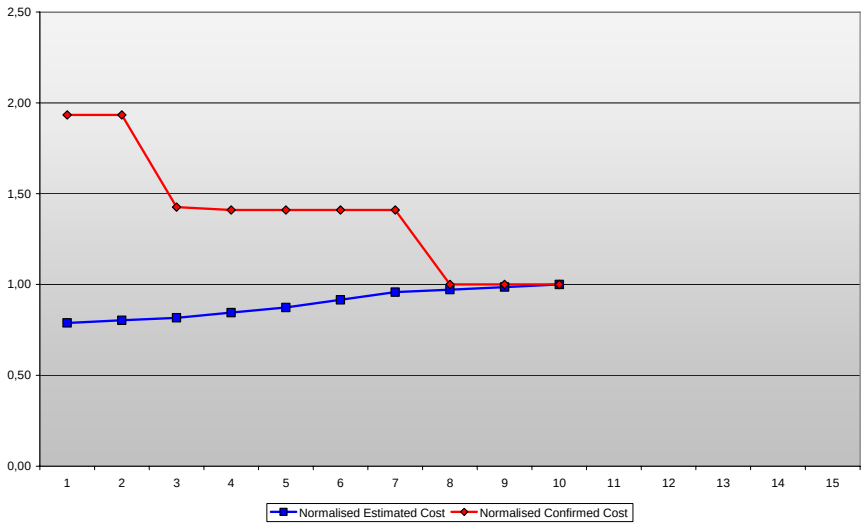


Figure 55 Evolution of optimality over negotiation rounds

For the above test, 10C2, Figure 56 depicts the processing time of the negotiation logic and the time consumed in SrNP-interactions as a function of the negotiation rounds. As we can observe, the time required for the logic to execute increases as negotiations progress, whereas the time spent on protocol interactions decreases. The same behaviour is noticed throughout all conducted tests. This behaviour was anticipated as the number of possible valid combinations to yield an estimated cost increases with the growing of knowledge of actual cost the providers can offer, gained from the previous rounds.

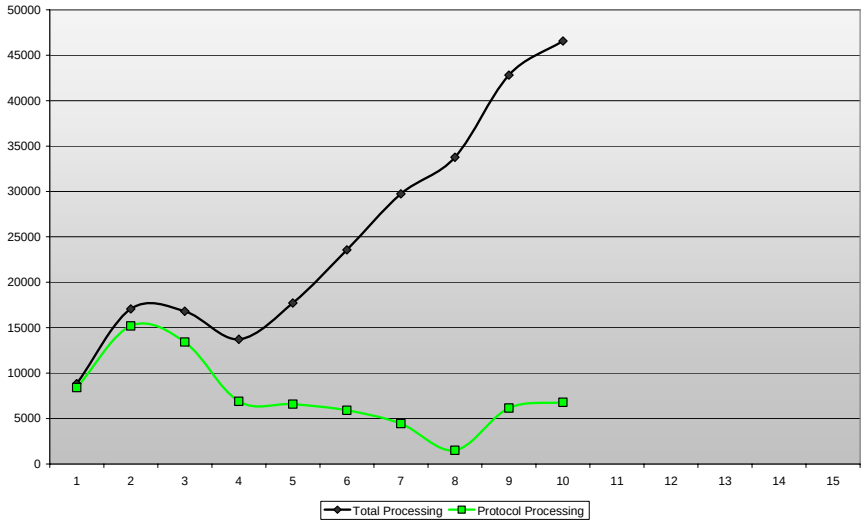


Figure 56 Evolution of processing time over negotiation rounds

Next, we try to assess the rate of decrease of the confirmed cost, therefore the acceleration in dropping down the cost of valid solutions found, over the negotiation rounds in relation to the parameters identified to influence the complexity of the negotiation logic.

Figure 57 depicts the normalised confirmed cost over its maximum value established at the first negotiation round, per round over 6 different test configurations; from top to down the 2B2 and 2B10, 2C2 and 2C10 and 2D2 and 2D10 test configurations. The three sets of these test configurations differ amongst them in the number of cost areas and the diversity of the cost-rate values per area; the two tests configurations within each of the three sets differ in the diversity of the cost-rate values amongst the providers.

As we can see, the more the diversity in the cost values per provider, the greater the drop in the confirmed cost. This is a reasonable result, considering that in cases where there is not significant difference in the offers made by a provider, the first agreed cost will be close to the minimum possible that can be found. From this result, we can also notice that the proposed algorithmic logic indeed pursues to the end of finding the minimum possible cost and indeed achieves that –assuming a liner step-wise decrease cost function per provider. Last, an observation to be made is that the diversity in the cost amongst the providers does not impact the dropping of cost from round to round, compared to the effect of cost diversity of a provider.

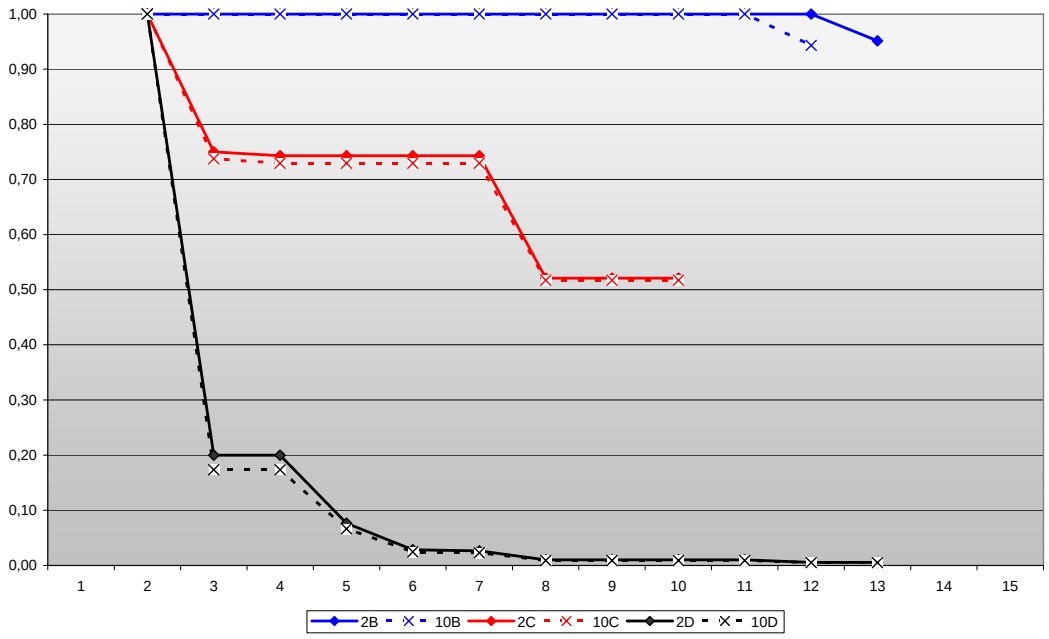


Figure 57 Rate of decrease of the confirmed cost

Following, we examine the effect of the parameters identified to affect the behaviour of the negotiation logic on the 'speed of convergence' represented by the number of negotiation rounds required to conclude successfully –achieve the minimum cost solution.

Figure 58 shows the number of negotiation rounds to successful conclusion for the conducted tests, which were executed in three cases, 1, 2 and 3 corresponding to the three different values for the target limit considered, 3/3, 2/3 and 1/3. These percentages mean that the logic can buy from each of the three providers only up to these percentages of the totally required bandwidth. Considering that the logic has been built assuming a linear step-wise cost function per provider, which indeed is the case in the conducted tests, we see that in the cases of 1/3 and 3/3 the logic concludes very fast. In these cases, the number of valid combinations of the amounts of bandwidth to purchase from each provider is limited by the constraint underlying the test case, case of 1/3, or the first to try to purchase indicates the cheapest solution, case of 3/3 –buy all from the cheapest. However, in cases, as in the case of 2/3, that the number of valid combinations is not limited or the cheapest solution cannot be identified, the logic needs to go through more steps for asking/identifying possible better combinations.

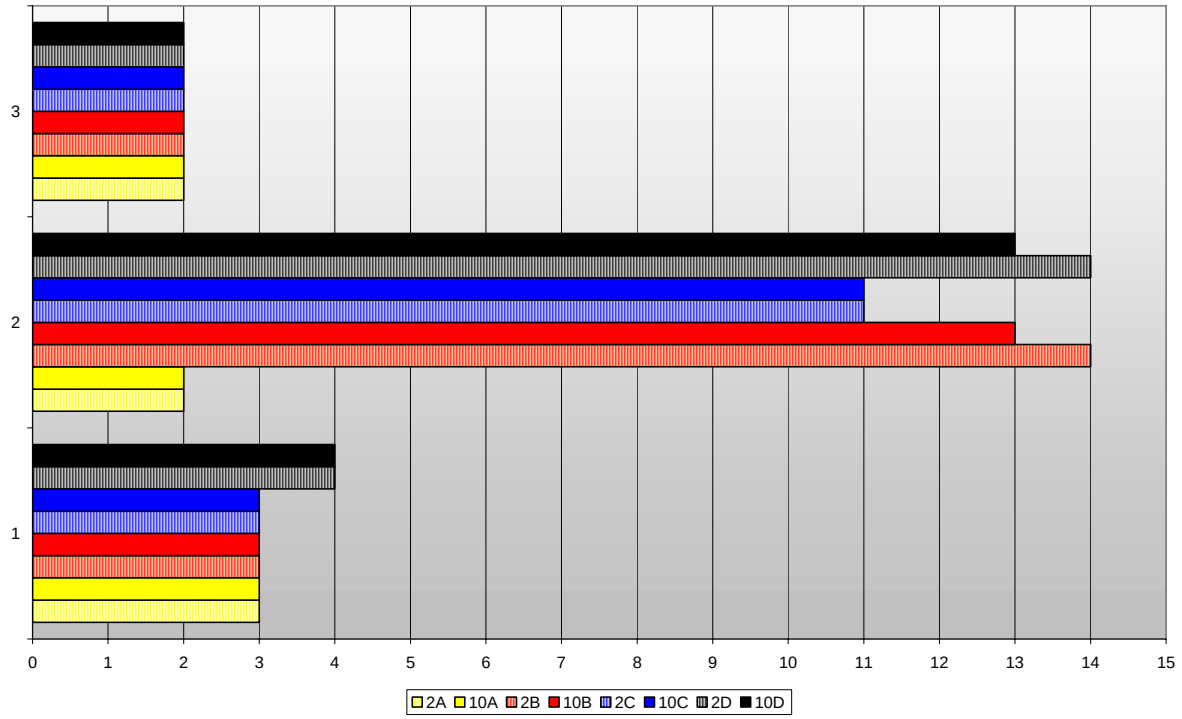


Figure 58 Number of negotiation rounds for successful conclusion

Last, we try to examine how the processing time of the logic is affected by the complexity of the external environment.

Figure 59 depicts the processing time of the negotiation logic as a function of the negotiation rounds in a number of test configurations representing different cases of complexity regarding the providers' cost functions. The following observations can be made. The processing time increases with the number of negotiations rounds, as we have seen before. The more the diversity in the cost functions of the providers the faster the increase of the processing time. The processing time increases from round to round not in a linear fashion; it is affected by both the diversity of the cost values within a provider as well as by the diversity of the cost values amongst providers. Note that as we have seen in a previous test, the diversity of the cost function within providers and not between providers mainly affects the number of negotiation rounds required to conclude the negotiations.

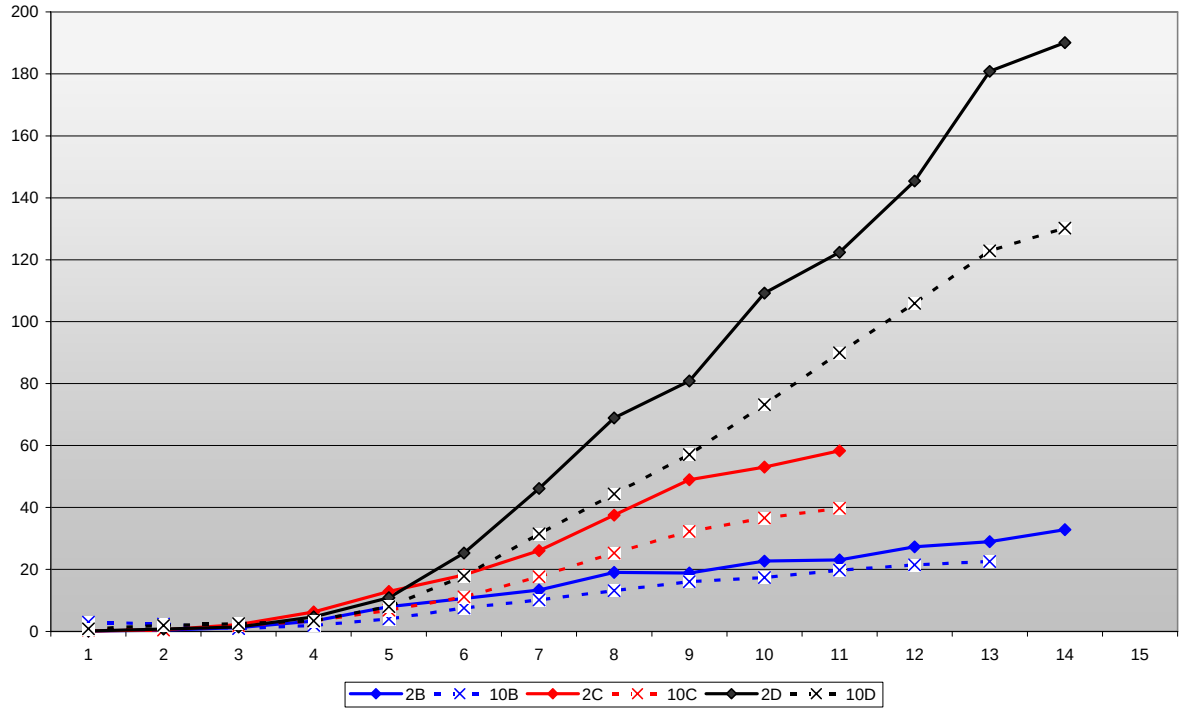


Figure 59 Processing time of the negotiation logic

5.1.4 Conclusions

The tests carried out show that it is possible to conduct negotiations in an automated fashion, proving the validity and feasibility of the proposed ordering and negotiation framework.

We showed that an algorithmic negotiation logic can be built and operated on top of the specified negotiations protocol SrNP, which has been designed to support any kind of negotiation logic. The negotiation logic we built addresses a particular case for negotiations and it has been demonstrated that can indeed achieve the optimum bargain, provided that its assumptions regarding the providers' cost functions hold. In this set-up, a number of tests were carried out for assessing its scalability and stability in relation to various parameters representing the complexity of the external environment in which is to operate –diversity of cost functions in terms of their values per and amongst providers. The tests yielded reasonable and justifiable results, further advocating the validity of automating the negotiation logic.

From the results of the particular negotiation case we implemented, we can draw the following:

The optimality of negotiation logic depends on the amount of prior knowledge one can have regarding the decision making logic of the parties to negotiate. As demonstrated, assuming that all providers employ a specific type of cost function, step-wise linear decreasing, an algorithm for conducting negotiations to find the best solution/bargain can be built and conclude successfully in a scalable and stable fashion.

Assuming that the 'trend' of the decision-making logic of the other negotiating parties is known, then negotiation logic can be built to yield the optimum solution. Given a known trend, the larger the diversity in the values of the benefit-metrics (e.g. cost) of the issues under negotiations, the more the chances in achieving a better beneficial agreement, however at the expense of increased number of negotiation rounds and processing time.

The complexity of negotiation logic depends on the complexity of the decision making logic of the other negotiating parties and grows from negotiation round to negotiation round in a not linear fashion.

5.2 SLS Order Handling Tests

5.2.1 Objectives

The *SLS Order Handling* functional block (see section 4.4 of [D1.2]) conducts negotiations with *pSLS Ordering* so that the best matching between service requests and available resources is achieved. The *SLS Order Handling* is decomposed into four major functions (see Figure 60): the *Negotiations Server*, the *SLS Translation*, the *Admission Logic* and the *SLS Establishment* functions.

The *Negotiations Server* function conducts negotiations for all pending SLS orders from different customers in parallel, using the underlying negotiation protocol. The *SLS Translation* function translates and maps the SLSs contained in an SLS order into their network view; further it performs validity checks against already established SLSs, primarily for ensuring uniqueness of customer/users identification. The *Admission Logic* compares the anticipated demand of the requested pSLS and of the already established pSLSs against the available resources provided in the *Resource Availability Matrices* and decides accordingly the acceptance of the requested pSLS. The accepted pSLSs are established via the *SLS Establishment* function responsible for maintaining the *pSLSs* repository and for communicating the relevant information to other system components that need to be updated.

5.2.2 Controlled and Uncontrolled Variables

Controlled Variables	
Satisfaction Level (SL)	The *Satisfaction Level* operational parameter of the *Admission Control* function (see section 4.4.3.1 of [D1.2]). Its permissible value range is [-1,1], however, for experimentation purpose we consider only the following three values: #1 *NoGuarantees* (-1) #2 *AlmostSatisfied* (0) #3 *FullySatisfied* (+1)

Table 19: SLS Order Handling Controlled Variables

Uncontrolled Variables		
Resource Availability (Availability)		The availability of network resources as provided in the *Resource Availability Matrices*. We consider *low* and *unlimited* settings.
Traffic Matrices size[1] *(TMSize)*	*number of lQCs (lQCs)*	The number of lQCs is one dimension of the internal Traffic Matrix. We only consider the fixed set of three lQCs: *Premium*, *Better-Than-Best-Effort* and *Best-Effort* lQCs.
	number of external interfaces (ExtInterfaces)	The number of external interfaces is the second dimension of the internal Traffic Matrix.

[1] The Resource Availability Matrices (see section 5.1.2.3 of [D1.2]) have the same size as the Traffic Matrices (see section 5.1.2.2 of [D1.2]).

	number of oQCs (oQCs)	The number of oQCs is one dimension of the external Traffic Matrix. We only consider the fixed set of three oQCs: *Premium*, *Better-Than-Best-Effort* and *Best-Effort* oQCs.
	number of external destination prefixes (ExtDestPrefixes)	The number of destination prefixes outside the AS is the second dimension of the external Traffic Matrix. The external destination prefixes result from the established pSLSs, hence their number is analogous to the number of the established pSLSs.
Traffic Forecast Parameters	*number of service classes (SrvClasses)*	Traffic forecast parameters such as *Multiplexing Factor* (MF) and *Aggregation Weight* (AW) refer to a *service class*. A service class groups a homogeneous set of traffic flows allowing for aggregation under the assumed service usage and traffic source patterns. The greater the number of service classes, the more granular the classification of traffic sources, hence the more homogenous the set of traffic flows and the more accurate the traffic forecast result. We consider just *one* or *many* service classes.
SLS Orders	*number of ordered SLSs (SLSsOrdered)*	The total number of non alternative SLSs contained in every SLS order placed during the experiment. Assuming only valid SLSs and *Availability* set to *infinite* then at the end of experiment there will be *SLSsOrdered* number of established SLSs.
	service types (SrvTypes)	The type of the service the SLS order belongs to. The supported service types are: #1 Internet access at loose QoS #2 Loose QoS tunnels in the Internet #3 Traffic inter-exchange at a loose QoS #4 Loose QoS tunnel extension #5 Internet access at a statistically guaranteed QoS #6 Statistically guaranteed QoS tunnels in the Internet The *all* setting signifies all supported service types may be used.

Table 20: SLS Order Handling Uncontrolled Variables

To facilitate experimentation we focus to a representative set of test configuration options for *TMSize* variable (see Table 21).

Traffic Matrices size (*TMsize*) test configuration option	***ExtInterfaces***	***ExtDestPrefixes***
#1 Small	small	Small
#2 Medium	medium	Medium
#3 LargeAtDestinations	medium	Large
#4 LargetAtEdges	large	Medium
#5 LargeAll	large	Large

Table 21: *Traffic Matrices Size* Test Configuration Options

5.2.3 Experimentation Environment

The test platform used is depicted in Figure 60.

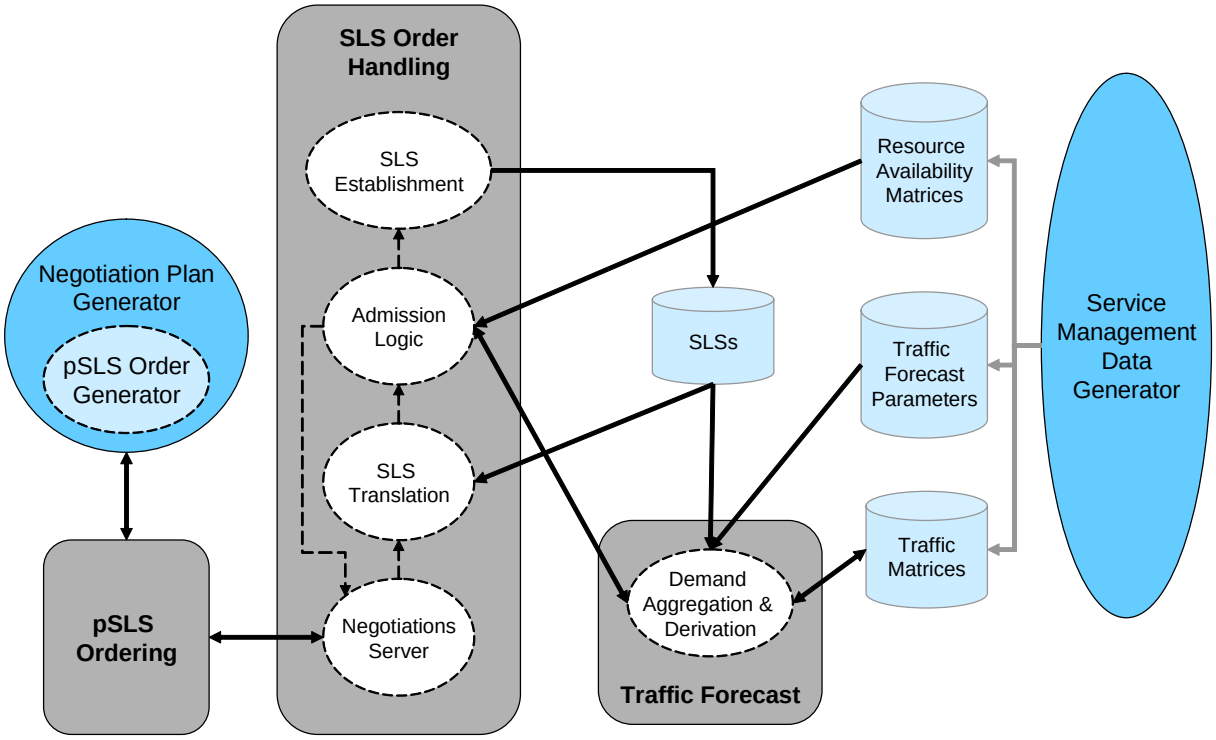


Figure 60: SLS Order Handling Experimentation Environment

5.2.4 Test Campaigns and Results

The tests undertaken are organised under the test suites outlined in Table 22 and described in detail in Table 23. All these tests have been successfully undertaken.

Test Suite Id	Objective
SLSOrderH/Funct/NServer	Functional validation of the implementation of the *Negotiations Server* function.
SLSOrderH/Funct/Translation	Functional validation of the service type dependent functions, namely *pSLS Translation* and *Establishment* functions.
SLSOrderH/Funct/Admission	Functional validation of the *Admission Control* function.

Table 22: SLS Order Handling Test Suites

Test Id	Purpose	Description
SLSOrderH/Funct/NServer	Verify handling of multiple parallel pSLS orders. Verify the FSM implementation is according to specifications.	Runtime parameters: *Diversity* fixed to *significant* covering all protocol message combinations, invalid ones too *Customers* fixed to *many* Test: Check the negotiation protocol embedded in *Negotiations Server* function works as specified.

SLSOrderH/Funct/Translation/1	Verify the implementation of pSLS translation and validation.	Runtime parameters: *SrvTypes* fixed to *all* Ordered pSLSs flow identification clauses configured to overlap and cause validation failure Test: Check pSLSs are translated correctly to the corresponding network view data structures. Ensure validity checks indeed fail.
SLSOrderH/Funct/Translation/2	Verify the implementation of pSLS validation and SLS establishment. Verify derivation of TE information required for q-BGP configuration in the testbed.	Static parameters: *SrvClasses* fixed to *many* Runtime parameters: *SrvTypes* fixed to *all* Test: Ensure validity checks indeed succeed. Check *Demand Aggregation and Derivation* calculations are in accordance to the provided traffic parameters. Check SLS establishment produces information as expected by the *SLS Invocation Handling* and the *Dynamic Inter-domain TE* components.
SLSOrderH/Funct/Admission	Verify the implementation of the admission control algorithm is according to specifications.	Static parameters: *Availability* varying between *low* and *unlimited* Runtime parameters: *SL* varying between *NoGuarantees*, *AlmostSatisfied* and *FullySatisfied* values Test: Check requested SLSs are admitted within the availability buffer as resized by the *SL*, while SLSs exceeding that buffer are rejected.

Table 23: SLS Order Handling Tests

5.2.5 Conclusions

The tests undertaken prove the validity and feasibility of implementation of the specified pSLS-handling functions; modelling, translation, information exchange to/from TE functions and admission control at pSLS request epochs

5.3 SLS Invocation Handling Tests

5.3.1 Intra-domain cSLS

5.3.1.1 Overview

In this section we will describe the objectives, controlled/uncontrolled variables, performance metrics and experimentation environment for the performance and stability tests of the intra-domain cSLS Invocation Handling Component with reference to [D3.1]. The functionality of this component, named MTAC, and the details of our implementation are described in [D1.3]. We will also briefly describe the other algorithms we implemented for comparison reasons.

5.3.1.1.1 Objectives

The objective of the performance and stability tests is to assess the performance of MTAC for intra-domain real-time traffic cSLSs under a variety of traffic scenarios and loading conditions and to compare it with the performance achieved by other algorithms in the literature for the same traffic scenarios and loading conditions.

5.3.1.1.2 Controlled/Uncontrolled Variables

The controlled variables are as specified in [D3.1]. The uncontrolled variables, with reference to [D3.1] are the packet loss rate of the l-QC employed for carrying the traffic of the intra-domain real-time cSLSs and the volume and characteristics of the traffic flows.

5.3.1.1.3 Performance Metrics

The performance metric is the trade-off between packet loss rate (PLR) and utilization/cSLS blocking rate achieved by MTAC for intra-domain real-time traffic cSLSs. The primary goal is to guarantee that the requested packet loss rate is achieved and the secondary goal is to maximise the resource utilization/minimise the cSLS blocking rate, subject to the PLR constraints.

5.3.1.1.4 Experimentation Environment

The experimental environment, with reference to [D3.1], is the intra-domain cSLS Invocation Handling software developed by UniS using the Network Simulator (ns-2). The algorithms are implemented in oTCL, which is the interface language of the simulator. The topology is a standard dumbbell topology (see Figure 61). We assume that the sources (cSLSs) connect to the ingress node through links with negligible congestion (zero losses) and that the ingress router first hop link is the bottleneck link.

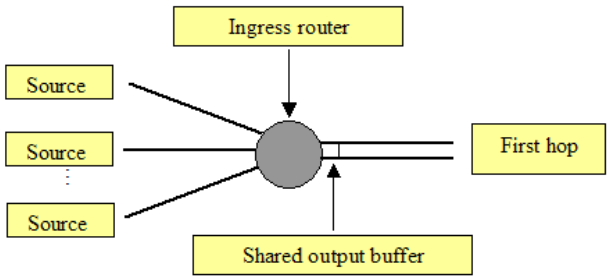


Figure 61: Simulation topology

5.3.1.1.5 Comparison Algorithms

In order to compare the performance of MTAC with other algorithms from the literature we implement two other algorithms.

The first algorithm is a measurement-based admission control scheme –we refer to it as MBAC- described by Zukerman et al in [ZUK] as Rate Envelope Multiplexing (REM), with adaptive weight factor and no histogram update. The reasons for the selection of the specific MBAC scheme for comparison with our scheme are that: (a) REM also makes the zero buffer approximation with respect to statistical multiplexing and (b) implementation-wise, in a similar fashion to our scheme, it requires only aggregate bandwidth measurements and the peak rate of the sources requesting admission in order to derive the admission control decision. The parameters involved in the implementation of MBAC are set to the values used in [ZUK].

The second algorithm is an endpoint admission control scheme –we refer to it as EAC- described by Karlsson et al in [KAR]. In order to test this scheme we implement an additional lower priority queue for the probing packets (out-of-band probing) that can store, as proposed in [KAR], a single probe

packet and which is only served when the higher priority real-time traffic queue is empty. As in [KAR], we set the probing rate equal to the peak rate of the source requesting admission, we consider probe durations of 0.5sec up to 5sec, and we also assume that there is no latency involved between the probing phase completion and the admission control decision.

5.3.1.2 Experiment Setup and Test Description

The algorithms are tested for two target l-QC PLR values: 0.01 and 0.001.

5.3.1.2.1 Topology

The details of the topology used for the intra-domain cSLS Invocation Handling tests are:

- Shared output buffer size: 5 packets
- Bandwidth assigned to l-QC: 3.33Mbps for target l-QC PLR value 0.01 and 3.56Mbps for target l-QC PLR 0.001.

5.3.1.2.2 Simulated Traffic

We consider two types of traffic sources for the simulations:

1. Voice-over-IP (VoIP) traffic sources: VoIP traffic sources are modelled as exponential ON-OFF sources with peak rate 64kbps, average ON duration 0.350sec, average OFF duration 0.650sec and average rate 22.4kbps [HAB].
2. Videoconference traffic sources: Videoconference traffic sources are modelled as H.263 encoded sources with peak rate 332.8kbps and average rate 64kbps [TKN].

The durations of VoIP and Videoconference traffic sources follows two independent exponential distributions with average durations h_{VoIP} and $h_{H.263}$ respectively.

For the simulations we consider the traffic scenarios:

1. VoIP traffic sources only (h_{VoIP} = 300sec)
2. Videoconference traffic sources only ($h_{H.263}$ =300sec)
3. Mixed VoIP and Videoconference traffic sources (h_{VoIP} =300sec and $h_{H.263}$ =180sec)

5.3.1.2.3 Traffic Volume

The invocation processes of both VoIP and Videoconference traffic sources are modelled as two independent Poisson arrival processes with different mean arrival rates l_{VoIP} and $l_{H.263}$ respectively.

The arrival rates are varied in order to produce various traffic loading conditions and examine the behaviour of the algorithms for these loading conditions. For the cases where both VoIP and Videoconference sources are employed (mixed traffic), the averages of their activation rates followed a ratio of 2:1

The value Load=1 corresponds to the average traffic load that that would be incurred by a VoIP source invocation rate equal to 1000 sources/hour. Given the average rates and durations of the VoIP and Videoconference traffic sources, Load=1 for the three simulated traffic scenarios corresponds to:

1. l_{VoIP} =1000 sources/hour (VoIP traffic sources only (h_{VoIP} = 300sec))
2. $l_{H.263}$ =350 sources/hour (Videoconference traffic sources only ($h_{H.263}$ =300sec))
3. l_{VoIP} =270 sources/hour and $l_{H.263}$ =135 sources/hour (Mixed VoIP and Videoconference traffic sources (h_{VoIP} =300sec and $h_{H.263}$ =180sec))

The simulated traffic loading conditions are: 0.5, 1, 2, 3, 4 and 5.

5.3.1.2.4 Algorithms Parameters

For the implementation of MTAC, as described in [D1.3] we use an exponential ON-OFF source with peak rate 64kbps, average ON duration 1.004sec and average OFF duration 1.587sec [CHU] as a *reference source* model and we fix the reference PLR level (e_{ref}) to the value 0.01.

For the implementation of MBAC, as already mentioned, we use the default values of [ZUK].

For the implementation of EAC we try probing durations 0.5sec, 1sec, 2sec, 3sec, 4sec and 5sec. The results that are presented regarding EAC are the ones for the probing duration giving the best trade-off between packet loss and utilization/blocking for each simulated traffic scenario.

5.3.1.3 Test Results

Each simulated scenario is run for 20 different randomly chosen seeds and for 4100sec, using the first 500sec as warming up period.

5.3.1.3.1 Functional Tests

MTAC is functioning properly.

5.3.1.3.2 Performance and Stability Tests

5.3.1.3.2.1 VoIP Sources

5.3.1.3.2.1.1 Target l-QC PLR 0.01

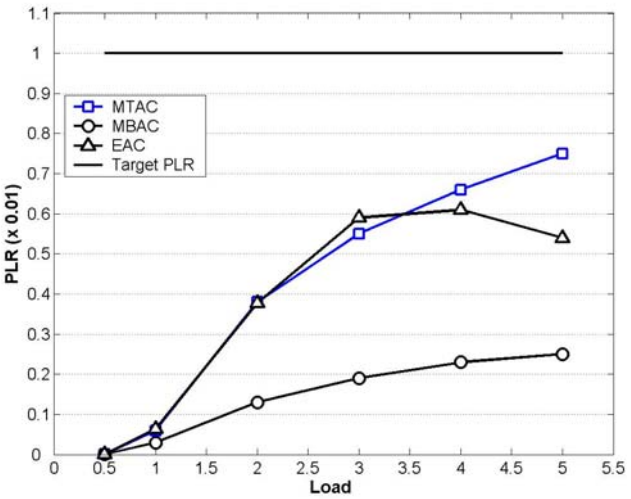


Figure 62: Incurred PLR for VoIP sources and target l-QC PLR 0.01

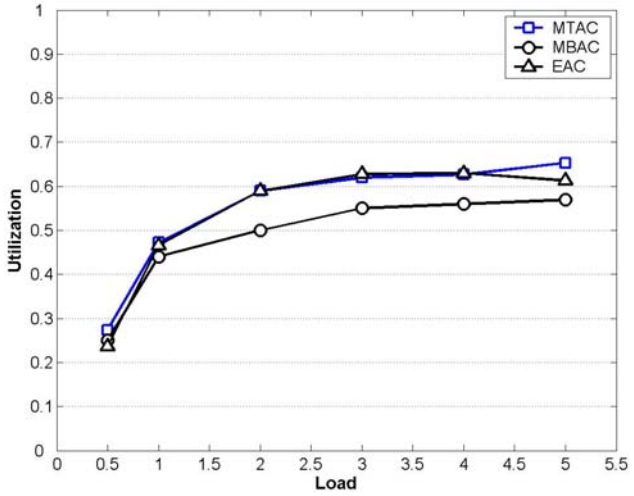


Figure 63: Achieved l-QC utilization for VoIP sources and target l-QC PLR 0.01

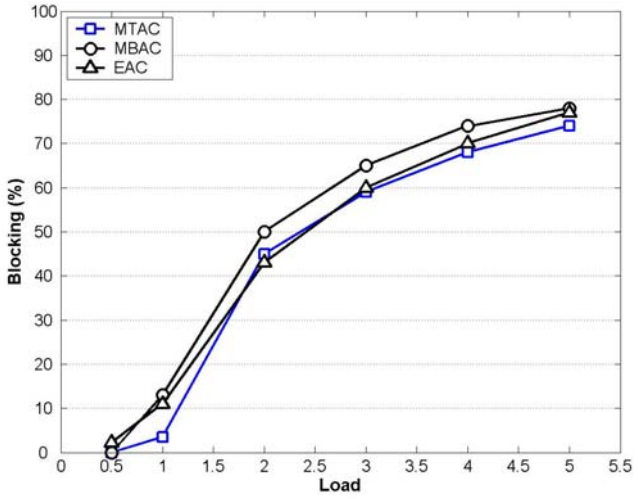


Figure 64: Incurred blocking for VoIP sources and target l-QC PLR 0.01

From the above figures, it can be seen that for VoIP sources and target l-QC PLR 0.01, the PLR achieved by MTAC always stays below the target PLR and, furthermore, MTAC is less conservative than MBAC and EAC, achieving therefore, on average, higher l-QC utilization and a lower blocking rate. For MTAC and MBAC, we observe an increase in the incurred PLR for increasing loading conditions. This is anticipated [GROS] because they both rely on measurements, so every new admission request has the potential of being a wrong decision. This means that a high source invocation rate is expected to have a negative effect on performance. For EAC we observe an increase in the incurred PLR and then a decrease. This happens for increasing loading conditions because simultaneous probing by many sources leads to a situation known as thrashing [BRES]. That is, even though the number of admitted flows is small, the cumulative level of probing packets prevents further admissions, driving therefore the utilization and the PLR to lower values.

5.3.1.3.2.1.2 Target l-QC PLR 0.001

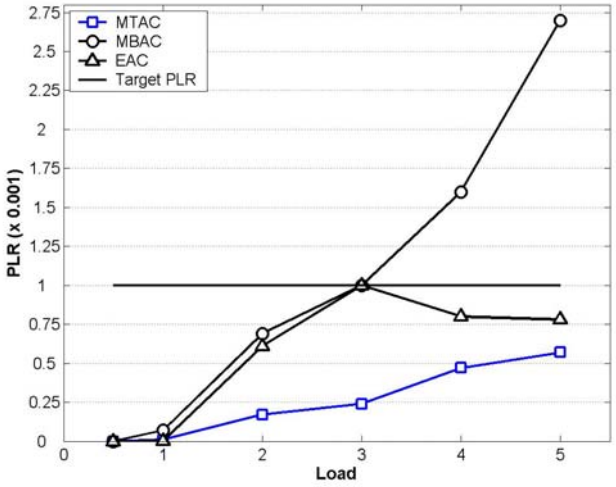


Figure 65: Incurred PLR for VoIP sources and target l-QC PLR 0.001

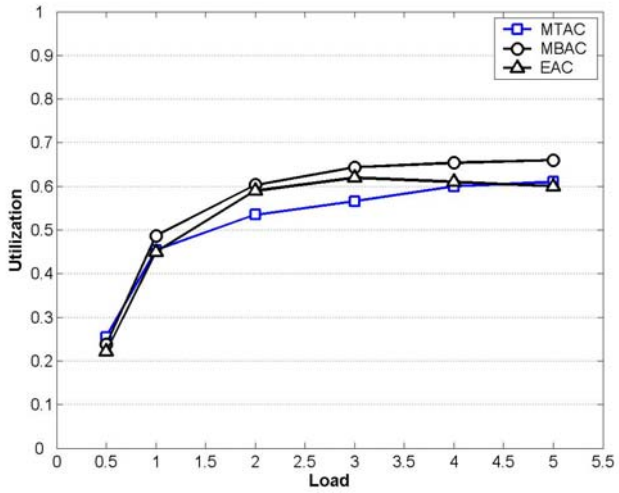


Figure 66: Achieved l-QC utilization for VoIP sources and target l-QC PLR 0.001

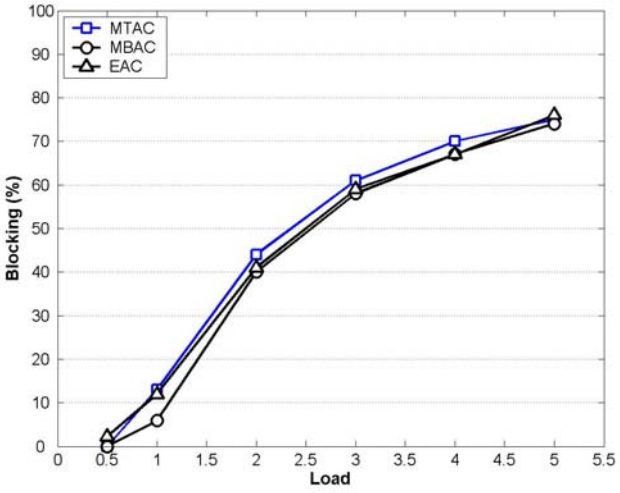


Figure 67: Incurred blocking for VoIP sources and target l-QC PLR 0.001

From the above figures, it can be seen that for VoIP sources and target l-QC PLR 0.001, MTAC is the more conservative of the three algorithms. MBAC violates the target l-QC PLR for increasing loading conditions and by a big margin. That means that in order for MBAC to be able to keep the incurred PLR below the target PLR, its tuning parameters should be reconfigured in an ad-hoc fashion until the desired result is achieved. For EAC we observe a similar thrashing situation as with the previous case.

5.3.1.3.2.2 Videoconference Sources

5.3.1.3.2.2.1 Target l-QC PLR 0.01

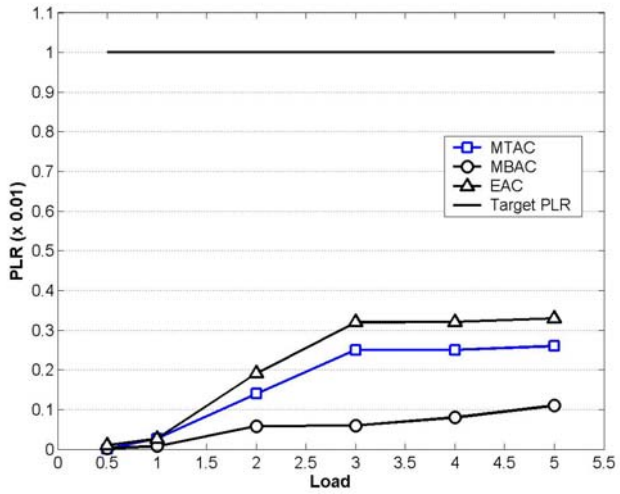


Figure 68: Incurred PLR for Videoconference sources and target l-QC PLR 0.01

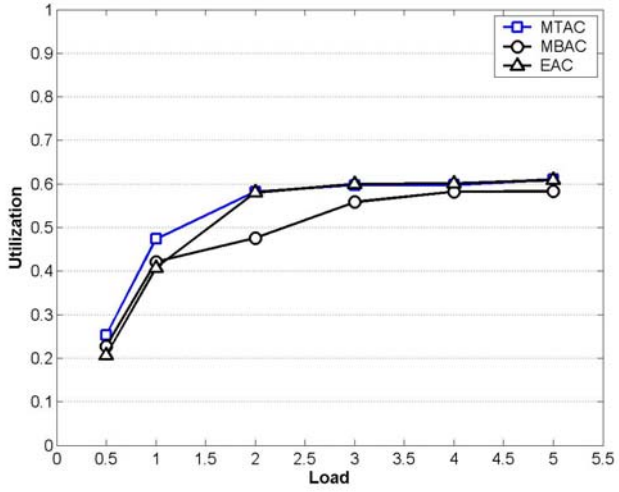


Figure 69: Achieved l-QC utilization for Videoconference sources and target l-QC PLR 0.01

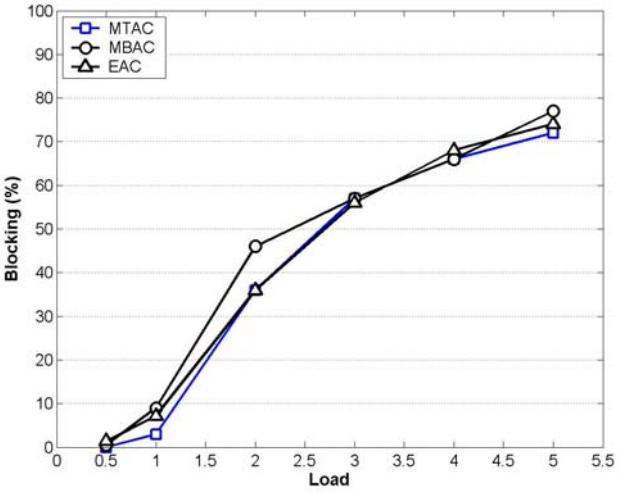


Figure 70: Incurred blocking for Videoconference sources and target l-QC PLR 0.01

For Videoconference sources and target l-QC PLR 0.01, all three algorithms are conservative. This can be partly attributed to the stringent admission control criterion (all algorithms make the worst case assumption that the new source will be transmitting at its peak rate) and the high peak rate of the videoconference sources compared to their average rate.

5.3.1.3.2.2.2 Target l-QC PLR 0.001

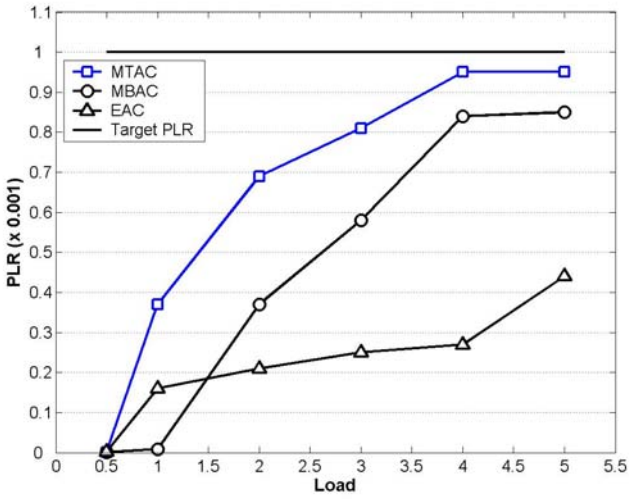


Figure 71: Incurred PLR for Videoconference sources and target l-QC PLR 0.001

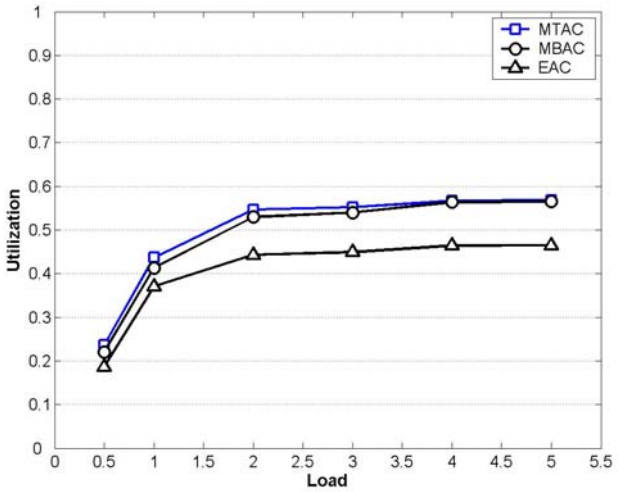


Figure 72: Achieved l-QC utilization for Videoconference sources and target l-QC PLR 0.001

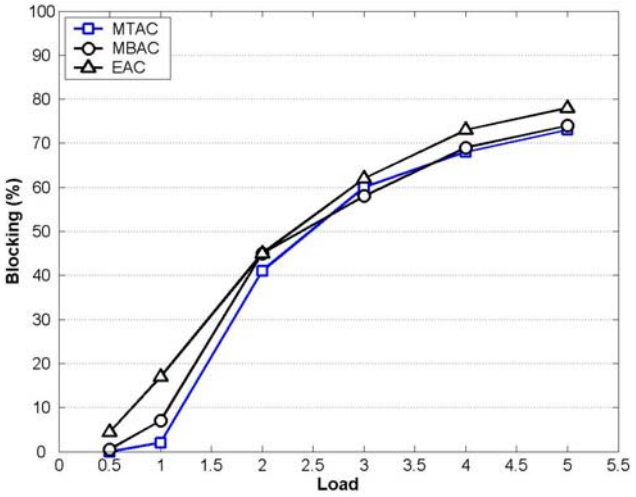


Figure 73: Incurred blocking for Videoconference sources and target l-QC PLR 0.001

For Videoconference sources and target l-QC PLR 0.001, MTAC is the less conservative algorithm achieving therefore better utilization and lower blocking. It needs to be mentioned that the objective of an admission control algorithm is not to achieve the lowest PLR possible, but to keep the achieved PLR within the limits of the target PLR, while maximizing the utilization and minimizing the blocking.

5.3.1.3.2.3 Mixed VoIP and Videoconference Sources

5.3.1.3.2.3.1 Target l-QC PLR 0.01

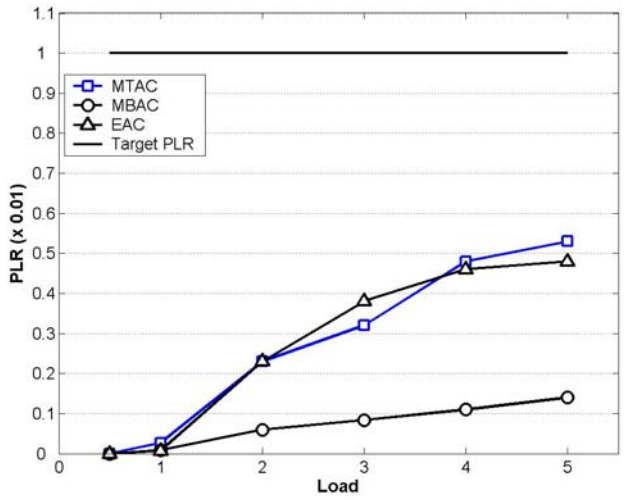


Figure 74: Incurred PLR for mixed VoIP and Videoconference sources and target l-QC PLR 0.01

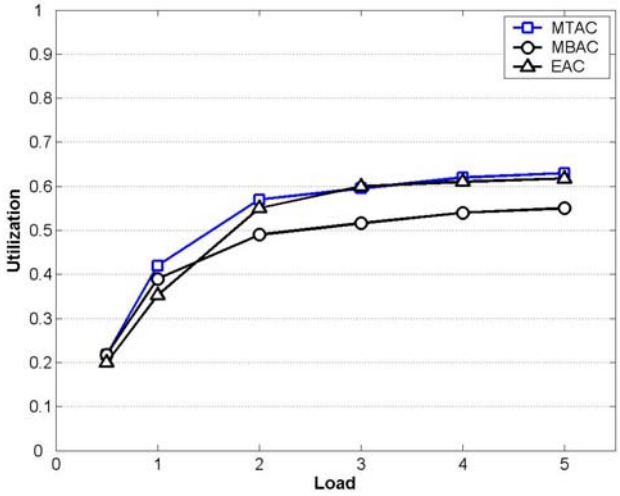


Figure 75: Achieved l-QC utilization for mixed VoIP and Videoconference sources for target l-QC PLR 0.01

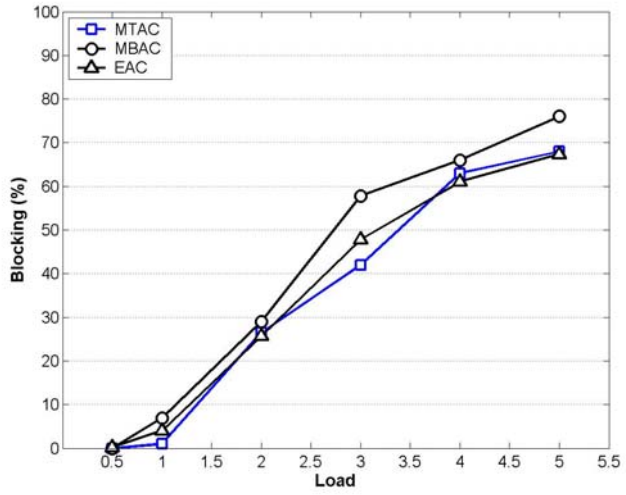


Figure 76: Incurred blocking for mixed VoIP and Videoconference sources for target l-QC PLR 0.01

For mixed traffic, all three algorithms satisfy the target l-QC PLR 0.01. MBAC is more conservative than MTAC and EAC, achieving therefore lower utilization and higher blocking.

5.3.1.3.2.3.2 Target l-QC PLR 0.001

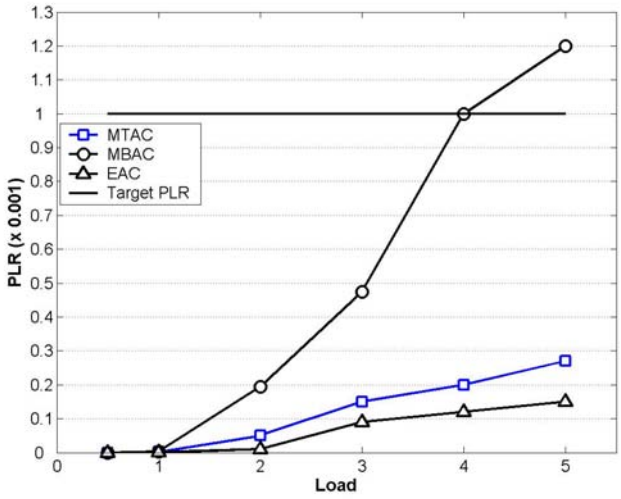


Figure 77: Incurred PLR for mixed VoIP and Videoconference sources and target l-QC PLR 0.001

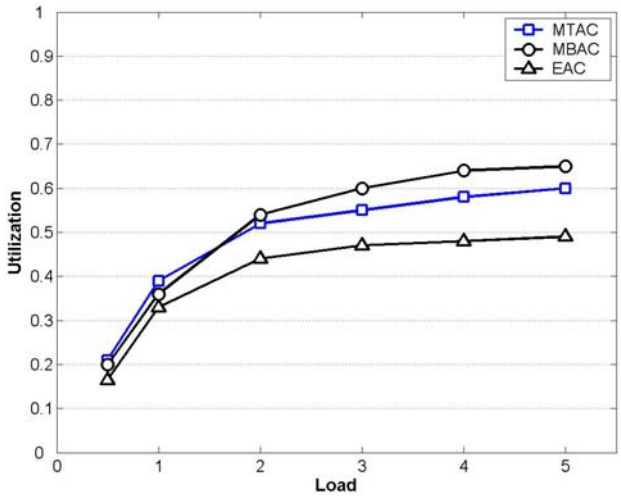


Figure 78: Achieved l-QC utilization for mixed VoIP and Videoconference sources for target l-QC PLR 0.001

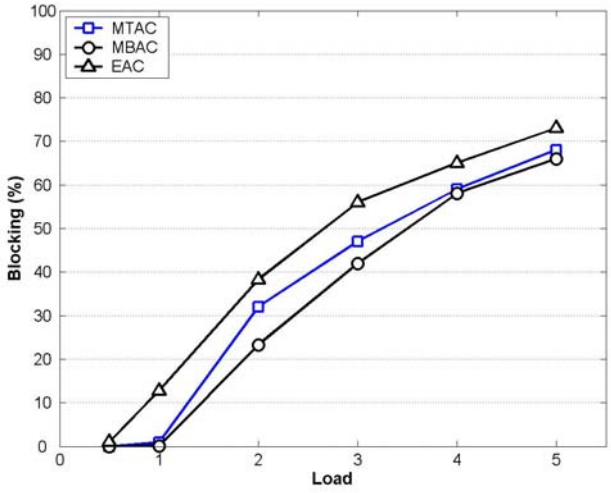


Figure 79: Incurred blocking for mixed VoIP and Videoconference sources for target l-QC PLR 0.001

For target l-QC PLR 0.001, MTAC and EAC achieve this PLR for all loading conditions with MTAC being less conservative, achieving a higher utilization. MBAC violates this PLR for very high loading conditions.

5.3.1.3.2.4 Aggregate Results

Taking into account all the simulated scenarios and traffic loading conditions, the average utilization and average blocking of the three tested algorithms are as follows:

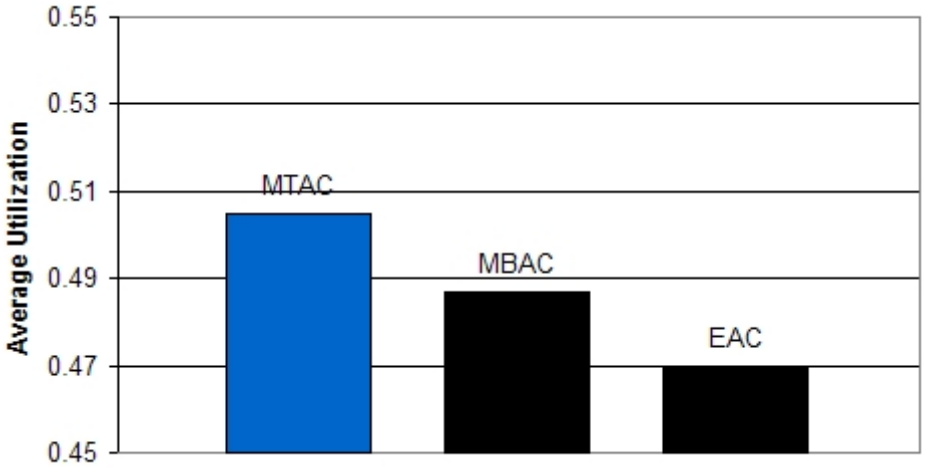


Figure 80: Average l-QC utilization

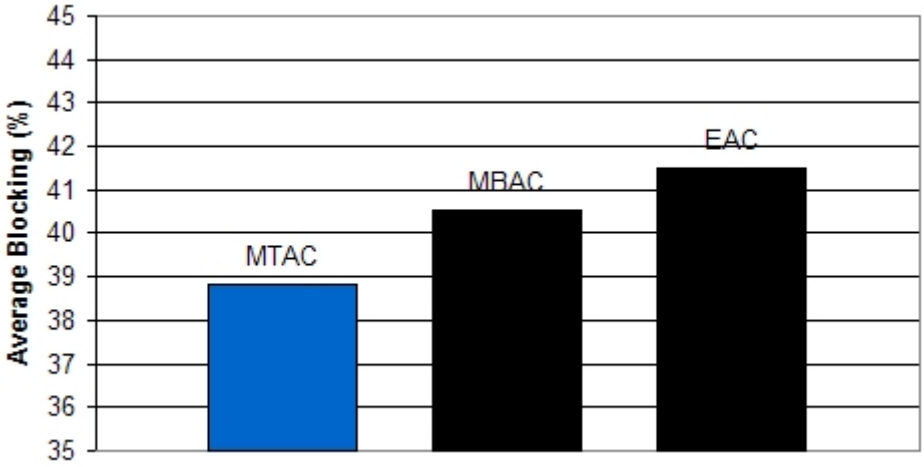


Figure 81: Average cSLS blocking rate

As it can be seen, MTAC achieves the highest average utilization and the lower blocking rate, while, as shown in the figures so far, keeping the incurred PLR below the target l-QC PLR in all simulated scenarios and for all traffic loading conditions. MBAC achieves the second highest average utilization and the second lower blocking rate, but, as already shown, fails to satisfy the target l-QC PLR without further ad-hoc adjustment of its tuning parameters. EAC achieves the worst performance, regarding average utilization and blocking rate, but manages to keep the incurred PLR below the target l-QC PLR. We need to state here though, that the results of EAC presented in the above figures are achieved by trying various values of its tuning parameter (probing period), according to the authors recommendations [KAR], and that for certain values of probing periods -not shown-, are worse than the results for MBAC.

5.3.1.4 Conclusions

The simulation results show that MTAC can perform reasonably well for a variety of traffic scenarios -for both short-range dependent (VoIP) and long range dependent (Videoconference) sources- and loading conditions without requiring any reconfiguration of its parameters and that it compares favourably against other algorithms existing in the literature for the same simulation setup.

While satisfying the target l-QC PLR, MTAC achieves on average 3.6% higher utilization than MBAC and 7.3% higher utilization than EAC. Regarding the cSLS blocking rate, MTAC achieves on average 4.2% lower blocking than MBAC and 6.5% lower blocking than EAC.

5.3.2 Inter-domain cSLS

5.3.2.1 Overview

In this section we will describe the objectives, controlled/uncontrolled variables, performance metrics and experimentation environment for the performance and stability tests of the inter-domain cSLS Invocation Handling Component with reference to [D3.1]. The functionality of this component, named e-MTAC, and the details of our implementation are described in [D1.3].

5.3.2.1.1 Objectives

The objective of the performance and stability tests is to assess the performance of e-MTAC for inter-domain real-time traffic cSLSs under a variety of traffic scenarios and loading conditions. Also in order to demonstrate the performance gains introduced by deploying status information from the inter-domain link, a comparison with the conventional MTAC scheme that does not take into account status information from the inter-domain link will be made. For the conventional MTAC scheme, the inter-

domain real-time traffic cSLSs are treated as in the case of intra-domain real-time traffic cSLSs, with the difference that the minimum available bandwidth (see [D1.3]) is not guaranteed edge-to edge, but end-to-end, taking into account the available inter-domain link capacity. That means that for the conventional MTAC scheme, the bandwidth value that will be used as a threshold for admission control at each ingress node, will be set equal to the minimum between the first-hop link capacity and the part of the inter-domain link capacity that can be logically allocated to the inter-domain real-time traffic cSLSs entering through that ingress node and exiting through that inter-domain link.

5.3.2.1.2 Controlled/Uncontrolled Variables

The controlled variables are as specified in [D3.1]. The uncontrolled variables, with reference to [D3.1] are the packet loss rate of the l-QC employed for carrying the traffic of the inter-domain real-time cSLSs and the volume and characteristics of the traffic flows.

5.3.2.1.3 Performance Metrics

As in section 5.3.1.1.3.

5.3.2.1.4 Experimentation Environment

The experimental environment, with reference to [D3.1], is the inter-domain cSLS Invocation Handling software developed by UniS using the Network Simulator (ns-2). The algorithms are implemented in oTCL, which is the interface language of the simulator. The topology used is shown in Figure 82.

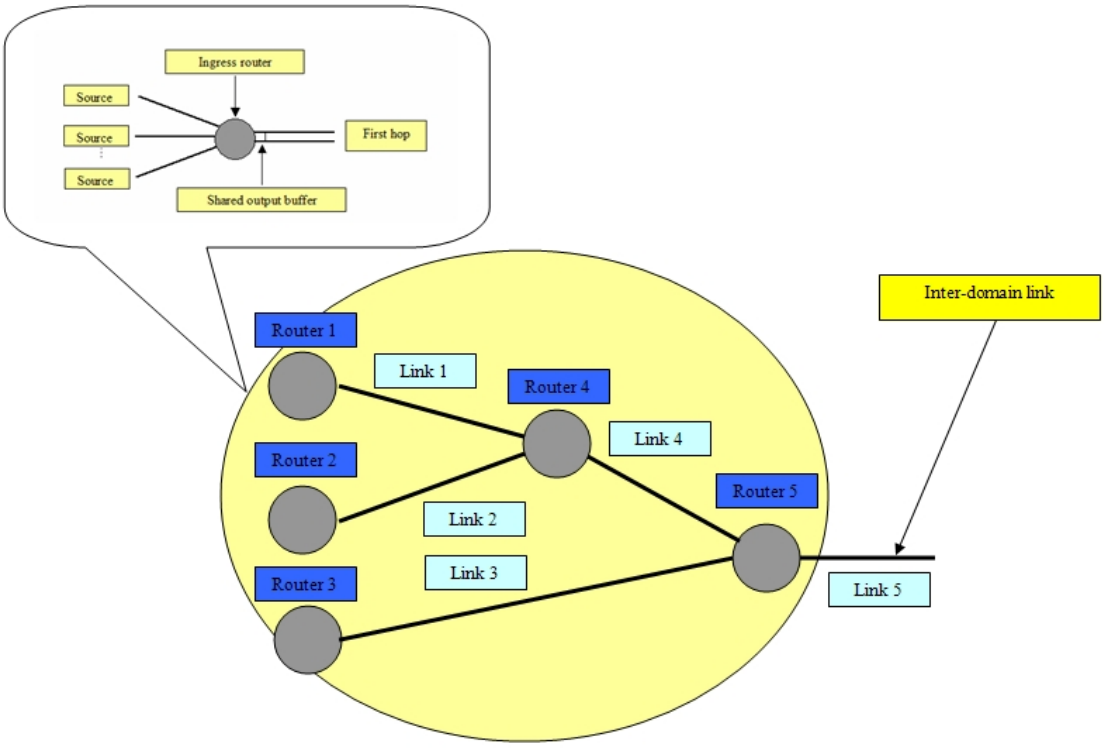


Figure 82: Simulation topology

5.3.2.2 *Experiment Setup and Test Description*

The algorithms are tested for target l-QC PLR value 0.01, setting $PLR_{ingress}$ equal to 0.005 for links 1, 2 and 3 and PLR_{egress} equal to 0.005 for link 5 (inter-domain link) for e-MTAC. For the conventional MTAC scheme, since it does not use any feedback information from the inter-domain link, the allowed target loss rates for links 1, 2 and 3 are set equal to the target l-QC PLR value; that is 0.01.

5.3.2.2.1 Topology

The details of the topology used for the inter-domain cSLS Invocation Handling tests were:

- Shared output buffer size: 5 packets for links 1, 2 and 3 and 2 packets for links 4 and 5.
- Bandwidth assigned to l-QC: 3.33Mbps for links 1, 2 and 3, 6.66Mbps for link 4 (no losses allowed in core network) and 8Mbps for link 5. That means that link 5 is "over-booked" with respect to the aggregate l-QC capacity reserved in links 1, 2 and 3 (it is 80% the aggregate capacities assigned to the l-QC at links 1, 2 and 3). For the conventional MTAC scheme, even though the bandwidth assigned in links 1, 2 and 3 is 3.33Mbps, the capacity that the sources can use in each one of these links is limited to 2.66Mbps so that the aggregate allocated capacity in these three links (3×2.66Mbps) does not exceed the total capacity allocated in link 5 for the l-QC and, therefore, link 5 does not introduce any additional losses to the losses incurred by the first hop links.

5.3.2.2.2 Simulated Traffic

We consider mixed VoIP and Videoconference sources, as described in section 5.3.1.2.2.

5.3.2.2.3 Traffic Volume

In order to examine the performance of e-MTAC for various loading conditions, we simulate loading conditions for links 1 (L1), 2 (L2) and 3(L3): 0.5, 1, 2, 3, 4 and 5 (see section 5.3.1.2.3 for the definition of loading condition). We will refer to this simulated scenario as symmetrical loading

In order to demonstrate the performance gains introduced by deploying status information from the inter-domain link we simulate for e-MTAC and MTAC loading conditions for links 1, 2 and 3 as follows:

1. L1=0.5, L2=1, L3=0.5-1-2-3-4-5 (we fix load 1 to 0.5, load 2 to 1 and we vary load 3 from 0.5 to 5). We will refer to this simulated scenario as asymmetrical loading I.
2. L1=0.5, L2=0.5-1-2-3-4-5, L3=0.5-1-2-3-4-5 (we fix load 1 to 0.5 and we vary load 2 and load 3 from 0.5 to 5). We will refer to this simulated scenario as asymmetrical loading II.

5.3.2.2.4 Algorithms Parameters

For the implementation of e-MTAC, as described in [D1.3] we use an exponential ON-OFF source with peak rate 64kbps, average ON duration 1.004sec and average OFF duration 1.587sec [CHU] as a *reference source* model and we fix the reference PLR level (e_{ref}) to the value 0.01 for the first hop links and to 0.1 for the inter-domain link.

For MTAC, the algorithm parameters are as in section 5.3.1.2.4.

5.3.2.3 Test Results

Each simulated scenario was run for 20 different randomly chosen seeds and for 4100sec, using the first 500sec as warming up period.

5.3.2.3.1 Functional Tests

e-MTAC is functioning properly.

5.3.2.3.2 Performance and Stability Tests

5.3.2.3.2.1 Symmetrical Loading

The total incurred PLR, the achieved inter-domain link utilization and the incurred blocking for e-MTAC as a function of L1, L2 and L3 are shown in the following figures.

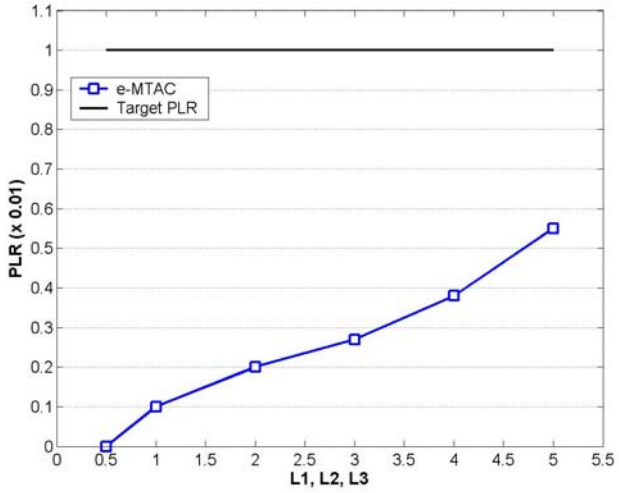


Figure 83: Total incurred PLR

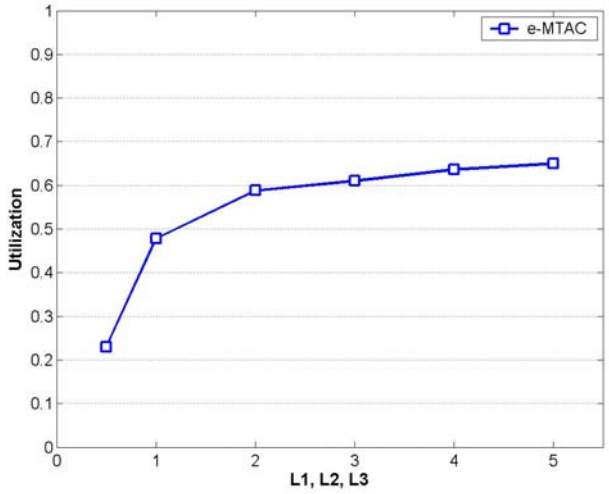


Figure 84: Inter-domain link utilization

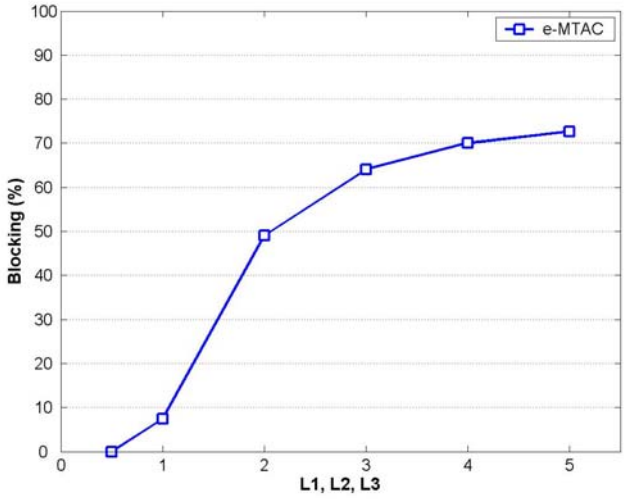


Figure 85: Incurred blocking

As it can be seen, e-MTAC keeps the incurred PLR below the target l-QC PLR for all simulated loading conditions and achieves satisfactory inter-domain link utilization. The performance gains, regarding inter-domain link utilization, by using status information from the inter-domain link, are demonstrated in the following figures where the achieved inter-domain link utilization of e-MTAC is compared to the inter-domain link utilization of MTAC.

5.3.2.3.2.2 Asymmetrical Loading I

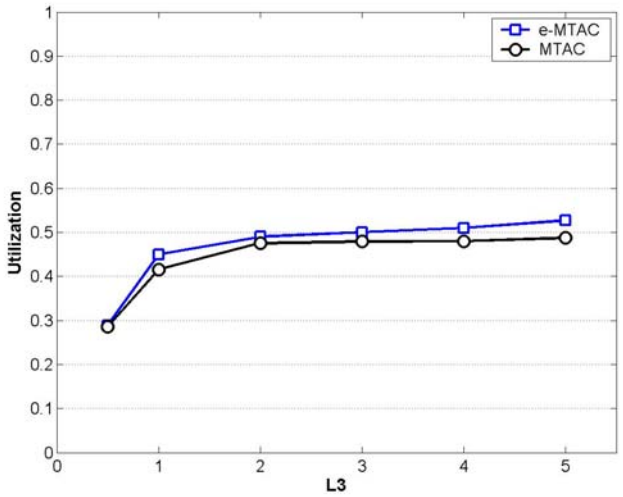


Figure 86: Utilization comparison for the inter-domain link

As it can be seen, e-MTAC achieves slightly better utilization than MTAC. The performance gain from using status information from the inter-domain link is relatively small, because of the low values of load in links 1 and 2 (0.5 and 1 respectively). In this case, because of the low values of load at links 1 and 2, the utilization gain is only a result of the fact that e-MTAC allows more traffic originating from router 3 to be admitted (taking advantage of the low contribution of traffic from routers 1 and 2 at the inter-domain link).

5.3.2.3.2.3 Asymmetrical Loading II

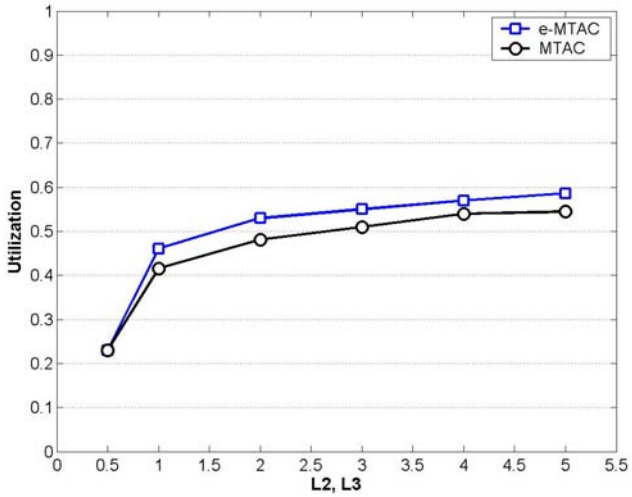


Figure 87: Utilization comparison for the inter-domain link

The utilization gain in this case is higher compared to the previous case because only load at link 1 is fixed to a low value. In this case, e-MTAC allows more traffic from both routers 2 and 3 to be admitted.

It needs to be mentioned that if the inter-domain link capacity was more "over-booked" with respect to the aggregate l-QC capacity reserved in links 1, 2 and 3 (e.g. if it was set to 50% of the aggregate l-QC capacity reserved in links 1, 2 and 3) then the utilization gains would be even higher.

5.3.2.4 Conclusions

The simulations show that e-MTAC can perform reasonably well for the simulated cases. The simulation results also illustrate the inter-domain link utilization gains that are achieved by incorporating status information from the inter-domain link in the admission control scheme.

6 SYSTEM-LEVEL SCALABILITY ANALYSIS

The MESCAL system has been built on a number of design principles and features to contribute to the system's scalability, summarized in the following:

- The MESCAL architecture is designed to function independently in each IP Network Provider (INP) domain. Each INP[2] only interacts with adjacent INPs. The interaction between INPs occurs at the service plane for pSLS ordering and at control plane for receiving/announcing reachability information. No protocols for resource reservation or for initiating the operations and actions of INPs are required.
- Service-layer co-operation between providers is achieved through scalable QoS peering models - namely cascaded and bilateral peering models.
- pSLSs are established off-line between two adjacent domains for transporting traffic at aggregate levels in order to satisfy a large population of users at both customer and provider levels.
- No explicit signalling is propagated at inter-domain level. Any resource reservation request in term of pSLSs are carried out off-line at aggregate level between two providers. Only QoS-based routing information is propagated at inter-domain level through q-BGP, which may optionally carry performance information.
- A number of off-line and independent processes are devised to function including QoS Class (QC) discovery, pSLS ordering, off-line traffic engineering.
- Lightweight dynamic traffic engineering functions are applied at aggregate levels.
- A two-level service admission control scheme is adopted. Subscription negotiation operating off-line, combined with light-weight admission control operating at service invocation instances, mainly relying on local information and coarse local network state indications.

The scalability of a solution/system is the ability for the system to function effectively and keep its performance at desired levels as the value of parameters influencing its behaviour increase. A scalable solution/system should be capable of being deployed at the scale of large networks offering a number of services to a large number of customers. Scalability in QoS-enabled IP networks has a number of distinct aspects at resource and service management levels, including network size, number and granularity of classes of service supported, the extent and complexity of service requests (c/pSLS) to manage, etc.

The pertinent parameters to be taken into account as scalability factors are as follows:

- The extent and complexity of message flow/processing for a new c/pSLS set-up during the c/pSLS negotiation.
- The extent of pSLS set-up per INP for offering inter-domain services
- The number of customer requests (cSLS) to be managed per INP
- The number and granularity of classes of service (QCs) to be offered
- The amount of routing announcements, size of routing tables, etc.

Generally, it is expected that a "no more than linear" dependency to the arrival rate of requests/messages indicates the system is prone to scale.

6.1 Comparison of CADENUS & MESCAL Scalability

In this section, we compare the CADENUS solution with MESCAL approach in terms of message flow in handling a new service request.

[2] In this document the terms INP, ISP, domain, and AS (Autonomous System) are used interchangeably.

The CADENUS architecture uses a business model that takes into account the stakeholders including service and network providers. The scope of the CADENUS business model is broader than MESCAL, with additional stakeholder and roles to be played in providing value-added services. MESCAL is only concerned with QoS-based IP connectivity services. However, the CADENUS architecture does not go into the details of how static and dynamic resource management and traffic engineering is achieved at the network level or how a bi-directional service is constructed, as MESCAL does.

The CADENUS project developed an architecture, which includes functional blocks at the user-provider interface within the service provider domain, and between the service provider and the network provider [CAD-D2.3]. CADENUS defined three key components: *Access Mediator*, *Service Mediator* and *Resource Mediator*. The overall mediation procedure includes the mapping of user-requested QoS to the appropriate network resources, taking into account existing business processes.

Access Mediator: It presents the current service offer to the user. The Access Mediator is responsible for selecting the appropriate service provider, according to the user's request. After authentication, the user requirements are captured, and the Access Mediator sends the information to the service provider who then employs the Service Mediators and Resource Mediators to map the requested and subsequently selected service into the deployed physical network.

Service Mediator: It is responsible for finding and in some cases building from individual elements, the service, and selecting the appropriate Resource Mediator. The Service Mediator is an off-line process, which supervises the incorporation of new services and the management of the physical access to these services via the appropriate underlying network, using the Resource Mediators. It is the task of the Service Mediator to prepare the service level agreements, and subsequently to authenticate the user and map service requests into appropriate network configuration information required by the Resource Mediators.

Resource Mediator: It is associated with the underlying network and its capabilities. There will be one Resource Mediator per administrative domain, and one *Network Controller* for each network technology within that domain. The Resource Mediator receives SLSs from network clients (i.e., Service Mediator). During the negotiation of an SLS spanning multiple domains, a certain number of Resource Mediators- those belonging to the crossed domains – must be involved in the negotiation phase. Each Resource Mediator in the chain is in charge of assuring that part of the service pertaining to its domain. This follows a forward cascaded model where a multi-domain SLS is split into two parts: a single domain SLS plus a remaining part that has to be enforced over one/more downstream domains until the end-to-end path is completed.

It should be noted that the main functionality of the *Network Controller* is intra-domain Admission Control to verify a service arising from an SLS can be accepted without jeopardising the allocated resources.

In CADENUS scalability study [Antonio04], [CAD-D8], the negotiation of a new SLA has been selected, since it exercises the scenario in terms of message-passing and processing, as well as number of entities involved. Figure 88 shows the CADENUS architecture as a cascaded queuing network. In this figure various mediators, together with the Network Controller and service directory modules are modelled as service centres.

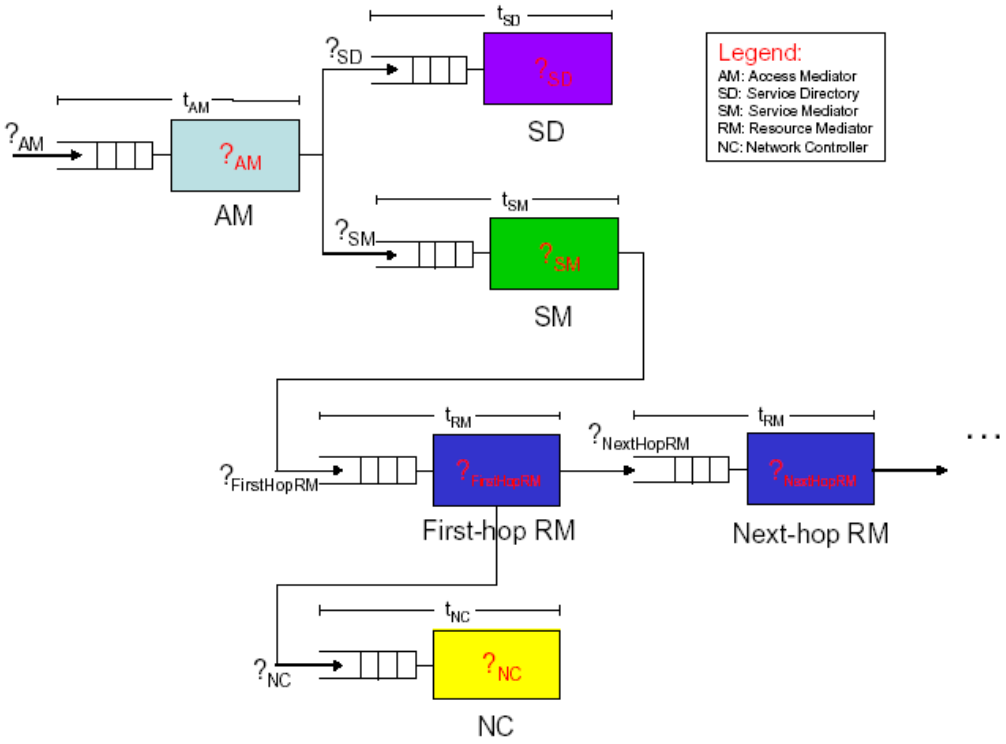


Figure 88: The CADENUS architecture as a queuing network (from CAD-D8).

Figure 89 shows the message flow in the negotiation phase in which an Access Mediator, the Service Directory, a Service Mediator, and all Resource Mediators and Network Controllers in the end-to-end service chain are involved.

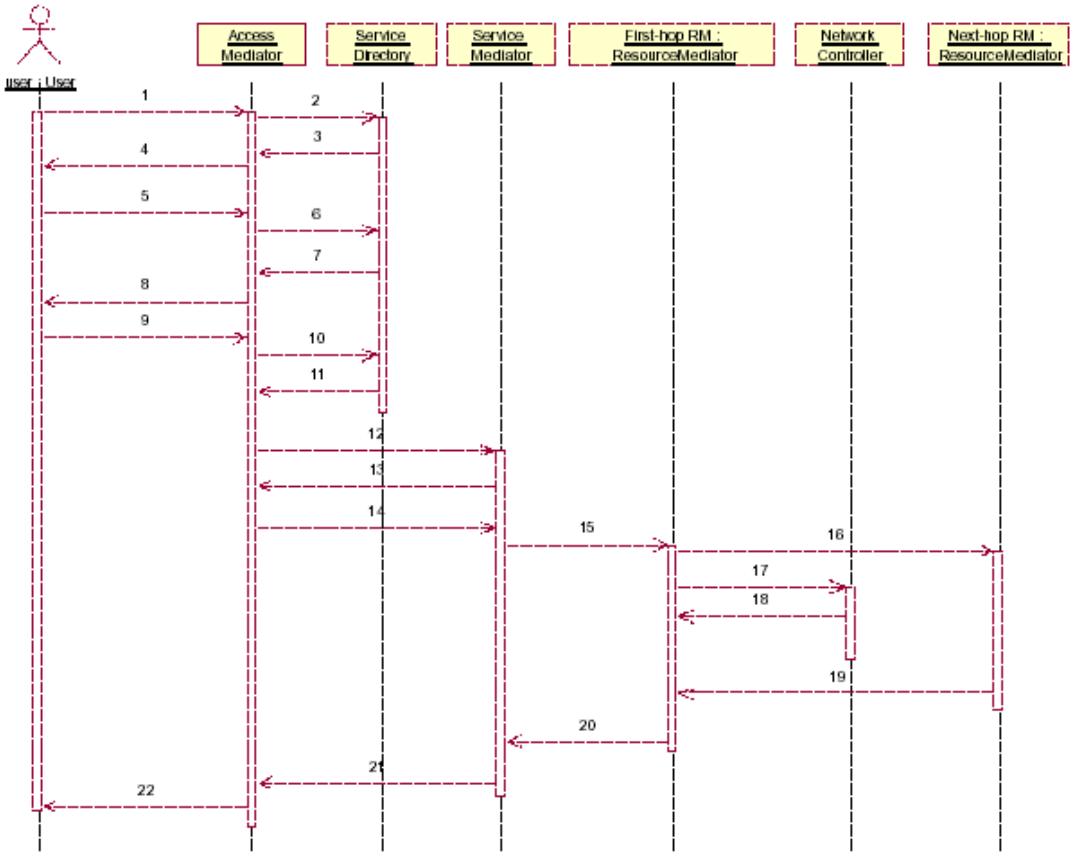


Figure 89: Message flow during service negotiation phase (from CAD-D8).

As shown in Figure 88 and Figure 89, the CADENUS solution involves a significant amount of signalling and processing to deal with a customer's (end-user) request (cSLA). An initial dialogue is performed between the end user and Access Mediator in order to select a service from a list offered by Access Mediator. Following the selection, a new set of parallel dialogues have to be carried out between Access Mediator and one or several candidate Service Mediator(s). Each Service Mediator in turn, has to contact the first Resource Mediator in the chain, (and the latter may have to contact others in the inter-domain chain based on the scope of SLSs), to allow Resource Mediators to make an evaluation of the impact the service is going to have on the network resources at each AS hop and derive a cost to be paid for the enforcement of service. The cSLA subscription operation is further performed by means of an admission control process at intra-domain level (a Network Controller function). This also implies the participation of all the network management entities in the end-to-end chain. Note that some network resources are already pre-reserved in each domain for which a Resource Mediator may reply to a request without performing any network re-configuration/re-dimensioning process. The computed cost is also returned to each Service Mediator by its corresponding Resource Mediator. The Service Mediator(s) return the final results to the Access Mediator, which provides the service list to the end user. Following end user selection and response, only one Service Mediator is eventually chosen and a transaction rollback should be performed by the Access Mediator for those Service Mediator(s) which have not been selected to provide the service.

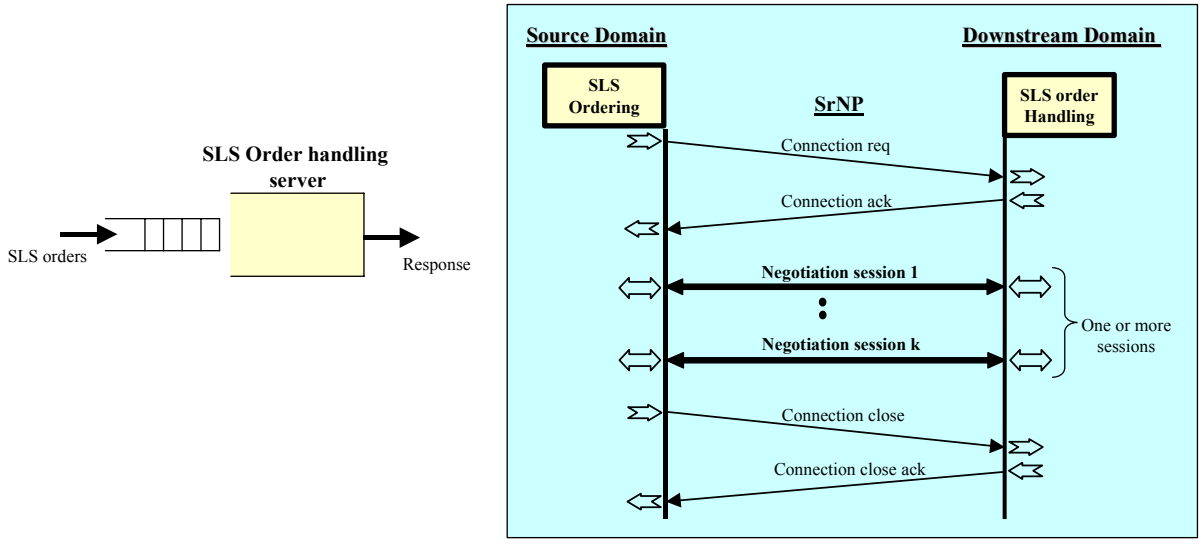
The process of (automated) service definition and service offering by the Service Providers is outside the scope of MESCAL. MESCAL focuses on the business relationships between customers and INPs and between INPs in order to provide QoS across multiple domains driven by agreed SLSs. As such, the primarily concern of MESCAL work is QoS-based IP connectivity services. QoS-based IP connectivity services offered by INPs are divided into elementary and complex connectivity services. Elementary connectivity services are strictly point-to-point and uni-directional. In this section, only elementary connectivity services are considered.

For a provider domain wishing to provide QCs from its domain to destinations outside its domain, a number of *QC-operations* are performed off-line to build end-to-end *QC*s. The *QC-advertisement* operation enables a provider domain to inform other providers of its QoS-class capabilities (*l-QC*s, *e-QC*s, or *m-QCs)*. The *QC-discovery* operation enables a provider to find the QCs offered by other provider domains. Following QC-discovery, the *QC-mapping* operation enables a provider domain to either build *e-QC*s by determining suitable combinations of the domain's own capabilities (*l-QC*s) with the QC capabilities offered by other downstream provider domains or to map its *l-QCs* to *m-QCs*. The *QC-binding* operation enables a provider domain to decide which of the possible QoS-mappings will be used for actually negotiating corresponding *pSLSs* with appropriate downstream providers. The *QC-implementation* operation enables a provider domain to implement a QoS-binding at the network (IP) layer.

In MESCAL, prior to offering any connectivity service to its customers, an INP creates the logical infrastructure in order to provide these services across multiple domains. Therefore, a provider domain independently performs QC-operations and resource provisioning for implementing *e-QC*/*m-QC* in advance. This type of off-line process is repeated recursively to enable other upstream domains to offer QoS-based services. In each step of the cascade, the upstream provider acts in the consumer role to the provider immediately downstream. It is each provider's responsibility to make appropriate *pSLSs* with the immediate downstream provider making it possible for individual customer IP QoS services to be created and managed along the entire route. The pSLSs are established between two adjacent INPs for exchanging traffic in the Internet, with the purpose of expanding the geographical span of their offered services. pSLSs support aggregated traffic (i.e. serving many customers), and it is assumed that they are in place prior to any cSLS agreements with end customers or pSLS agreements upstream peers. After the appropriate pSLSs are put in place, an INP can accept service requests (c/pSLSs) and offer connectivity services to its customers or peers.

In the cascaded model, each INP makes *pSLS* contracts with the adjacent interconnected INPs but not with providers more than "one hop away". A service dialogue is only performed between the end user and INP or between two neighbouring INPs in order to establish a connectivity service. Figure 90 shows the MESCAL model in servicing the SLS requests and the message flow in the negotiation

phase between two adjacent domains, i.e., SLS ordering of a source domain and SLS order handling of downstream domain.



a): SLS orders queue and order handling server **b): Message flow during SLS negotiation phase.**

Figure 90: MESCAL model in SLS negotiation.

Therefore, acceptance/rejection of any service requests at subscription epoch and admission control decisions at invocation epoch are performed locally at the INP level. There is no need to contact other INPs in the inter-domain chain to fulfil a service request. This approach creates a scalable solution in terms of message flow because it avoids propagation of the service request to downstream INPs in forward direction towards the end-to-end chain for every requested service.

The complicated set of signalling/message flow transactions used in CADENUS is as follows:

- $\{User\}_K \rightarrow \{Access\ Mediator\} \rightarrow \{Service\ Mediator\}_L \rightarrow \{Resource\ Mediator\}_N$ where K, L, N≥1
- Pre-allocation/rollback actions in different Resource Mediator(s).

While, the MESCAL design philosophy, as briefly explained above and specified in more detail in [D1.1], [D1.2], avoids the complicated set of signalling/message flow transactions used in CADENUS:

- $\{Customer/Provider: SLS\ Ordering\}_K \rightarrow \{SLS\ order\ Handling\ @\ INP2\}$ where $K \geq 1$

6.2 Scalability of Inter-Provider Peering Models

A number of QoS peering models can be used for the interconnection and service-layer interactions between providers' for offering QoS services across multiple domains. The type of inter-domain peering impacts the service negotiation procedures, the required signalling protocols, the path discovery through QoS binding, and path selection. Any solution for QoS peering should function effectively and in a scalable manner. Mescal studied three peering models (source-based, cascaded, and bilateral) that are explained briefly below.

In the ***source-based model***[3], an IP Network Provider (INP) negotiates *pSLSs* directly with a number of downstream providers to construct an end-to-end QoS service. With this model, service peers are not necessarily BGP peers. The source point requires an up-to-date topology of the Internet to discover domains to negotiate with and to select end-to-end routes. In addition it needs to know the domains' advertised *l-QCs* in order to perform mapping and binding of these *l-QCs* to form *e-QC*s. The source INP directly establishes *pSLSs* with a set of potential domains (neighbour, transit, and distant ASs) in

[3] Source-based model is referred to as Centralised model in D1.4.

order to reach a set of destinations and offer an end-to-end QoS-based service. Although it is possible to find and set-up optimal routes to the destinations since the source point has access to the overall QoS-based topology, the need for accurate topological and QoS related information of the Internet is a major drawback of this model. It may be feasible for a relatively small number of domains, but it raises scalability concerns when a large number of networks are involved. The source INP will end up with many *pSLSs* to manage.

In the ***cascaded model***, an INP only negotiates *pSLSs* with its immediate neighbouring provider/s to construct an end-to-end QoS service. Thus, the QoS peering agreements are between adjacent neighbours, but not between providers more than "one hop away". There is no need for complete topology related information, except routing information. This type of peering agreement provides the QoS connectivity from a customer to reachable destinations that may be several domains away. Setting-up *pSLSs* with defined scope and distinct performance characteristics between adjacent INPs is the compelling feature of this model. For QoS-Class discovery and selection, each INP in the chain needs to know its adjacent neighbours and the status of related interconnection links. In addition, each INP needs to know the *e-QC*s advertised by its neighbouring domains for binding with its own *l-QC*s in order to implement its own *e-QC*s, which may subsequently be advertised to its customers and upstream domains. This is true for every INP involved in the chain in order to implement its *e-QC*s. Each INP has only a limited number of *pSLSs* to manage (see next section) making the cascaded model more scalable than source-based model.

The ***bilateral model*** relies on the cascaded model and the use of the *m-QC* concept. Setting-up *pSLSs* with open scope (i.e., no explicit reachability information) and no distinct performance characteristics but simple compliance with well-known m-QC behaviours between adjacent INPs is the compelling feature of this model. In this model, there is no end-to-end QoS guarantees defined and consequently there is no need to build *e-QC*s, which are the fundamental differences between this model and cascaded model. The bilateral model does not provide any end-to-end bandwidth guarantees because it enables any destination to be reached, without prior explicit indication in the *pSLS*. Each domain is engineered to support a number of local QoS classes (i.e. *l-QC*s). These *l-QC*s are mapped to globally well-known *m-QC*s. Each AS advertises the *m-QC*s that it supports in its administrative domain. Other domains can make *pSLS* arrangement in cascaded fashion with this domain to make use of offered *m-QC*s. Although, inter-domain routing is *pSLS* constrained, each domain can find out whether it can reach certain destinations in an *m-QC* plane through a BGP-like protocol (q-BGP) [Bouca05]. The basic requirement for a domain is to have one pSLS agreement with its adjacent domain to join an m-QC plane. This makes the bi-lateral model even more scalable than the other models.

6.3 The Extent of pSLS Set-up

Here, we study the scalability of pSLSs to manage when a specific QoS peering model is employed. Two different connectivity topologies are considered: a star topology (Figure 91) and a multi-tiered hierarchy topology (Figure 92).

In a simple star topology, the number of pSLSs to establish for all three models is an order of $O(N_d)$ as

$N_p = N_{qc} * (N_d - 1)$ where:

N_{qc} = Number of QCs offered by AS1 to its customers to reach customers in AS2, AS3, or ASn (specified as a constant value).

N_d = Number of domains (INPs)

N_p = Number of *pSLSs* from a source domain (i.e., AS1) to reach all domains for all e-QCs in either source-based or cascaded models. N_p is regarded as the number of *pSLSs* to join all m-QC planes in bilateral model.

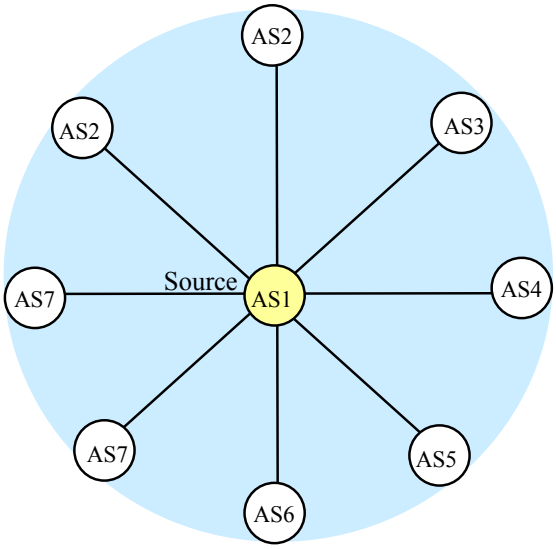


Figure 91: Star topology for connectivity.

A three-tiered hierarchy topology shows a model of the global Internet, organised as a collection of independently operated networks, shown in Figure 92. Here, we are mainly concerned about the customer-provider relationship.

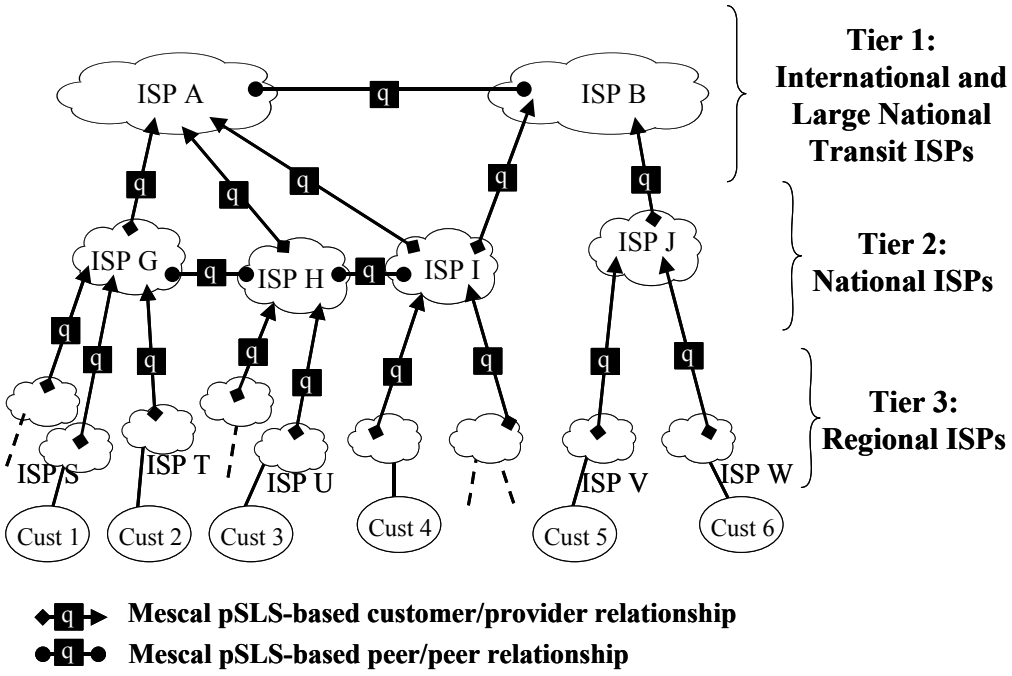


Figure 92: Three-Tier Internet model pSLS-based agreements.

Considering the above connectivity, the number of pSLSs to establish for all three models is as follows. It should be noted that:

N_p = Number of *pSLSs* from a source domain to reach all domains for all e-QCs in either source-based or cascaded models. N_p is regarded as the number of *pSLSs* to join all m-QC planes in bilateral model.

N_{qc} = Number of QCs (*e-QC* or *m-QC*) offered by a source INP (ISP-S) to its customers to reach customers in ISP-T, ISP-U, or ISP-V. Here, the number of peering points (K) shared between two domains is set to one. Hence, N_{qc} is regarded as a constant value that is studied in more detail in the next section.

Source-based Model: $N_p = N_{qc} * \left[\frac{N_d * (N_d - 1)}{2} \right]$, in the order of $O(N_d^2)$

Cascaded model: $N_p = N_{qc} * (N_d - 1)$, in the order of $O(N_d)$

Bilateral model: $N_p = N_{qc}$

The bilateral scalability figure (as in cascaded model) depends on the number of peering points (K) a domain share with its adjacent domains. With a single peering point, a domain can join the *m-QC* based parallel Internet. With more peering points, more pSLSs can be established for the same *m-QC* to improve the reachability, resiliency, etc. Figure 93 shows the scalability of different peering models in terms of pSLS set-up, where N_{qc} and K are set to one.

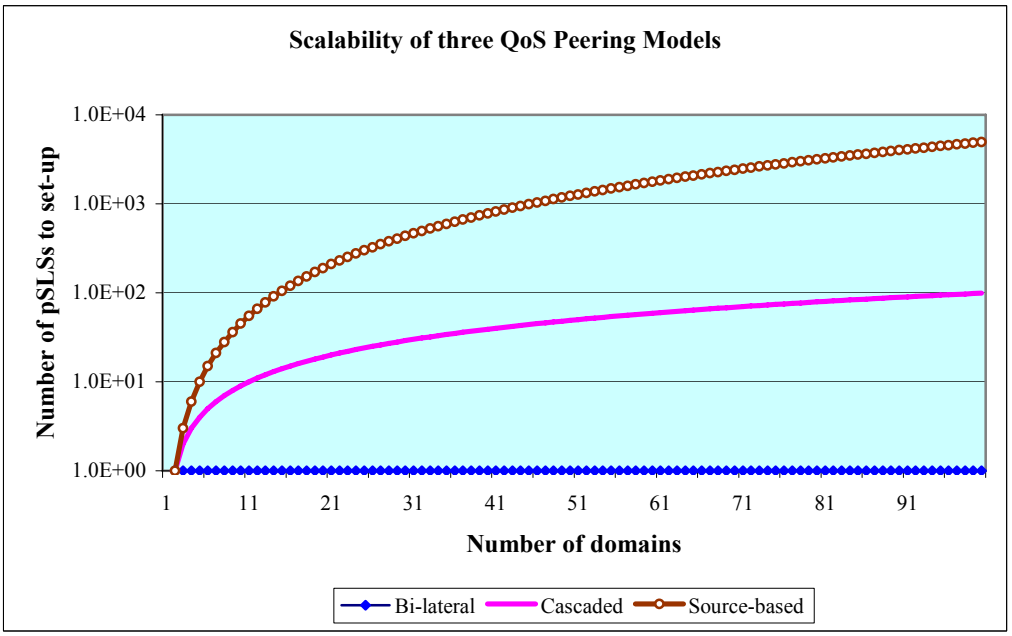


Figure 93: The trend of pSLS set-up in each peering model.

6.4 Number and Granularity of QCs

In MESCAL, three Solution Options are proposed to support a diversity of services. These service options are aimed at customers requiring different performance levels; Loose Guarantee (LG) for mass market with better-than-best-effort service levels, Hard Guarantee (HG) for a small number of value-added services with guaranteed bandwidth and performance and Statistical Guarantee (SG) for a range of services between the two extremes. The SG solution option is implemented using cascaded model while LG and HG solution options are implemented using bilateral model.

The scalability figures, shown in the previous section, are dependent on the number of QCs offered. Due to the definition of the LG and HG solution options, only a limited number of well-known *m-QCs* are used globally, whereas in SG the QC binding is more flexible, increasing the number of offered QCs and increasing the number of pSLSs to set-up.

In SG, the use of the DiffServ DSCP to distinguish QCs means that the maximum number of offered QCs for a given destination can be no more than the maximum allowable number of DSCPs, i.e. 64 in the IPv4 realm. In the worst case, the number of combinations for offering a single QC is $\{64 * N_c\}$ where 'N_c' is the number of peer ASs, and thus $\{(64^2) * N_c\}$ for offering all the possible QCs. If two ASs have on average K peering points then the worst-case scalability factor for supporting and offering the maximum number of QCs becomes:

$$N_{qc} = \{(64^2) * N_c * K\}$$

We can see that the SG scales linearly with the number of peer ASs and the number of peering points for each peer AS.

The proposed Solution Options also introduce an increase to the number of BGP announcements proportional to the number of QCs supported. In LG, this is limited to the number *m-QC* planes and in SG, where q-BGP is optionally used, it depends on the number of e-QCs offered between the two peer domains having local significance.

In addition, the number of *m-QCs* in LG and *e-QCs* in SG have direct impact on the size of routing tables, but the size increase will depend on the level of address aggregation.

The scalability of HG as it uses MPLS-TE could be a concern. In order to establish a full mesh logical network, an order of $O(N_e^2)$ unidirectional LSP tunnels needs to be established where 'N_e' is the number of edge nodes, which can be very large. In fact, it is even worse than $O(N_e^2)$ since multiple LSP paths are used for load sharing and different QoS services may use different LSPs between two node pairs. HG is scalable and feasible only if a limited number of LSPs are to be established. Therefore, HG has scalability concerns for large deployments in terms of end-to-end inter-domain tunnel set-up but as stated above HG Solution Option is not aimed at the mass-market deployment but at specialised services where its scalability is less of a concern.

6.5 Summary

This study of the scalability of MESCAL has

- identified the key scalability attributes of the MESCAL solution.
- compared the message flows associated with a service request in both the CADENUS and MESCAL architectures and shows how the cascaded approach in the MESCAL solution leads to a significant reduction in the amount of inter-domain signalling relative to the CADENUS approach.
- analyses various peering models (source, cascaded and bilateral) and shows that the models used in the MESCAL solution (cascaded and bilateral) are scalable in terms of the number of pSLSs required for large networks.
- shown that the three solution options (Loose, Statistical and Hard Guarantee) have scalability in terms of QoS Classes that is matched to their intended usage i.e. the mass deployment option (Loose) has excellent scalability, the Statistical option scales linearly while the Hard Guarantee option does not scale well for large deployments, but it is targeted at specialised high value services and therefore will not be mass deployed.

7 CONCLUSIONS

7.1 Overview

The overall MESCAL solution for Internet QoS provisioning can be summarised as follows: provider domains determine QoS routes by means of q-BGP, which is activated between adjacent provider domains following establishment of peering agreements, pSLSs, to exchange aggregate QoS traffic. MESCAL has proposed three instances of solutions, which we have called solution options, for delivering the type of QoS required across the spectrum of services/applications, namely loose, statistical and hard QoS guarantees.

This deliverable has presented the tests undertaken to assess the validity and performance of the MESCAL functionality for inter-domain QoS delivery and has described the yielded results. In addition to the scalability results obtained through experimentation. further aspects of the scalability of the MESCAL solution have been analysed at a theoretical level.

Overall, based on the yielded experimentation results, the following conclusions can be drawn:

- Prototype implementation and the tests undertaken in the testbed prove the overall validity of the MESCAL solution for delivering QoS across multiple domains and the proposed algorithms, schemes and protocols; and show that the specified functionality can be feasibly implemented.
- Simulation results indicate that better performing routes for carrying QoS traffic can be established through the proposed solution compared to classical BGP. The specified intra- and inter-domain traffic engineering and c/pSLS admission control algorithms are of reasonable performance, yielding favourable results compared to ad-hoc configurations or alternative schemes.
- Theoretical analysis shows that the proposed solutions, for providing QoS with loose, hard or statistical guarantees in the Internet, scale in terms of the QoS-classes to be used/offered.
- By virtue of its design, the MESCAL inter-domain QoS solution is inherently scalable and realistic; it relies on interactions between adjacencies, it does not rely on per flow end-to-end bandwidth reservations and QoS signalling means.

Finally, it should be noted that the MESCAL solution for providing QoS in the Internet supplements the current best-effort Internet and does not distort current business relationships between providers.

The following summarise the main points and conclusions drawn from experimentation for each of the major functional aspects of the MESCAL solution.

7.2 Implementation of the MESCAL Solution

MESCAL implemented a testbed of experimental Linux-based routers for ‘proving the concept’ of the specified QoS solution in a realistic network environment. The primary purpose of the testbed was to assess the feasibility of the enhancements and interactions required at the IP level for employing q-BGP as the inter-domain routing protocol between adjacent ASs and the computation of QoS-constrained paths.

Within the limitations on the number of emulated domains in the testbed, *it was proved that the specified q-BGP protocol achieved its functional objectives and the MESCAL solution for inter-domain QoS delivery is valid and feasible*. In particular:

- QoS-aware inter-domain routes can be constructed by using q-BGP; different routes, in terms of AS paths and end-to-end QoS performance, are determined per QoS-class and the traffic following a QoS route receives the appropriate QoS-class treatment within each domain.
- The specification of the q-BGP protocol and the associated QoS-aware route selection process can lead to feasible implementations.

- It is feasible to implement the necessary data plane interactions (DSCP swapping, q-FIB, etc.) to provide end-to-end QoS as specified.
- q-BGP can interoperate with classical BGP, due to the use of capability negotiation procedure.
- No significant impairment of the performance of q-BGP was observed, compared to classical BGP, as a result of the addition of QoS-related information.
- q-BGP is able to select distinct routes per QoS class;
- q-BGP reacts to the change of QoS class configuration that can occur on the AS chain.

Furthermore, through testbed tests it was *proved the validity and feasibility of the specified PCS-based approach for building inter-domain MPLS LSPs was proven* – for providing hard QoS guarantees. In particular:

- It is possible to compute inter-domain QoS constrained paths.
- The specification of the communication protocol between PCSs (PCP) is a basis for feasible implementation of the protocol.
- It is feasible to integrate the operation of q-BGP with the PCS to discover QoS-aware path candidates.
- The speed of path computation indicates that the PCS-based approach is a viable solution for larger networks.
- The activation of the solution option to provide hard QoS guarantees does not impact the size of inter-domain routing tables - one entry per QoS-class is required.

7.3 Scalability of the MESCAL Solution

The scalability analysis has shown that *the MESCAL solution is scalable*. The design of the solution approach is based on a set of principles that contribute to overall scalability. In particular:

- Provider domains interact only with adjacent domains with distinct interfaces at the service and IP planes.
- Inter-domain service layer interactions are achieved through cascaded and bilateral QoS peering arrangements, which are shown to be scalable in terms of the number of pSLSs required for large networks.
- pSLSs address aggregate traffic flows between domains; per-flow end-to-end QoS signalling and bandwidth reservations for QoS are not required.
- As QoS routes are constrained between those provider-domains, that have established pSLSs, providers have increased levels of control in handling the volumes of QoS traffic transported through their domains.
- QoS peering is simplified through the use of Meta-QoS-Classes, reducing the complexity of QoS bindings between adjacent domains, and making the inclusion of QoS-related information in BGP scalable at the scale of the Internet.
- The MESCAL solution requires the set-up of marking/remarking mechanisms at the domains' edges and the employment of QoS-aware routing, resulting in expansion of the routing table space; both these aspects, scale with the number of pSLSs, which in turn scales with the number of QoS-classes and provider domains participating in the QoS solution.
- The three solution options proposed by MESCAL to address the diversity of customer requirements for QoS services have scalability properties that are appropriate to their expected deployment i.e. mass market solutions relying on loose QoS are highly scalable, solutions for more specialized QoS needs are less scalable.

7.4 q-BGP

In addition to proving the validity and feasibility of q-BGP through testbed implementation, simulations have been conducted for assessing and getting insight into its behaviour and impact on network performance in larger Internet-like topologies. The conclusions of this work can be summarised as follows:

Injecting QoS information into BGP when coupled with a QoS-aware route selection process can result in better performing routes across meta-QoS-class planes compared to classical BGP. However, poor selection of q-BGP policy parameters may degrade performance compared to standard BGP.

It has been demonstrated that, in addition to any service differentiation implemented by utilising different PHBs/packet forwarding priorities within the routers of each AS, the application of appropriate route selection policies on advertised QoS attributes can also deliver QoS differentiation.

While the quantity of q-BGP messages is greater than for plain BGP the results indicate that when scaled to Internet-sized AS topologies there is only a three-fold increase in q-BGP-update messages.

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 broader selection criteria.

In summary, the addition of administratively set QoS attributes to BGP can be achieved in a scalable and incremental manner that maintains the scalability of classical BGP while yielding better performance in terms of end-to-end delivered QoS. This maintains a certain level of predictability for INPs while improving performance.

7.5 Off-line TE

For realizing the off-line decision-making processes required by the MESCAL solution for dimensioning provider domains with the appropriate amount of intra- and inter-domain resources, suitable traffic engineering algorithms for uni- and multi-cast traffic have been developed and tested in computer/simulation environments. Experimentation showed that the *developed algorithms achieved their functional objectives with reasonable performance, yielding favourable results when compared – where possible - to ad-hoc configurations and alternative schemes*. In particular, simulation results show that:

Inter-domain TE

- The specified genetic-based algorithm for off-line inter-domain traffic engineering can find near-optimum solutions/allocations for accommodating QoS-sensitive traffic demands assigning them to intra-domain resources (represented by l-QCs) and inter-domain resources (represented by pSLSs).
- The genetic-based algorithm outperforms ad-hoc random-based assignment approaches in the sense that the determined bindings (combinations of l-QCs and pSLSs) result in significant lower cost (represented by the sum of the pSLS cost, intra-domain TE cost and inter-domain link utilisation); it has been shown that under simplified conditions the cost of the genetic algorithm is close to a theoretical lower bound cost.
- The algorithm scales with the number of traffic demands, l-QCs and pSLSs, yielding processing times in the order of minutes to hours, which is acceptable for its prescribed time-scale of operation (in the order of days to weeks).

Intra-domain TE

- DSCP-aware routing can successfully be used for providing individual routing to different traffic classes. Bandwidth- as well as hop-count-optimised routing per QoS class can be configured to run in parallel on the same physical network, using the proposed link weight optimisation techniques.

- The load is balanced more evenly across the network than with standard all-class shortest path routing on inverse capacity link weights.
- The approach is scalable to large ASs with more than 500 nodes.

Inter- and intra-domain TE

- An integrated approach to off-line TE results in lower cost TE solutions with lower total consumed bandwidth than a decoupled approach in which inter-domain and intra-domain solutions are treated separately.

Multicast TE

- The specified scheme for intra- and inter-domain multicast TE scheme, which is based on a genetic algorithm approach, can produce effective resource optimisation solutions with constrained bandwidth capacity.
- The algorithm outperforms alternative, shortest-path-based, schemes, resulting in improved network design -savings in consumed intra-domain bandwidth and also balanced inter-domain link utilisation- as well as in reduced blocking rate for group join requests.
- The proposed TE scheme scales with the number of multicast groups and network nodes, yielding processing times in the order of minutes, which is acceptable for off-line TE computations.
- By applying per l-QC trees (engineered through the aforementioned scheme) for each QoS class per group, fairness problems amongst QoS-classes that appear in DiffServ aware multicast can be avoided.

7.6 c/pSLS Management

The pSLS-centric interactions between providers for negotiating pSLSs as well as the interfaces between the service handling and the TE functions within a provider domain, required by the MESCAL solution, have been developed and tested for their validity and performance. The results prove that the *process of pSLS establishment between providers can be feasibly realised in a highly-comprehensive manner, hiding underlying complexity; also, that the specified pSLS handling functions can safely and efficiently –with increased levels of automation and flexibility- realise the decisions and/or provide the input required to the inter-domain TE functions, off-line and q-BGP*. In particular, it was shown that:

- It is possible to describe in a highly-abstracted form the essential aspects of pSLSs, as appropriate to the type of QoS exchange and the underlying business relationships between providers, hiding underlying complexity and realisation details.
- The concepts and notions required by the operation of the pSLS-aware service layer functions of the MESCAL QoS solution are consistent and can lead to implementation; on pSLS establishment, it is feasible and scalable to derive all information required by the inter-domain TE functions - traffic matrix and q-BGP configuration information.
- It is feasible to carry out pSLS ordering and negotiations in an automated fashion, facilitating therefore the process of pSLS establishment between providers; it is possible to fully automate even the logic of negotiating pSLSs, proving the validity of the proposed ordering and negotiation framework.
- On pSLS request epochs, admission control may be exerted for managing the trade-off between long-term performance of the engineered domain and accepted subscriptions/contracts; the proposed algorithm is of polynomial complexity with respect to the number of established pSLSs.
- The developed cSLS invocation handling algorithms perform reasonably well for a variety of traffic scenarios, satisfying the target packet loss rate while achieving satisfactory resource utilization; in addition, inter-domain link utilization gains can be achieved by utilising inter-domain link status information in the admission control scheme.

8 REFERENCES

[Akam99] Akamai, "Internet Bottlenecks: the Case for Edge Delivery Services," Akamai White Paper, 1999.

[Antonio04] S.D'Antonio, M.Esposito, S.P.Romano, G.Ventre, "Assessing the scalability of component-based frameworks: the CADENUS case study", ACM SIGMETRICS Performance Evaluation Review, Special Issue of E-commerce, Volume 32, Issue 3, pp. 34-43, December 2004.

[Bouca05] M.Boucadair Ed., "QoS-Enhanced Border Gateway Protocol", Internet Draft, <draft-boucadair-qos-bgp-spec-00.txt>, June 2005.

[BRES] L.Breslau, E.Knightly, S.Shenker, I.Stoica and Z.Zhang "Endpoint Admission Control: Architectural Issues and Performance", SIGCOMM 2000, Stockholm 2000.

[Brite] http://www.cs.bu.edu/BRITE

[bu02] T. Bu and D. Towsley, "On Distinguishing between Internet Power Law Topology Generators", IEEE Infocom 2002

[CADENUS] European IST (Information Society Technologies) research project, for more on IST-CADENUS visit: http://wwwcadenus.fokus.fraunhofer.de/old_entry.html .

[CAD-D2.3] S.P.Romano, ed., "Resource Management in SLA Networks", D2.3 CADENUS Deliverable, May 2003.

[CAD-D8] S.P.Romano, ed., "Cadenus Scalability Analysis", D8 CADENUS Deliverable, June 2003.

[CHU] C.Chuah, L.Subramarian and R.Katz "Furies: A Scalable Framework for Traffic Policing and Admission Control", May 2001, U.C Berkeley Technical Report No. UCB/CSD-01-1144.

[D1.1] P.Flegkas, et al., D1.1 MESCAL Deliverable, "Specification of Business Models and a Functional Architecture for Inter-domain QoS Delivery", http://www.mescal.org/, June 2003.

[D1.2] M.Howarth, et al., D1.2 MESCAL Deliverable, "Initial specification of protocols and algorithms for inter-domain SLS management and traffic engineering for QoS-based IP service delivery and their test requirements", http://www.mescal.org/, January 2004.

[D1.3] N.Wang, et al., D1.3 MESCAL Deliverable, "Final specification of protocols and algorithms for inter-domain SLS management and traffic engineering for QoS-based IP service delivery", http://www.mescal.org/, June 2005.

[D3.1] E.Mykoniati, et al., D3.1 MESCAL Deliverable, "Specification of test campaigns and experimentation plans for performance analysis and prototype validation", http://www.mescal.org/, September 2004.

[fal99] M. Faloutsos, P. Faloutsos and C. Faloutsos, "On Power-Law Relationships of the Internet Topology", SIGCOMM 1999.

[Feam03] N.Feamster, J.Borkenhagen and J.Rexford, "Guidelines for Interdomain traffic engineering," *ACM Computer Communications Review*, Oct 2003.

[Fortz00] Fortz, B., and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," paper presented at INFOCOM 2000. *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*. IEEE, Vol.2, 2000.

[For02] B.Fortz and M.Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE Journal on Selected Areas in Communications*, Vol. 20, pp.756-767, 2002.

[GROS] M.Grossglauser and D.Tse "A Framework for Robust Measurement-Based Admission Control", IEEE/ACM Transactions on Networking, June 1999.

[HAB] I.Habib and T.Saadawi "Multimedia Traffic Characteristics in Broadband Networks", IEEE Communications Magazine, July 1992.

[KAR] V.Elek, G.Karlsson and R.Ronngren "Admission Control based on End-to End Measurements", IEEE INFOCOM 2000.

[TKN] http://www-tkn.ee.tu-berlin.de/research/trace/trace.html.

[Wang99] Y.Wang and Z.Wang, "Explicit Routing Algorithms for Internet Traffic Engineering," *Proceedings IEEE ICCCN* 1999, pp. 582-588.

[Xiao00] X.Xiao et al., "Traffic Engineering with MPLS in the Internet," *IEEE Network*, Vol. 14, No. 2, pp. 28-33, 2000.

[ZUK] T.Lee and M.Zukerman, "Admission Control for Bursty Multimedia Traffic", IEEE INFOCOM 2001.