AGAVE

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

IST-027609

D3.1: Initial Specification of Mechanisms, Algorithms and Protocols for Engineering the Parallel Internets

Document Identifier: AGAVE/WP3/UniS/D3.1/final/public	
Deliverable Type: Report **Deliverable Nature:** Public	**Contractual Date:** 30 November, 2006 **Actual Date:** 14 December, 2006

Editor:	Ning Wang (UniS)
Authors:	*TID:* M. L. García Osma, A. J. Elizondo, J. Rodríguez Sánchez *FTR&D:* M. Boucadair, B. Decraene, B. Lemoine, J.L. Le Roux *Algo:* E. Mykoniati, P. Georgatsos *UCL.uk:* D. Griffin, J. Spencer; J. Griem *UniS:* N. Wang, M. Amin, K. H. Ho, M. Howarth, G. Pavlou *UCL.be:* B. Quoitin, O. Bonaventure

Abstract: This deliverable is a result of WP3 of the AGAVE project. It provides the initial specifications of the proposed algorithms and mechanisms for implementing Network Planes (NPs) and also binding NPs to form Parallel Internets (PIs) that offer end-to-end service differentiation. Lightweight routing mechanisms are proposed both intra- and inter-domain, including Multi-path Routing with Dynamic Variance (MRDV), Multi-topology routing, overlay routing, IP tunnelling and QoS-enhanced BGP (q-BGP). In addition, MPLS fast rerouting, BGP based egress point selection and BGP planned maintenance are proposed for implementing Network Planes that offer high QoS availability and also for traffic optimisation purposes in case of network failures. Service level and network level monitoring mechanisms are also introduced in this document.

Keywords: Quality of Service (QoS), Traffic engineering, Network Planes, Parallel Internets, VoIP, MRDV, Multi-topology routing, Overlay routing, IP tunnelling, q-BGP, Resilience, Network monitoring

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

Information Society and Media

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

Executive Summary

This deliverable is a result of AC3.1, AC3.2 and AC3.3 activities members of WP3 work package. Hereafter, we provide the list of WP3 objectives:

- Specify mechanisms, algorithms and protocols for the realisation of Network Planes (NPs);
- Specify mechanisms, algorithms and protocols for the enhancement of inter-domain routing mechanisms to realise Parallel Internets (PIs);
- Specify an overall engineering approach using simulation and testbed approaches, and to specify the components realising the algorithms and protocols for the Parallel Internets;
- Select appropriate implementation methodologies, technologies and environments, for both simulations and testbed;
- Design and implement the components realising the Parallel Internets, through customisation of tools and existing components and development of new components as appropriate;
- Specify test objectives and requirements for evaluating the validity of the proposed specifications.

This document specifies the algorithms and mechanisms for implementing Network Planes within individual IP Network Providers' (INPs) domains, and also for binding NPs across multiple domains for end-to-end service differentiation purposes.

IP routing algorithms and mechanisms are proposed as the one of the building block for implementing NPs. In this document we introduce three lightweight intra-domain routing mechanisms, namely Multi-Paths Routing with Dynamic Variance (MRDV), Multi-topology routing and INP-level overlay routing, through which NPs can be realised. With sophisticated resource provisioning on top of these routing mechanisms, edge-to-edge service differentiation can be achieved within individual INP's domains.

In order to enable end-to-end service differentiation across multiple INPs' domains, inter-domain routing mechanisms are also proposed for horizontally binding individual NPs in different INPs. In this document QoS-Enhanced BGP (q-BGP) and IP tunnelling based approaches are introduced for this purpose. Resilience requirements are also addressed for (1) services that need high QoS availability and (2) robust traffic engineering in NP-aware domains. Specifically, description on MPLS fast rerouting and BGP based egress point selection algorithms are presented in this document in case of network failures.

Finally service engineering mechanisms implemented by Service Providers (SPs) are introduced, mainly on service level monitoring mechanisms for assuring QoS availability that has been contracted with customers.

The validation and evaluation on the proposed algorithms/mechanisms will be done in WP4.

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

1.1 Overview of WP3

WP3*, Parallel Internets Engineering*, undertakes the specification, design and implementation of appropriate mechanisms, algorithms and protocols for realising the Network Planes (NPs) and their interconnection over the Internet in order to provide end-to-end Quality of Services (QoS). There are three activities involved in WP3, namely: Network Plane realisation and engineering, Inter-domain routing and Design and implementation.

AC3.1 Network Plane Realisation and Engineering is responsible for specifying mechanisms, algorithms and protocols for engineering Network Planes within a single administrative domain. Appropriate mechanisms/protocols have been investigated for implementing Network Planes, such as Multi-Topology Routing (MTR), MPLS, overlay routing, and MRDV [CALL05]. In addition, service engineering at the service provider level is also addressed, and relevant work items mainly include SLS monitoring.

AC3.2 Inter-domain Routing is responsible for specifying mechanisms, algorithms and protocols for end-to-end QoS delivery across multiple domains. Both standard BGP and its QoS enhancement (q-BGP [BOUC05]) are being investigated as the underlying platform for inter-domain routing for QoS, resilience and TE purposes. In addition, IP tunnelling and INP-level inter-domain overlay routing mechanisms and algorithms are designed and specified in this activity.

AC3.3 Design and Implementation undertakes the implementation of the Network Planes and Parallel Internets components specified in AC3.1 and AC3.2. Testbeds and simulation tools will be selected and customised. Suitable open-source software and proprietary software owned by partners will be reviewed and selected where appropriate. The components required to realise the Parallel Internets will be designed and implemented. Enhancements to brought-in software and integration into the simulators will be also undertaken.

1.2 Structure of this document

This document is structured as follows:

- Section 2, *Design overview* gives a top-level description on the design of Network Planes and Parallel Internets proposed in the AGAVE project.
- Section 3, *Network Plane engineering* specifies the proposed algorithms and mechanisms for engineering Network Planes within individual INPs. Specifically, Multi-Path Routing with Dynamic Variance (MRDV), Multi-topology routing, INP-level overlay routing are introduced as lightweight approaches to implement Network Planes. In addition, Network Plane monitoring and NP resilience/fast reroute issues are also discussed in this section.
- Section 4, *Network Plane binding* specifies the proposed algorithms and mechanisms for horizontally binding Network Planes from INPs in order to form end-to-end Parallel Internets. IP tunnelling and q-BGP are specified as the NP binding mechanisms. Inter-domain resilience issues are also addressed on both MPLS and BGP domains.
- Section 5, *Service engineering* describes Service Provider level QoS provisioning on top of the physical network owned by INPs. Specifically, service level monitoring mechanisms are introduced for SLS assurance with end users.
- Section 6, *Summary* provides a brief summary of this document.

2 DESIGN OVERVIEW

2.1 About Network Planes and Parallel Internets

As it is mentioned in [D1.1] and [D2.1], Network Planes are designed in the AGAVE project for the purpose of service differentiation within individual INPs' infrastructure, and also they can be horizontally bound to form Parallel Internets (PIs) in order to enable end-to-end service differentiation. It should be noted that, the concept of "service differentiation" in the AGAVE project not only refer to the traditional QoS differentiation in terms of delay, packet loss ratio etc., but also include the availability of the offered IP connectivity service. Given this generalised QoS requirements to be satisfied by Network Planes, our proposed approach is not only to consider the traditional DiffServ-like forwarding differentiation, but more importantly, to associate several dimensions including routing differentiation and resource management in a lightweight fashion. We will both propose new routing paradigms, and also investigate how legacy routing technologies can be used as the underlying control plane mechanism for provisioning different level of QoS. Based on the specific service requirements, and form a routing perspective, INPs may choose one ore multiple routing procedures for implementing the NPs within their own networks (e.g. multi-topology, distinct route selection process, activation failure detection means like Bi-directional Forwarding Detection, etc.). Furthermore, individual NPs may also bind their NPs to form PIs through homogeneous or even heterogeneous routing protocols/mechanisms that are used to implement NPs in their individual networks so as to provide consistent IP transfer treatment as requested by some Service Providers (implicitly by services and/or applications).

2.2 Focus on routing

Figure 1 shows the routing techniques that will be considered in the AGAVE project for the implementation of Network Planes. These routing techniques are positioned in a spectrum according to their capabilities for delivering service differentiation. At the left end of the spectrum, all the routing techniques support the default Best-Effort (BE) service as well as enhanced QoS without any committed guarantees, which is known as *Better-Than-Best-Effort (BTBE) service*. The common practice for providing BTBE service is to apply QoS monitoring technologies in order to perform path selections according to the measured QoS performance. As the mechanisms such as overlay routing and IP tunnelling do not need to control the entire physical path but only a small number of "isolated" network elements within a network or even across multiple INPs, they can be regarded as the most lightweight solution to QoS provisioning. A subset of the routing technologies shown in the figure may also offer statistical QoS guarantees, meaning that the QoS requirements can be statistically fulfilled based on proper resource dimensioning on top of these routing paradigms. A typical example is to intelligently compute the MTR IGP weight for each link within the network and optimally assign traffic demands to individual routing topologies according to the specified QoS requirements. Finally, MPLS base QoS solutions are able to offer strict QoS guarantees thanks to the capabilities in explicit path selection and resource reservation along the selected paths. Based on the above description, it is not difficult to imagine the trade-off between the strictness of QoS guarantees and the complexity: the stricter the required QoS guarantees, the higher the introduced complexity. In the AGAVE project, we consider the entire spectrum of the service (mainly QoS and availability) requirements, from BTBE to hard guarantees, as well as the corresponding tools for resource management, depending on the QoS and availability requirements of individual Network Planes. Finally, it is worth mentioning that, the mapping between these routing technologies and Network Planes is not on a one-to-one basis, but in a very flexible way. For example, each of the protocols that have multi-topology capabilities, such as q-BGP, MT-IGP and MRDV, can be either used to implement one distinct Network Plane or used as a common routing platform that serves multiple NPs.

The intra- and inter-domain routing mechanisms shown in Figure 1 can be bound to form Parallel Internets that offers end-to-end Service Differentiation (see the arrows). For instance, intra-domain Label Switched Paths (LSPs) can be optimally stitched/nested together in order to offer hard guarantees across the premium Network Planes engineered by individual INPs. Another example is that dedicated routing topologies can be allocated to a joined MT-IGP and q-BGP routing infrastructure, both of which are multi-topology aware, for end-to-end service differentiation purposes. In addition to routing differentiation, the realisation of Network Planes may involve other mechanisms, e.g., DiffServ where appropriate. All these technologies related to the engineering of Network Planes will be addressed in this work package. The selection of the appropriate technologies for implementing/binding NPs according to specific service requirements is done by the functional block of Network Plane Design & Creation in the AGAVE reference architecture.

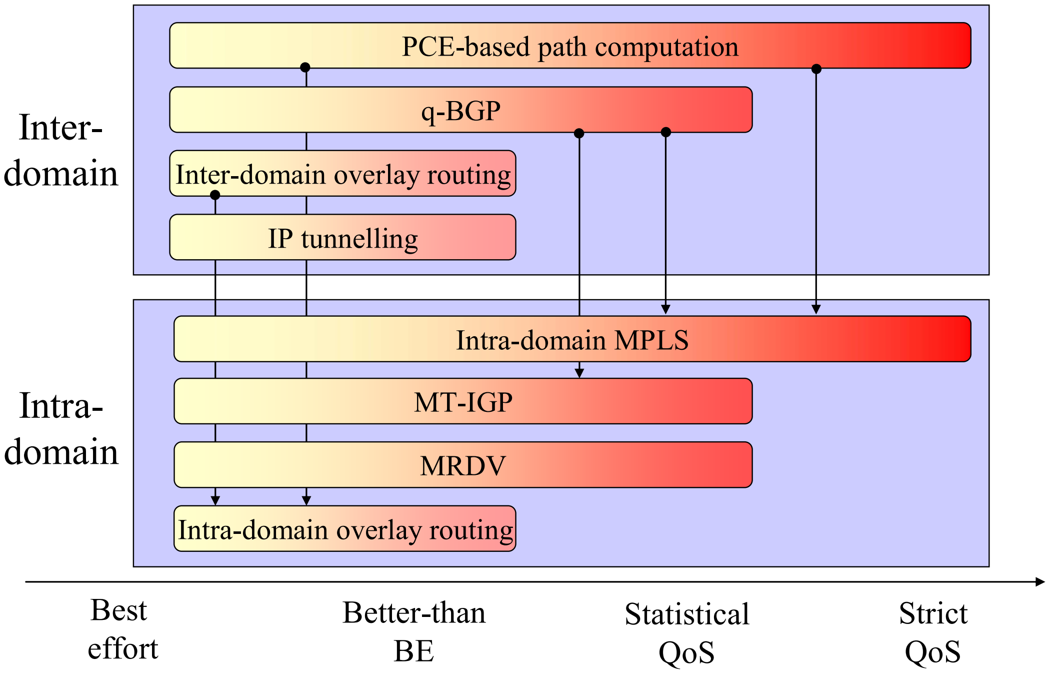


Figure 1 Routing spectrum for QoS support

2.3 NP robustness and resilience

As it is mentioned above, availability is an important aspect to be considered in the design of Network Planes. For those Network Planes that are designed to carry highly available services, dedicated recovery mechanisms are needed to cope with network failures. In the AGAVE project, we will investigate how recovery requirements by some NPs can be satisfied through specific (re-)routing techniques. Specifically, MPLS fast re-routing and BGP planned maintenance will be considered with the objective to minimise the service unavailability caused by the failure of ASBRs. For example, a typical requirement for VoIP media flows is a Loss of Connectivity (LoC) of less than 200ms, while currently the BGP convergence time after link failures can be up to 100 seconds. In this case, some fast rerouting techniques are needed in order to reduce the gap for the purpose of ensuring the availability required by VoIP applications. In addition to recovery mentioned above, traffic-oriented resilience is also needed for traffic optimization purposes. Even for those Network Planes that do not have high requirement on recovery (e.g. need fast rerouting to achieve a bounded LoC time), traffic re-distribution after network failures should not cause congestion during the period when the link is down. Towards this end, resilience-enabled traffic engineering techniques will be designed with the objective to guarantee that traffic distribution is optimised in both the normal state and the failure state.

2.4 NP Monitoring

Network Plane Monitoring mechanisms will also be designed in order to provide the necessary input to verify the availability of the provisioned QoS within NPs, and also for dynamic QoS control by NP Provisioning and Maintenance. First of all, NP monitoring interacts with CPA/NIA Assurance to provide appropriate data to verify whether the currently offered service meets the QoS requirement as agreed in the CPA/NIA assurance clauses. In addition, network monitoring may also provide up-to-date feedback on the QoS performance and also traffic distribution condition (e.g., maximum link utilisation) to the online resource and routing control mechanisms of Network Plane Provisioning and Maintenance, which are responsible for taking appropriate actions according to the monitored network/traffic dynamics.

3 NETWORK PLANE ENGINEERING

3.1 Network Plane mechanisms and algorithms

3.1.1 Multi-topology routing

Currently, intra-domain multi-topology IP routing protocols include Multi-Topology OSPF (MT-OSPF) [PSEN06] and Multi-Topology IS-IS [PRZY05]. In order to provide the original IGP protocols with the additional ability of viewing the physical network as multiple independent logical IP topologies independently, each network link is associated with multiple link weights, each identified by a specific Multi-topology Identifier (MT-ID). The original purpose of these protocol extensions was to route different types of traffic such as unicast/multicast or IPv4/IPv6 traffic with dedicated intra-domain paths. In this section, we describe how the multi-topology routing technique can be used for implementing QoS-aware Network Planes in the AGAVE project.

There exist two options to apply multi-topology routing (MTR) techniques to Network Plane engineering. The first option, which we call N-to-one MTR mapping, is to create multiple *equivalent* routing topologies inside one NP for internal load sharing or resilience purposes, meaning that this specific NP is implemented with a set of routing topologies dictated by a single multi-topology IP routing protocol. The second option is that one single routing topology is mapped to a single NP with distinct QoS requirement, and we call this option one-to-one MTR mapping. In this latter case, each MTR topology is engineered specifically according to the QoS requirements for the corresponding Network Plane. Of course, the above two options can be combined to form a more general scenario (Hybrid MTR mapping) – multiple NPs are implemented with MTR for service differentiation, while within some NPs it is still possible to maintain multiple equivalent routing topologies for internal load sharing and resilience purposes, while some others are implemented with one single MTR topology. In this case, the NP design and creation functional block should consider the appropriate number of MTR routing topologies to be implemented and also their allocation to individual Network Planes.

[[Please note that the detailed specification of NP engineering with MTR and the associated algorithms has been suppressed in the public version of this document but will be published in a later stage.]]

3.1.2 INP level overlay routing

Nowadays, overlay networks are often used by both SPs and INPs for the purpose of flexible path selection according to specific requirements such as QoS provisioning, Traffic Engineering and resilience. From service provider’s point of view, the common practice is to perform application layer routing between dedicated service overlay nodes (e.g., to relay VoIP flows across desired media gateway). It should be noted that the SP does not have any knowledge about the physical topology and routing configuration done by the INP. On the other hand, INPs may also use the concept of overlay routing to detour customer traffic from default IGP/BGP paths in order to achieve specific goals such as QoS provisioning [LI04], resilience [ANDE01] and Traffic Engineering [AKEL04][HAN05]. Compared to the design of overlay network of an SP where the underlying physical network (owned by the INP) is effectively a black box, the establishment of INP-level overlays is more sophisticated because the relevant design may take into account the physical network topology and IP routing configurations in order to improve the performance and efficiency of the overlay network. In this section we introduce an NP provisioning scheme using overlay technologies. The main objective is to provide intelligent mechanisms for dynamic path selections within IGP/BGP domains in order to retain the required QoS performances. This scheme can be regarded as a complementary approach to multi-topology routing in the sense that dynamic QoS-aware routing can be enabled even within one single Network Plane (IP routing topology).

The basic idea of overlay networking is to create a virtual network over the physical infrastructure, and apply dedicated routing algorithms on top of this overlay network so as to bypass the underlying

IP topology and routing paradigms such as OSPF and BGP. The design of overlay networks often involves two major stages: topology design and routing design.

- Topology design is responsible for strategically placing a set of overlay nodes within the network, and also establishing overlay links between overlay nodes to form a logical network. Each overlay link may contain one or multiple physical links in the underlying network.
- The task of overlay routing is to apply efficient path selection algorithms on top of the overlay topology to deliver customer traffic. From a routing point of view, overlay networks can be classified into single hop overlays (SHOs) and multi-hop overlays (MHOs). In SHO, only one single overlay node is used for each pair of source and destination, which is normally not on the default IGP/BGP path between the node pair. This is done to detour the traffic from the path that is selected by the conventional IP routing paradigm. In MHO, multiple overlay nodes can jointly work in path selection for individual customer flows. A typical example of MHO is to compute explicit overlay paths hop by hop.

It should be noted that, it is different from MPLS where the underlying IP routing table is completely overridden, because routing between overlay nodes in the physical network still relies on IGP/BGP routing tables [QIU03, LIU05]. In this case, IGP/BGP routing configuration at the underlying IP layer may also impact the performance of overlay networks on top of it. In the next two sections, we will introduce the proposed algorithm for constructing INP-level overlay networks and also describe how overlay routing is performed to improve the QoS performance.

[[Please note that the detailed specification of overlay topology design, routing topology design and Inter-domain overlay networks has been suppressed in the public version of this document but will be published in a later stage.]]

3.1.3 MRDV

In this section, a new routing alternative for DiffRout (Differentiated Routing) NP engineering based on multipath routing mechanisms is introduced. Two main challenges have to be faced: firstly, different paths must be considered in order to create the necessary NPs to meet the QoS requirements for each service; and secondly, optimisation of network resources.

However, the use of multipath algorithms can generate the appearance of routing loops decreasing the efficiency of the routing algorithm. Therefore, a mechanism to avoid loops must also be used.

The work described in this section focuses on intra-domain solutions, that is, for networks managed by a unique operator and running just one instance of a routing protocol.

3.1.3.1 An alternative routing strategy: MRDV

MRDV [REAM02] combines multipath routing with variance and distributed dynamic routing protocols. The core concept of the MRDV algorithm is that alternative paths to route traffic towards a destination are considered when minimum cost paths are congested. Multipath with variance routing algorithms allow traffic to each destination to be carried by other paths in addition to the paths with the minimum cost if the comparison between its metric and a threshold meets the following rule:

$$M \leq M_{\min} \cdot V \tag{1}$$

where M is the metric of the path, $M_{\min}$ is the metric of the optimal path, and V is the variance parameter. It must be noted that ECMP is the particular case when $V = 1$.

MRDV adjusts the variance parameter dynamically, according to the average load that the router detects in the next hop of the optimal path towards the destination. A different variance is defined for each output interface: every router monitors load in its adjacent links and modifies the variance of those interfaces according to their load.

According to the variance, new paths will be considered as suitable: load is distributed among these suitable paths, but the traffic offered to every path is inversely proportional to the path cost, so that the less cost a path has, the more traffic it receives. MRDV distributes traffic properly even when not all the interfaces are overloaded. In this case, only these overloaded links overflow traffic to other interfaces. Therefore, this algorithm is decentralized, lightweight and IP compatible, and also adds the ability to adapt the variance to the traffic demand automatically.

With this approach, every router reacts to its own view of the network state: the average load of its adjacent links. The forwarding decisions are only based on local information and not on global information, as happens with other routing solutions that modify link costs according to the network status. However, two issues must be considered to prevent instability problems in MRDV. First, the variance must describe a hysteresis cycle, where relative increments in variance are proportional to relative increments in average load. Considering that the minimum variance is 1 (ECMP situation), the expression will be the following:

$$\left.\begin{array}{r}\dfrac{\partial V}{V}=K\dfrac{\partial \rho}{\rho}\\ V(\rho=0)=1\\ V(\rho=1)=V_{\max}\end{array}\right\}\Rightarrow\quad V=1+(V_{\max}-1)\cdot\rho^{K}\ ,\tag{2}$$

where K is any real positive number and a design parameter, and V_{max} is the maximum possible variance.

Therefore, the hysteresis cycle is defined by the values of K for each of the two sections (from now on, K_{up} for the ascending curve V_{up}, and K_{dn} for the descending curve V_{dn}) and a common parameter V_{max} for the maximum variance. These parameters define the behaviour of the algorithm. For simplicity, $K_{up}=1/K_{dn}$ is proposed.

The other key issue regarding MRDV stability is the choice of the frequency to refresh the variance parameter as a trade-off between response time and accuracy in measures. Based on our experience with MRDV simulations [RAEG06], the update interval should never be less than about ten seconds, since a shorter update interval could lead to a too unstable behaviour in the presence of bursty traffic.

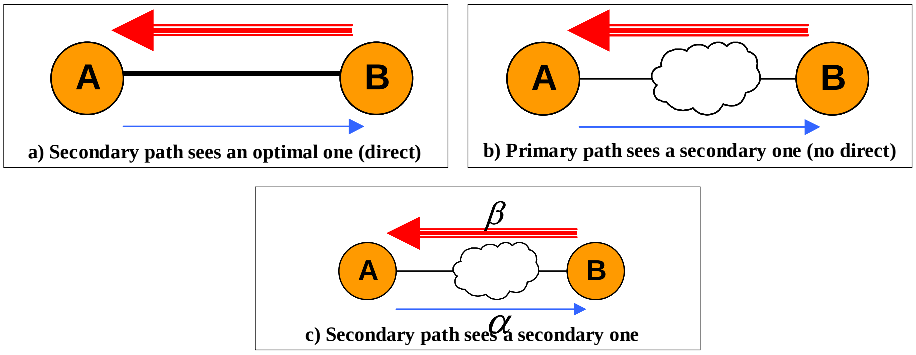
MRDV has been implemented in Network Simulator 2 (ns-2) [NSIM06] and evaluated in different scenarios. Detailed results can be seen in [CAGR06], where MRDV is compared with OSPF without and with ECMP. In a realistic scenario with a typical backbone topology composed of 12 nodes and traffic with different burstiness degrees, the network is able to carry around 35% more traffic with MRDV than OSPF without ECMP, and around 15% more than OSPF with ECMP. In spite of these promising results, routing loops were affecting negatively to the traffic performance in these simulations. Therefore, the LAP protocol was designed in order to avoid these [CAGR06]. This protocol is introduced in the following section.

3.1.3.2 Loop avoidance protocol (LAP)

This protocol distinguishes between two types of loops:

- **Primary loops** or direct (only one hop) loops. In this type of loops, a secondary path sees an optimal one. This situation is shown in Figure 2.a where node A tries to route traffic to a destination D through a secondary path, which has node B as its next hop. However, B has A as its next hop to reach D in its optimal path. This kind of loops is the most predominant one.
- **Secondary loops** include two sub-cases:
    - **Primary path sees a secondary one**: as in the primary loops, a secondary path sees an optimal one. Figure 2.b shows a loop between A and B where there is an optimal path form B to A and a secondary one from A to B to reach the same destination. However, in this case, B has not A as its next hop to reach D in its optimal path.
    - **Secondary path sees a secondary one**: this case is different from the previous two. In this situation a secondary path sees a secondary one. Each secondary path has its

own percentage of routed traffic. Figure 2.c shows a loop caused by two secondary paths, with percentages α and β. It is important to note that this case also includes that scenario where A and B are neighbours.



a) Secondary path sees an optimal one (direct)

b) Primary path sees a secondary one (no direct)

c) Secondary path sees a secondary one

Figure 2 Types of loops

Taking into account this classification, two different mechanisms are proposed to be used when a node is going to install a new secondary path: avoidance of primary loops and avoidance of secondary loops, described in Sections 3.1.3.2.1 and 3.1.3.2.2, respectively.

3.1.3.2.1 Avoidance of primary loops

Avoiding primary loops only requires a simple process to be computed at each router. When a router *A* is going to install a new sub-optimal path through the next hop *NH*, if *NH* has *A* as next hop for its optimal path, the new path is not installed. Since *A* knows both the topology and the link-state information of the network, it is able to infer the optimal paths of *NH* applying the Dijkstra algorithm [CLR90] without any further information exchange.

3.1.3.2.2 Avoidance of secondary loops

In this case, an information exchange is required in order to infer the forwarding and return proportions, α and β (Figure 2.c). Once these are inferred, if β is greater than α, the secondary path is installed. For this purpose, a new message is defined; this is the LAPM (Loop Avoidance Protocol Message) which is the normalized message to be used for this information exchange. Its structure is shown in Figure 3.

Source Node	Destination Node	Next Hop Node	Sink Node	Proportion	Return Proportion	Hops

Figure 3 Structure of the LAPM

LAPM is composed of the following fields:

- *Source Node*: identifies the node that wants to establish the secondary path
- *Destination Node*: identifier of the destination node
- *Next Hop Node*: identifier of the next hop of the secondary path that the *Source Node* wants to establish to reach the *Destination*
- *Sink Node*: identifier of the node that starts the return phase
- *Proportion*: direct proportion of the traffic sent by *Source Node* to the *Destination* through the *Next Hop* that reaches the *Sink Node*
- *Return Proportion*: portion of the traffic sent by the *Sink Node* to the *Destination* that reaches the *Source Node*
- *Hops*: number of nodes the packet can encounter during the transport before being dropped

This mechanism defines three main phases:

- *forward phase:* focused on calculating the percentage of routed traffic by the forward path.
- *return phase*: in order to obtain the percentage of routed traffic by the reverse path.
- *discovery phase:* only triggered if a loop is discovered, i.e., the secondary path is deleted if the *Proportion* is less than the *Return Proportion*.

The *forward phase* is triggered by a node *A*, trying to establish a new secondary path to the destination node *B* by routing a traffic percentage, *p,* through the next hop *NH*. This node sends a new LAPM to the *NH* node, initialized with the set of values *(N,D,NH,-1, p, -1.0, MaxHops)* according to the LAPM format defined in Figure 3. *MaxHops* is the maximum number of hops the LAPM message can leap in the network. This is a configurable parameter that defines the depth of the algorithm and represents a trade-off between loops avoidance and extra load in the network.

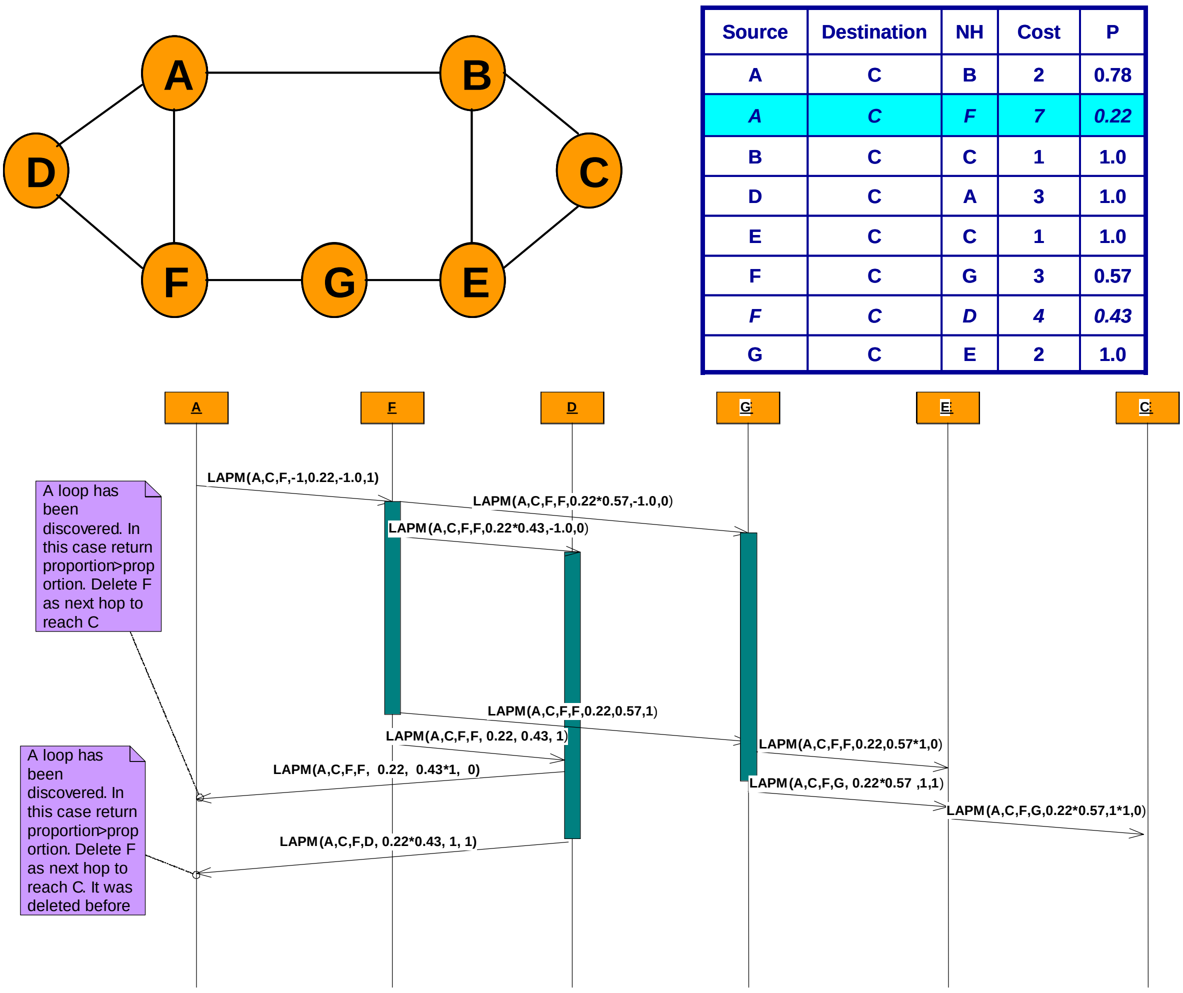
When the node *NH* receives the LAPM, it firstly checks if the received message is a forwarding LAPM (*Return Proportion* field is equal to –1). In this case, if the *Hops* field is greater than zero, the node must resend the message to its next hops to reach *B* updating the values of the *Proportion* field, taking into account the percentage of traffic that the node routes through each one, p_i; therefore, for each next hop to reach *B*, the node has to resend the received LAPM with updated values for *Proportion* and *Hops*. The updated values for these fields are *Proportion**p_i and *Hops*-1 respectively.

In addition to sending the updated LAPM to its next hops, every time a node receives a forward LAPM message it starts a *return phase*. In order to aggregate the forwarding proportions belonging to the same routing tree (whose key is defined by the fields *Source Node*, *Destination* and *Next Hop*), each node must maintain a list with the sum of the *Proportion* fields received in different LAPMs for the same routing tree. Consequently, when the *return phase* starts, if there is another registry in the list for that routing tree, the value of its proportion is updated (the received *Proportion* is added to the stored value). If not, a new registry is added to the list with the values included in the received LAPM, and a timer is triggered for that registry. When this timer expires, the node sends a new LAPM to each next hop of its routing table to reach *D*. The node initializes a new LAPM with the values stored in the list for the *Source Node*, *Destination*, *Next Hop* and *Proportion* and for the fields *Return Proportion*, *Sink Node* and *Hops* it uses p_i (traffic proportion routed to reach *B* from the specific next hop), the identifier of the node and the configurable parameter *MaxHops*, respectively.

On the other hand, when a returning LAPM is received (*Return Proportion* different from *–1*), the node checks if the value of the *Source Node* field is equal to its own node identifier. If this condition is met, a secondary loop has been discovered and the *discovery phase* starts. Otherwise, if the *Hops* field is greater than zero, the return phase continues and the LAPM is resent to all the next hops to reach *B* updating the values of the *Return Proportion* to *ReturnProportion** p_i, where p_i is the proportion of traffic sent by this hop; and *Hops* to *Hops-1*.

Similarly to the return phase policy, in the *discovery phase* each node must also maintain a list to manage the received return LAPMs containing information about the *Destination*, the initial *Next Hop Node*, the *Sink Node* (that one that initialized the return phase), the *Proportion*, the *Return Proportion* and a timer to check if the path must be deleted. Therefore, when a loop is discovered, the node firstly checks if the *Return Proportion* of the received LAPM is greater than the *Proportion* contained in the same message. In this case, the node deletes from its routing table the *Next Hop* to reach the *Destination*. If this is not the case, and there is another registry in the list with the same values of *Destination*, *Next Hop*, *Proportion* and *Sink Node*, the value of the *Return Proportion* is updated by means of adding the just-received *Return Proportion*. If not, the node introduces a new registry in the list with the values received in the LAPM and the initial value of the timer (also proportional to the *MaxHops* configuration parameter). When the timer expires the node checks if fixed *Proportion* is equal or lower than the store, and maybe updated, *Return Proportion*. In this case, the secondary path to reach the *Destination* through the *Next Hop* is deleted.

Figure 4 shows an example of how the LAP algorithm works to avoid a secondary loop.


A
B
C
D
E
F
G
A
F
D
G
E
C
LAPM(A,C,F,-1,0.22,-1.0,1)
LAPM(A,C,F,F,0.22*0.57,-1.0,0)
LAPM(A,C,F,F,0.22*0.43,-1.0,0)
LAPM(A,C,F,F,0.22,0.57,1)
LAPM(A,C,F,F, 0.22, 0.43, 1)
LAPM(A,C,F,F, 0.22, 0.43*1, 0)
LAPM(A,C,F,F,0.22,0.57*1,0)
LAPM(A,C,F,G, 0.22*0.57 ,1,1)
LAPM(A,C,F,G,0.22*0.57,1*1,0)
LAPM(A,C,F,D, 0.22*0.43, 1, 1)

Source	Destination	NH	Cost	P
A	C	B	2	0.78
A	*C*	*F*	*7*	*0.22*
B	C	C	1	1.0
D	C	A	3	1.0
E	C	C	1	1.0
F	C	G	3	0.57
F	*C*	*D*	*4*	*0.43*
G	C	E	2	1.0

A loop has been discovered. In this case return proportion>proportion. Delete F as next hop to reach C

A loop has been discovered. In this case return proportion>proportion. Delete F as next hop to reach C. It was deleted before

Figure 4 Example of the avoidance of secondary loops

This figure shows an example topology (top-left), the paths to reach the node *C* from all the nodes in the example network (top right) and a sequence diagram with all the messages exchanged by the nodes in the network when *A* wants to establish a new secondary path to reach *C* (bottom). In this example, we can distinguish the forward phase (*A* sends a new LAPM to *F*, *F* resends this message to its next hops with updated values of the *proportion* and so on), the return phase (e.g. when the timer expires, *F* sends to all its next hops to reach *C* a new LAPM with the initialized value of *return proportion*) and final the discovery phase (*A* receives a LAPM with the *Source* field equals to *A*, and compares the values of the *proportion* fields and deletes from its routing table *F* as a possible next hop to reach *C*).

3.1.3.3 Extension of MRDV to multiple traffic classes

Until now, a new multipath routing algorithm has been presented. However, one of the key objectives of the AGAVE project is to provide means to provision different classes of QoS. Thus, MRDV algorithm must be extended so that it can operate with different classes of traffic, differentiating the quality demanded by each of them.

In MRDV, load is not equally distributed among paths, but traffic offered to every path is inversely proportional to the path cost. This distribution is done over the whole aggregated traffic in a link, independently of the number of classes which are being carried by it. In this sense, MRDV does not distinguish between different QoS classes.

In the following sections, a proposal for new version of the MRDV algorithm is introduced being able to differentiate traffic classes and generate different Network Planes.

3.1.3.3.1 Proposed algorithm description

The extension of MRDV should only differ from its original version in the way that traffic is distributed among the different possible paths. As previously done with the original MRDV, the decision of assigning different amounts of traffic to different output interfaces is taken locally by each node. The difference resides in that the node should now take into account the different classes of service.

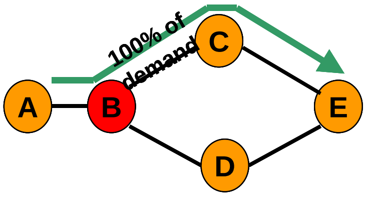
Each router could know each service type looking at the DSCP field (located in the Type of Service (ToS) field of the IPv4 header, or in the Class of Service (CoS) field of the IPv6 header), as DiffServ does.

Taking into account all this, in order to implement NPs to cater for traffic with different QoS requirements the following modification to the MRDV algorithm is proposed. Rather than a common variance parameter for all traffic a separate parameter is maintained for each traffic class on each output interface. When a router calculates a variance parameter value for a particular traffic class it considers the load generated by that traffic class and that offered by all the higher priority traffic classes. Therefore, under high local conditions, lower priority classes will have a higher variance and their traffic will be distributed over more paths. This way, higher priority classes will have more traffic on paths with lower cost, and higher cost paths are left for lower priority classes.

The overall amount of traffic class differentiation introduced by MRDV is configured by adjusting parameters $V_{\max}$ and K. This provides the means for an INP to tune its offering of qualitative service differentiation such as the Olympic (gold, silver, bronze) service classes.

In order to explain the algorithm through a simple example, Figure 5 and Figure 6 graphically describe the mechanism. As the figures show, router *b* has two output interfaces to reach router *e*: interface *b*→*c* and interface *b*→*d*. In this case, the path through router *c* has a cost of 4, while the path through router *d* has a cost of 5. Thus, the alternative path (through *c*) will only be used when the variance parameter is equal or greater than 1.25. The values of $V_{\max}$ and *K* were set to 3 and 1.3, respectively. Thus, attending to the formula in (1) this situation happens when the load in the next hop of the primary path (through *c*) is over a threshold of 0.202.

When the first traffic class is the only one present in the network, the *b*→*c* link is not sufficiently loaded because an average load of 0.12 is considered to be low. Therefore, all the traffic is distributed through the minimum cost path. This situation is presented in Figure 5:



$$V = 1 + (3-1)\cdot 0.12^{1.3} = 1.127$$

$$No.Paths\ Class\ 1 = 1$$

Routing Table

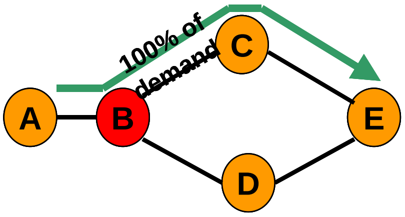
Source A

Destination E

NH	Cost
C	4
D	5

Figure 5 Mechanism operation in a lightly loaded state with one traffic class

However, this situation changes when the second traffic class is present in the network. In this case, traffic class 1 only considers its load. Therefore, its traffic is routed through the minimum cost path only. On the other hand, traffic class 2 not only considers its traffic for the calculation of the variance parameter but also the one corresponding to traffic class 1. In this case, the variance corresponding to the second traffic class is higher than 1.25 and its traffic is distributed through the path *b*→*c* and through the path *b*→*d*. As aforementioned, the proportion of traffic distributed through each link is proportional to the path cost.



$$V = 1 + (3-1)\cdot 0.12^{1.3} = 1.127$$

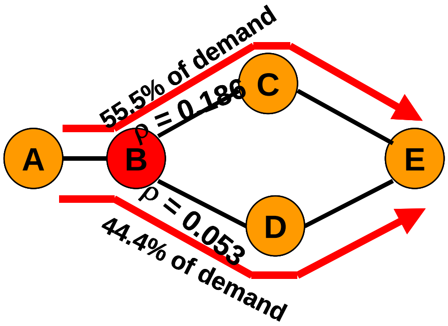
$$No.Paths\ Class\ 1 = 1$$

Routing Table

Source A

Destination E

NH	Cost
C	4
D	5



$$V = 1 + (3-1)\cdot 0.24^{1.3} = 1.3128$$

$$No.Paths\ Class\ 2 = 2$$

Figure 6 Mechanism operation in a more loaded state with two traffic classes

As it can be observed from the example, the traffic class with higher priority is served using the optimum path, that provides a better QoS. On the other hand, the second traffic class has to distribute its traffic through the two possible paths. Besides, by using this algorithm, every router sees a different network, which also makes its behaviour different from each other. Since variance adapts dynamically to average load, it is possible for the flows to be distributed according to the network load, avoiding the concentration of traffic in a few links (the least cost ones). The key is that not only optimum paths are employed (as done by ECMP), but also a complete set of next best paths. This set of paths is chosen according to a more or less restrictive rule (by increasing or decreasing variance) depending on how loaded the adjacent links are.

In short, a router must invoke the procedure described in Figure 7 for each destination included in the routing table. This figure explains, not only the hysteresis cycle, but also how load is unequally distributed among paths. Traffic offered to every path is inversely proportional to the path cost, so that the less cost a path has, the more traffic it receives.

INPUT:

- $\rho[\kappa]$: Current average load of the next hop of the primary path for each traffic class κ.
- $\upsilon[\kappa]$: Current average load of the next hop of the primary path, considered for variance parameter calculation, for each traffic class κ.
- $\upsilon_{old}[\kappa]$: Previous average load of the next hop of the primary path (in the previous update) , considered for variance parameter calculation, for each traffic class κ.
- $V_{old}[\kappa]$: Variance in the previous update for each traffic class κ.
- $C[i]$: Vector of costs for every feasible path to reach the destination. ($i \in [1$, *max. no. of feasible paths*], where $i = 1$ is the primary path).

OUTPUT:

- $W[i][\kappa]$: Vector of traffic-share weights for every feasible path to reach the destination ($i \in [1$, *max. no. of feasible paths*], where $i = 1$ is the primary path) and for each traffic class κ

ALGORITHM:

```
1. Computation of the new value of $\upsilon[\kappa]$
        for $\kappa = 1$ to max. no. of traffic classes do
            if ($\kappa = 1$) then
                $\upsilon[\kappa] = \rho[\kappa]$
            else
                $\upsilon[\kappa] = \upsilon[\kappa-1] + \rho[\kappa]$
2. Computation of the new value for variance of traffic class $\kappa$, $V[\kappa]$:
        if ($\upsilon[\kappa] > \upsilon_{old}[\kappa]$) then
            if ($V_{up}[\kappa](\upsilon[\kappa]) > V_{old}[\kappa]$) then
                $V[\kappa] = V_{up}[\kappa](\upsilon[\kappa])$
            else $V[\kappa] = V_{old}[\kappa]$
        elseif ($\upsilon[\kappa] < \upsilon_{old}[\kappa]$) then
            if ($V_{dn}[\kappa](\upsilon[\kappa]) < V_{old}[\kappa]$) then
                $V[\kappa] = V_{dn}[\kappa](\upsilon[\kappa])$
            else $V[\kappa] = V_{old}[\kappa]$
        else $V[\kappa] = V_{old}[\kappa]$

3. Computation of $W[i][\kappa]$:
        $invC_{total} = 0$
        for $i = 1$ to max. no. of feasible paths do
            if ($C[i] \leq C[1] \cdot V[\kappa]$) then
                $invC_{total} = invC_{total} + (1 / C[i])$
        for $i = 1$ to max. no. of feasible paths do
            if ($C[i] \leq C[1] \cdot V[\kappa]$) then
                $W[i][\kappa] = 1 / (C[i] \cdot invC_{total})$
            else $W[i][\kappa] = 0$
```

Figure 7 Description of the algorithm

3.2 AGAVE monitoring architecture

Within AGAVE project, we define a monitoring architecture focusing on Service Providers, IP Network Providers and end users (also named Customer). This architecture defines the interfaces invoked when deploying inter-provider service offerings, the monitoring points and the monitoring servers that collect and synthesize information from monitoring points.

3.2.1 Interfaces

Within monitoring architecture, input and output interfaces are represented separately so that monitoring functions can be associated to one or the other depending on the business role. A total of four interfaces are represented in the figure below:

- *Service Level Interface* (SLI): between a Customer and a Service Provider. This interface supports all signalling and media exchanges between the Customer and the Service Provider that are needed for the Customer identification, service establishment and release and the SLA management. These exchanges do not occur directly but may cross the IP infrastructure of the INP offering connectivity services to the customer;
- *SP Interconnection Interface* (SII): between two Service Providers. This interface enables all signalling and media exchanges between Service Providers that are needed for service establishment and release and for the SIA management such as exchange of monitoring data for verification and assurance purposes. These exchanges might occur directly (in case of back to back SP equipments), but will usually cross the IP infrastructure of one or several INPs that enable connectivity between the two Service Providers;
- *Connectivity Provisioning Interface* (CPI): between a Service Provider and an IP Network Provider. Within monitoring architecture, this interface supports the transport requirements of the traffic generated by the SP and carried by the INP and the CPA management requirements. The interconnection of the SP and the INP equipment might be direct (in case of local CPA agreement) or through one or several INPs (in case of remote CPA agreement).
- *INP Interconnection Interface* (NII): between two IP Network Providers that have agreed a NIA. Within NP monitoring architecture, this interface supports any flows exchanged between one Network Plane of one INP to one Network Plane of the other INP and the information exchanged for NIA management.

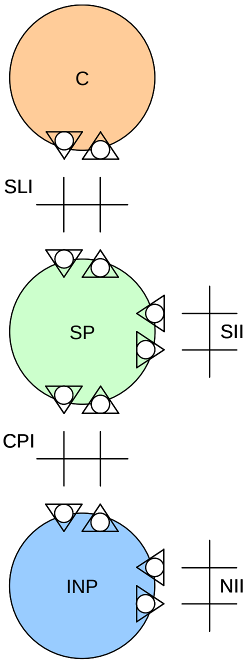


Figure 8 VoIP interfaces

3.2.2 Monitoring points

A monitoring point can be defined as a network element function with defined processing capabilities that might send and/or receive traffic at the service level or transparently as IP traffic at the IP level.

Active monitoring points will send and receive test traffic only generated for the purpose of monitoring, and passive monitoring points will only receive real traffic generated by end users. Active and/or passive monitoring points might be deployed to any input or output interface of the monitoring model.

Monitoring points might have limited processing capabilities, such as capabilities related to computing statistics for each metric over a monitoring period, storing these statistics before sending them to a Monitoring Server.

A monitoring mechanism might use a mix of active and passive monitoring points, located at any input/output of the monitoring architecture. For instance a monitoring point located at CPI interface will perform monitoring for all flows between a given SP and a given INP, another monitoring point located at NII interface will perform monitoring for all flows exchanged between two given INPs.

3.2.3 Monitoring server

The Monitoring Server (MS) is responsible for managing the monitoring point in accordance to CPA, SIA and NIA agreements. The Service Provider Monitoring Server will manage monitoring points located at SLI, SII and CPI interfaces. The IP Network Provider Monitoring Server will manage monitoring points located at CPI interfaces.

The Monitoring Server is responsible for activating monitoring points at each boundary identified by the agreements between parties. The monitoring point parameters such as the monitored metrics, the computed statistics and the monitoring period might also be set by the Monitoring Server.

The Monitoring Server will receive the metric statistics from the monitoring points and will have to perform the storage of these statistics. Based on these statistics, the Monitoring Server will have to generate synthetic reports that might be exchanged between SP and INP. Notice that parameters like synthetic report periodicity, hours to monitor during the day or other characteristics of the synthesis are set following the agreement requirements.

In order to generate synthetic reports, the SP Monitoring Server will have to compare different metrics:

- SIA synthetic report should enable a SP to check whether it has met the requirements (Assurance) of the SII and to check whether the peer SP has met its requirements (Verification). For that purpose, metrics monitored at SII and CPI interfaces will have to be compared.
- CPA synthetic report should enable a SP to check whether the INP has met the requirements of CPA (Verification). For that purpose, metrics monitored at CPI interface will have to be used.
- SLA synthetic report should enable a SP to check if a given Customer SLA has been met or not (Assurance). For that purpose, metrics monitored at SLI interface will have to be used.

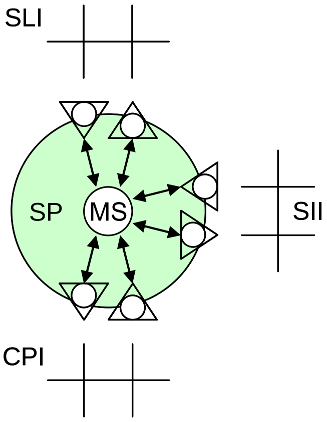


Figure 9 Service Provider Monitoring Server

In order to generate synthetic reports, the INP Monitoring Server will have to compare different metrics:

- CPA synthetic report should enable an INP to check whether it has met the requirements of the CPA (Assurance). For that purpose, metrics monitored at CPI interface will have to be used.
- NIA synthetic report should enable an INP to check whether it has met the requirements (Assurance) of the NIA and to prove that the peer INP has not met its requirement (Verification). For that purpose, IP traffic metrics monitored at CPI and NII interfaces will have to be compared.

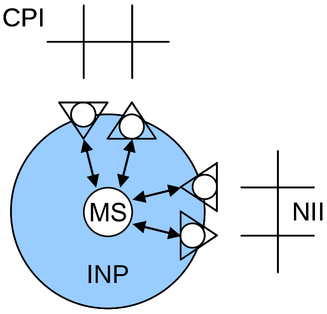


Figure 10 INP Monitoring Server

3.2.4 Examples

This section aims to illustrate the application of the monitoring architecture to some particular situations raised in WP1. The first situation is the case of a simple call establishment between a source Customer S and a destination Customer D (refer to Figure 7 of [D1.1]). The following figure illustrates the corresponding architecture. The signalling flows exchanged between S, SP1, SP2 and D could be monitored by passive monitoring points highlighted in bold.

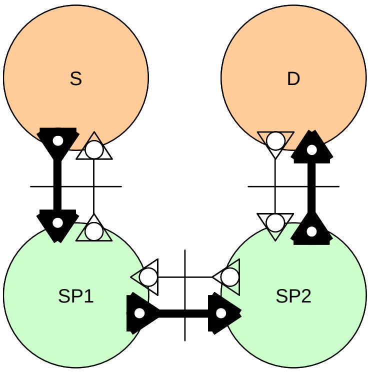


Figure 11 Call establishment monitoring

The second situation is the case of a single hop remote CPA (refer to Figure 24 of [D1.1]). The following figure illustrates the corresponding monitoring architecture. Although SP1 has a direct interface to INP1, this interface is not considered as a CPI interface. The actual CPI interface starts from SP1 and ends to INP2. The corresponding monitoring point at INP2 will only monitor signalling or media flows exchanged between SP1 and INP2.

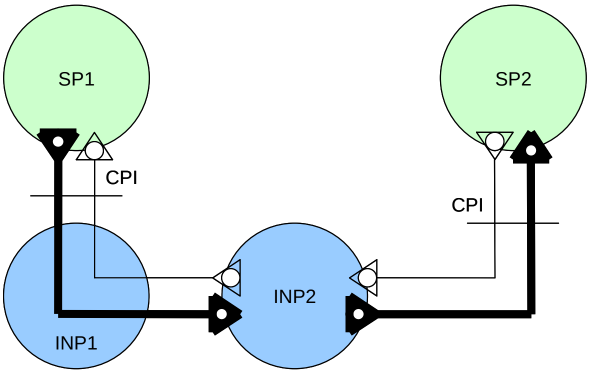


Figure 12 Remote CPI

3.2.5 Other considerations

It has to be noticed that the different parties involved in monitoring will have at least to agree on the metrics to monitor and the way they should be reported so that issuing and reception of complaints is agreed by each party. However, a common monitoring mechanism is not required if monitored metrics are still comparable.

Synchronization between monitoring points might be necessary within a SP or an INP so that monitoring periods start and end at the same time and so that monitored metrics can be compared with the same reference time.

3.3 Network Plane monitoring

This section specifies the framework for NP monitoring mechanisms within the context of AGAVE functional architecture as defined in WP1 [D1.1] and using monitoring architecture defined in section 3.2.

3.3.1Objectives

As defined in WP1 [D1.1], NP monitoring function should meet the following requirements:

- (R1) Monitor activities per Network Plane. *Indicators such as effective load/throughput per interface, packet loss, delay and jitter between two interfaces*;
- (R2) Monitor activities per CPA. *Identification of monitored data associated with a given CPA should be performed, in order to enable exchange of information related to the CPA with the corresponding Service Provider.*
- (R3) Monitor activities per NIA. *Identification of monitored data associated with a given NIA should be performed, in order to enable exchange of information related to the NIA with the corresponding IP Network Provider.*

3.3.2NP monitoring implementations

Active or passive monitoring points might apply to any input or output interface of the NP monitoring model. On any interfaces (CPI or NII), the NP monitoring points will monitor network quality metrics (packet loss, delay and jitter) or other network metrics (traffic load, etc.).

Passive monitoring will only make use of passive monitoring points, so that only real IP traffic generated by end users will be monitored. Several issues might be encountered:

- Many flows and flow types have to be monitored at the same time. The complexity of the monitoring function will increase with the number of flows and the number of flow types to monitor. Aggregation based on IP address, IP sub network or input/output interface should be used to reduce complexity.
- Packet loss, delay and jitter of a given flow can only be monitored through correlation between measurements performed at INP ingress and egress. A precise synchronization mechanism has to be setup in order to make this correlation.

Active monitoring will make use of active monitoring points sending and receiving IP test traffic. Several issues might also be encountered:

- Active monitoring generates additional IP traffic that does not generate revenues.
- Active monitoring only measures quality of several IP test flows among the real IP flows, so that it cannot be representative of any traffic flow, in terms of packet length, sending rate and periodicity.
- The evaluation of the amount of real IP flows that might be impacted by the same degradation encountered by IP test traffic can not be done on line. An off line weighting taking into account rush hours and off-peak hours has to be performed.
- In order to evaluate the amount of real IP flows that might be impacted by a degradation encountered by IP test traffic, the traffic load still have to be monitored in a passive way.

3.3.3Other considerations

INP monitoring servers will only give information about Network Plane performance in intra domain. This information should be aggregated in order to reflect the horizontal binding of Network Planes. Delay measurements or unavailability could be for instance added in order to get the end to end delay and unavailability across a set of different INP Network Planes.

Jitter metrics measured by VoIP Service Providers and delay variation measured by IP Network Providers will be compared when exchanging information between them through the interaction of Assurance and Verification functional blocks. The definition of these metrics should match as much as possible in order to enable this comparison.

3.4 Resilience and Fast Reroute

As IP networks have been designed for Best Effort traffic, no effort has been done to set up their availability up to the carrier class reference of five nines (0.99999). In order to meet requirements of

CPA agreements they have with Service Providers offering real time applications such as voice over IP, video telephony over IP that require high availability, IP Network Providers will have to build Network Planes with a better availability than for Best Effort applications. As described in WP1 ([D1.1] Appendix B), availability depends on the Mean Time Between Failures (MTBF) and on the Mean Time To Restore (MTTR). In order to increase availability for specific services, INP will have to decrease MTTR and to increase MTBF.

3.4.1 Reducing MTTR

In best effort IP networks, resilience to failures is based on dynamic routing protocols. In case of a link or an IP node failure, distance vector or link state IP routing protocols update the IP node routing tables so that IP traffic shall be routed to a valid path towards its destination. However, the slow convergence of the distributed routing tables hardly enables to reroute IP traffic in less than 1 second. In order to decrease MTTR, INP should use an alternative technology, such as Multi Protocol Label Switching (MPLS) that enables to set up additional resilience mechanisms.

MPLS technology can be implemented in different ways. The usage of Label Discovery Protocol (LDP, IETF RFC 3036) for Label Switched Path (LSP) establishment enables to dynamically reroute LSP in case of link or IP nodes failures, but LSP rerouting with LDP relies on IP routing protocol and need additional time to reroute LSP, so that LDP LSP traffic interruption will still remain above 1 second.

The usage of Resource Reservation Protocol (RSVP-TE, IETF RFC 3209) enables to specify an explicit route when establishing an LSP. This route can be computed by the head router of the LSP if a link state IP routing protocol such as Intermediate System to Intermediate System (IS-IS) or Open Shortest Path First (OSPF) protocol is used. However, an intermediate router along the explicit route might not accept the establishment of the LSP due to lack of resources, so that the LSP establishment might not succeed. In that case, the head end router should attempt to establish again the LSP with another explicit route. Two resilience mechanisms can be set up that way. The restoration mechanism based on a new explicit route computation and LSP establishment attempt after a link or an IP node has failed remains slower than 1 second, because it relies on IP routing protocol convergence. The protection mechanism based on the pre-establishment of disjoint active and standby LSP enables to switch traffic from active to standby LSP once the failure of the active LSP has been notified to the head end router. Recovery time depends on the speed of notification messages that are propagated by intermediate routers back to the head end one.

Restoration or protection mechanisms triggered by head end router do not enable to guaranty repair times less than 100 ms, only a local mechanism like MPLS Fast Reroute (MPLS FRR, IETF RFC 4090) does. MPLS FRR protection consists in locally protecting an active LSP by a set of several pre-established standby LSP that enable to protect against link or IP nodes failures. Any intermediate router along the active path that detects a local link or IP adjacent node failure can immediately redirect active LSP to the standby LSP protecting the failing element. A given standby LSP can protect a single active LSP (one to one backup) or several active LSPs (many to one backup). As MPLS FRR mechanism is only local, the repair time can be decomposed into a detection time (typically 50 ms) and a predictable protection time that depends on the number of active LSP to protect and on the number of standby LSP. Depending on the local router speed and usage, the number of active and standby LSP should be limited to a proper value so that protection time should not exceed 50 ms.

3.4.2 Increasing MTBF

The second way to increase Network Plane availability is to increase the Mean Time Between Failures. Best Effort IP networks are built to provide IP connectivity taken into account any kind of single link or IP node failure. Dynamic routing protocol is then supposed to reroute traffic each time a link or an IP node is out of order. However, no distinction is made between a real failure which occurrence is not predictable and an action performed during a maintenance operation, such as the activation/deactivation of an IP link or a software or hardware reboot of an IP node. All those events will trigger a dynamic update of the routing tables and an IP traffic interruption as long as the routing table update process has not converged.

In order to increase MTBF for a given Network Plane, IP Network Providers should implement mechanisms that enable to prevent an automatic update of the overall network configuration for predictable actions performed during maintenance operations.

Prior to any planned maintenance, the Graceful Shutdown [ALI06] could be applied to the high availability Network Plane. For LSP established with RSVP-TE, this mechanism consists in allowing an intermediate router to send a RSVP-TE notification for each LSP using a link that will be shut down or attached to a node that will be shut down. Using this notification containing the identifier of the link that will be shut down, the head end router will compute a new explicit route excluding that link and establish a new LSP. Using a make before break procedure, head end router will switch traffic onto the new LSP before removing the old one.

4 NETWORK PLANE BINDING

4.1 IP tunnelling

4.1.1 Introduction

The current Internet routing architecture, which dates from the mid 90's [RFC1771], has been designed to provide reachability among Internet domains and to ensure a best-effort transport service. A consequence of this design is that the inter-domain routing protocol, BGP, is unaware of the paths performance. Today, a growing number of applications are emerging that would benefit from improved or guaranteed performance. Voice or Video over IP, for example, are applications where a bounded latency has a direct impact on the users' perception of the performance. *Virtual Private Networks* (VPNs) are another service where performance and robustness matter. In parallel to this, Internet users are now getting prepared to pay for increased performance. Though, up to now technical means to provide better-than-best-effort service in the Internet have not been implemented.

The large majority of proposals for deploying guaranteed performance services in the Internet need significant changes if not a radically new architecture. To deploy QoS at the inter-domain level, one needs to ensure coherence and consistency of treatment when crossing several independent domains. Techniques allowing treating packets in a differentiated manner should be deployed inside each domain [RFC2475]. In addition, mechanisms such as QoS NLRI [CRIS03], or q-BGP [BOUC05] should be introduced to propagate information on the quality of available routes. The Hybrid Link-state Path-vector protocol (HLP) [SUBR05] is another inter-domain routing protocol proposed as a replacement for BGP that could increase the diversity of Internet paths. However, the above proposals require that the majority of the domains support new protocols. In the case that such mechanisms are ever deployed in the Internet, it is unlikely that this will happen before a long time.

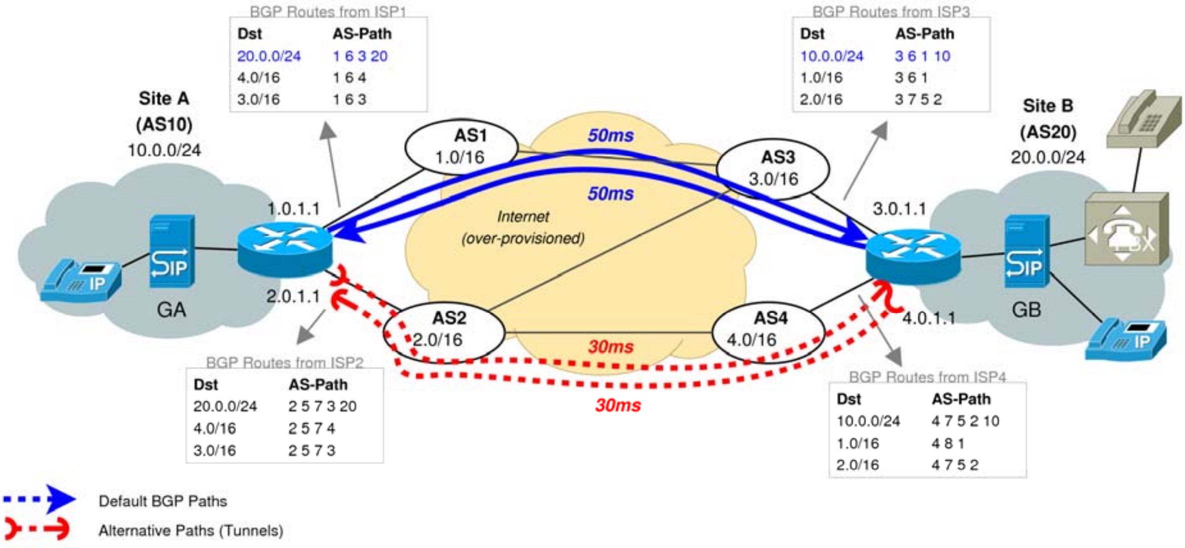
Meanwhile, it is still possible to provide a better-than-best-effort service even if most domains do not support traffic differentiation mechanisms or the above routing protocols. In this chapter, we describe and lay out the architecture of a lightweight approach to provision a better-than-best-effort service relying on the current Internet routing and the use of IP tunnels. Before delving into the details of the solution, we need to clarify the place held by the IP Tunneling solution in the general AGAVE framework. One of the means proposed by the AGAVE project for providing lightweight end-to-end QoS in the Internet is the creation of Parallel Internets. A Parallel Internet is an interconnection of Network Planes managed by multiple INPs for the purpose of providing specific performance guarantees or services consistently across multiple INPs.

The IP Tunneling solution we propose does not aim at entirely building such Parallel Internets. There are two main reasons for this. The first reason is that the IP Tunneling solution does not target all the INPs, but only a subset of the stub domains. IP tunnels are established between specific pairs of stub domains and for forwarding a subset of the traffic flows. That means that the solution will not establish tunnels towards all destinations (which would not be a scalable approach). The second reason is that IP tunneling alone cannot provide strict quantitative QoS guarantees. It rather builds on the availability of excess resources in the Internet for providing performance enhancements. Previous studies have shown that such resources exist but are not currently exploited [LAUN05]. If resources can be leveraged by IP tunnels for satisfying the network operator or customer requests, the IP Tunneling approach will use them. However, if these resources are not available, the requestor will be notified and a best-effort service will still be provided.

This document is organized as follows. We first give in Section 4.1.2 an example where using IP tunnels is an efficient solution for improving the performance of the traffic exchanged by two Internet domains. Then, we state the problem we want to solve with IP tunnels and what issues need to be tackled in Section 4.1.3. We lay out the functional architecture of the solution in Section 4.1.4. We detail the system components in Section 4.1.5 and we further discuss how they could be implemented. Finally, we conclude in Section 4.1.6.

4.1.2 Overview

Suppose that we are in a situation where a company has two sites *A* (*AS10*) and *B* (*AS20*). Site *A* is multi-homed to *AS1* and *AS2* while site *B* is multi-homed to *AS3* and *AS4*. The company is currently using VoIP to place calls between users located in the two sites. For this purpose, a SIP Proxy Server is deployed in each site: *GA* is the SIP Proxy Server in *A* and *GB* is in *B*. For the moment, they suffer from horrible delays between the two sites, due to the routing choices made by BGP (see Figure 13). Indeed, the border router in site *A* has received two routes towards the prefix of site *B*: one with AS-Path (1 6 3 20) and the other with AS-Path (2 5 7 3 20). Note that we do not show the intermediate ASes 5, 6 and 7 in the figure. The first route, through *AS1*, is preferred since its AS-Path is shorter. On the other side, site *B* has received one route towards *AS10* from *AS3* with AS-Path (3 6 1 10) and another one from AS4 with AS-Path (4 7 5 2 10). The border router of site *B* has selected the route through *AS3*. However, the latency of the path (1 6 3) is 50 ms while that of path (2 5 7 4) is 30 ms. It is frequent that the quality of a BGP route as determined by the BGP decision process is not correlated with its latency, as shown by [HUFF02]. In addition, there are often alternative paths learned by BGP that are not used for forwarding packets, as shown by [LAUN05]. These paths may often offer better latency than the best BGP routes [QUOI06].



BGP Routes from ISP1

Dst	AS-Path
20.0.0/24	1 6 3 20
4.0/16	1 6 4
3.0/16	1 6 3

BGP Routes from ISP3

Dst	AS-Path
10.0.0/24	3 6 1 10
1.0/16	3 6 1
2.0/16	3 7 5 2

BGP Routes from ISP2

Dst	AS-Path
20.0.0/24	2 5 7 3 20
4.0/16	2 5 7 4
3.0/16	2 5 7 3

BGP Routes from ISP4

Dst	AS-Path
10.0.0/24	4 7 5 2 10
1.0/16	4 8 1
2.0/16	4 7 5 2

Figure 13 Using tunnels to improve the latency between two SIP gateways.

The objective of the two sites is therefore to use the alternative path with the lowest latency for the traffic exchanged between the two SIP gateways. The remaining of the traffic should continue to go through the current BGP route as it is assumed that using this route is cheaper than sending and receiving traffic through *ISP2*.

From the viewpoint of site *A*, it is possible to send the traffic destined to *GB* over the peering link with *ISP2*, since a route towards the prefix of site *B* is received from *ISP2*. However, this traffic would still enter site *B* through *AS3* and the latency of this path (2 5 7 3 20) is not better than that of the best BGP route. In addition, it is not possible for site *A* so send its traffic to site *B* through *AS4* by influencing the BGP routing decisions. It is depending on the routing decisions taken in *AS2*. One possible solution in this case is to encapsulate the traffic destined to site *B* in a tunnel whose tail-end is the IP address of the border router of site *B* attached to *AS4*. This IP address is *4.0.1.1* and it belongs to the prefix *4.0/16* advertised by *AS4* and reachable from site *A* through the AS-Path (2 7 5 4). The packets sent through this tunnel will follow the path with the lower latency. Solving the problem in the reverse direction, i.e. for the traffic sent by site *B* to site *A* is possible with a similar solution, as shown in Figure 13. Note that although the inter-domain paths shown in Figure 13 looks

Such tunnels can be setup manually, but this is a slow and error-prone process. Moreover, the latency along the path through *ISP2* is subject to changes, due to the evolution of the traffic conditions or even to route changes between *ISP3* and *ISP2*. For these reasons, an automated establishment of these tunnels is preferable, which need to be coupled with a path performance monitoring process.

In the remaining of the document, we describe the architecture of a framework that allows to exploit the diversity of inter-domain paths by relying on the establishment of IP tunnels. The framework allows to specify the metric that must be optimized (end-to-end latency, available bandwidth ...) for a given destination or for a given source/destination flow. The framework should also allow monitoring the current performance of the available Internet paths and automatically select the best suited path. The framework should of course avoid frequent path switching for obvious stability reasons.

4.1.3 Problem statement

Generally speaking, the problem we want to solve is the following. Given a set of cooperating sites which each have multiple ingress/egress points, find and setup the best suited inter-domain paths for exchanging traffic among them (Figure 14 illustrates the case of 2 participants). The best paths are paths that optimize the local objectives of each participant while satisfying their local constraints. The local objectives of a site could be for instance to use the paths with the lowest latency or the highest bandwidth for a given traffic flow specification [RFC1363]. In the example of Figure 14, the constraint for one flow from INP *A* to INP *B* could be to use the path with the lowest latency. The default path, exiting *A* at egress *A3* and entering *B* at ingress *B2* might have a higher delay $d_{3,1}$ than the path through *A1* and *B1* with delay $d_{1,1}$. Using the path *A1-B1* requires *A* to direct the traffic towards *B1* through *A1*. In addition, encapsulating the packets into a tunnel could be needed since the routing decisions taken by routers between *A1* and *B1* (in the "*Internet cloud*") might direct the traffic through another ingress of INP *B*. If the path followed by the traffic flow must be controlled in both directions, i.e. from INP *A* to INP *B* and the other way round, then two tunnels must be established, one for each direction.

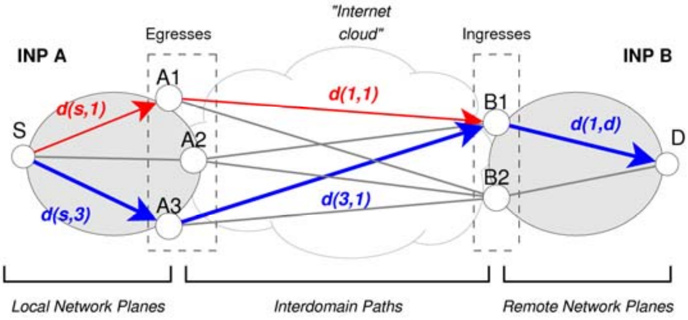


Figure 14 - Multiple paths between two AS.

The optimization objectives that can be achieved using IP Tunneling are not limited to improving the latency of a set of inter-domain paths. Objectives such as load-balancing [QUOI05, QUOI06] or minimizing the peering cost, for instance, can involve moving traffic flows from a peering to another in both directions. For example, INP *A* might want to balance the traffic load that is exiting and entering its network on the various border routers (*A1-A3*). This might require choosing a different egress for some traffic flows exiting *A* but also asking *B* to direct some traffic flows entering *A* on a different egress point. Simultaneous optimization of multiple objectives might also be conceived.

In addition to these optimization objectives, each site might define constraints on the utilization of its own resources by others. For instance, a site might define that a maximum bandwidth of an access link is devoted to the cooperating peers. Another constraint would be to allow only selected participants to make use of an access link.

The above problem statement can be refined in several sub-problems as follows. First, each participant should be able to **discover the other participants** along with their capabilities (ingress/egress points) and constraints. We envision two possible approaches. In the first one, all the participants have an a priori knowledge of each other, either because they belong to the same administrative authority or because they participate to a common application/service. In this case, a protocol can be used among them to advertise and discover capabilities and constraints or these capabilities can be exchanged manually. In the second approach, the system is open and each participant registers in a global directory service. All the other participants are therefore able to discover who controls the resources they need by browsing the global directory. We will assume the second approach in the remaining of the document.

The second problem is the **selection of the paths** that will be "installed" to forward the traffic. This selection is the outcome of a distributed multi-objective optimization process. The following questions are open:

1. There might be a lot of different paths and the assignment of each traffic flow on a path that both meets the flow requirements and satisfies the global optimization objective can be complex due to the combinatorial number of possibilities.
2. It might not always be possible to break the ties between two solutions. Typically, if one wants to simultaneously minimize the latency and maximize the bandwidth allocated to a particular flow, multiple incomparable solutions might exist. For instance a solution with lower delay and lowest bandwidth cannot be compared to a solution with higher bandwidth but higher latency (no solution dominates the other).
3. The objective functions of the different sites might be conflicting (see Figure 15). For instance, consider a situation involving two multi-homed sites *A* and *B* which currently experience a latency of 100ms and an available bandwidth of 5Mb/s along the default BGP path. Site *A* might want to optimize its routing for latency and sends its traffic along an alternate path with a 50ms latency and an available bandwidth of 5Mb/s. To the opposite of *A*, site *B* wants to optimize bandwidth and decides to direct its traffic along another path with 100ms latency but with an higher available bandwidth: 10Mb/s. In this case, not all the objectives will be met since the traffic in one direction will follow *A*'s path and come back through *B*'s path. Finally, the path selection may cause forwarding instabilities if not carefully done.
4. The selected paths need to be installed in the network. There are two parts in this process. First, if IP tunnels are needed for exploiting some of the selected paths, they will have to be established. This will typically require configuration changes on the border routers of each participant. Second, the traffic flows must be directed inside the tunnels they are associated with. This might require announcing more specific routes within the network of each participant (in case that no Network Planes are supported) or assigning the traffic flow in the selected Network Plane. In the case of multi-topology intra-domain routing [PRZY06, PSEC06] for instance, this is achieved by configuring the *Provider Equipment* (PE) router that connects the source network so that the specified traffic is marked with the *Differentiated Service Code Point* (DSCP) value associated with the selected routing topology. If, instead of multi-topology routing, we rely on intra-domain MPLS LSP from the PE router to the egress ASBR, we would need to add the right MPLS label to the packets directed from the PE to the egress ASBR. We will also consider an optional bandwidth reservation in the local and remote Network Planes (if supported).

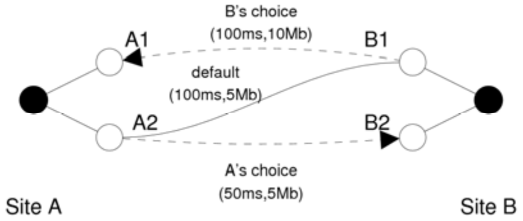


Figure 15 - Conflicting objectives.

In addition to this, we must be able to measure the performance of the available inter-domain paths in order to compare them and select the most suitable ones. The performance metrics that would be considered in the case of the IP tunneling approach are the end-to-end path latency and the available bandwidth. The measurement would typically take place once a participant has learned that another participant has multiple ingress points available and when it needs to figure out what are the current performances of the paths going through these ingresses. In addition, once a path is selected to carry traffic, it needs to be continuously monitored in order to detect performance degradation and trigger the selection of an alternative path. There are two main problems with the measurement. First, active measurement might be needed to obtain the performance of the various paths. Second, it might not be possible to perform this active measurement for all paths without installing the necessary state in the network, i.e. by establishing the IP tunnels.

Finally, security issues could be raised by the ability to direct traffic towards specific entry points in the participant networks. It could be possible for an attacker to target *Denial of Service* (DoS) attacks to specific access links of a participant network. It could also be possible for an attacker to forge encapsulated packets directed to a tunnel tail-end in order to perform spoofing and attack another network. In addition to this, the introduction of new protocols for discovering the other neighbors and their capabilities needs to be done carefully. Special attention must be paid to the ability to authenticate the other participants and the management messages they issue. Indeed, an attacker could try to steal the identity of a participant and advertise erroneous information, redirect traffic towards its own network for spying reasons or even cause a *Distributed Denial of Service* (DDoS) attack by redirecting traffic from other participants to the victim.

4.1.4 Functional architecture

In this section we describe the functional architecture of the IP Tunneling solution and how it integrates in the general AGAVE framework. In particular, the IP Tunneling solution makes use of the Network Planes deployed in the cooperating stub domains when such Network Planes are available.

4.1.4.1 Overview

We show in Figure 16 which functional components of the AGAVE framework are involved in the IP Tunneling solution. We detail the purpose of each component in the following paragraphs. There are three main parts: (1) the components responsible for defining the system configuration and the traffic flow constraints; (2) the components responsible for discovering and negotiating inter-domain paths and (3) the components responsible for selecting which paths must be used to meet the constraints.

First, the *CPA Order Handling* and the *Business-based Network Development* components are responsible for defining the system configuration and the traffic flow constraints. The role of the *CPA Order Handling* component is to receive the traffic flow constraints from the network operator, from a customer network operator or from a service provider; to check their validity and to store these constraints for future use. Typical *CPA Orders* contain latency and bandwidth constraints for specific traffic flows. The role of the *Business-based Network Development* component is to receive the definition of the network objectives and the system configuration, to check their validity and to store them for future use. The typical global network objectives considered in the IP Tunneling approach are minimizing the peering cost and balancing the traffic load over the peering links.

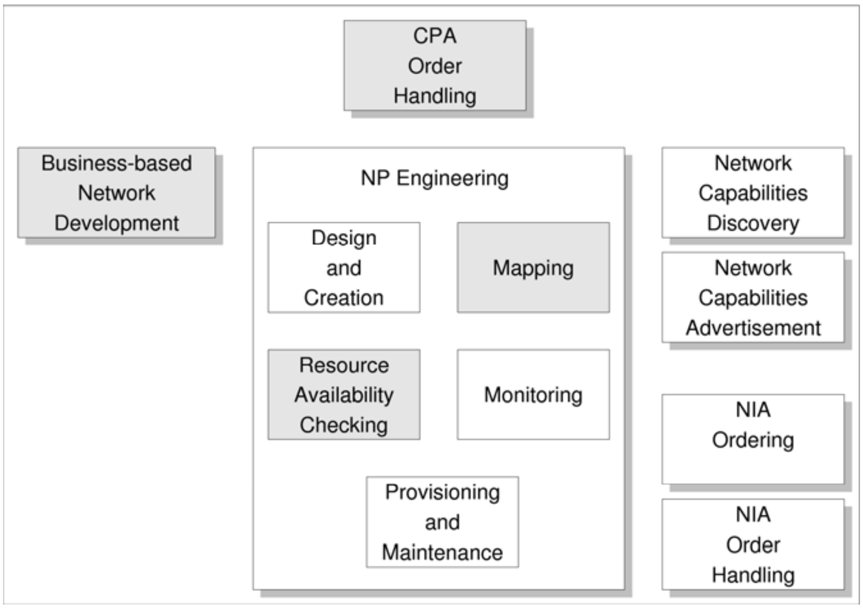

CPA Order Handling
Business-based Network Development
NP Engineering
Design and Creation
Mapping
Resource Availability Checking
Monitoring
Provisioning and Maintenance
Network Capabilities Discovery
Network Capabilities Advertisement
NIA Ordering
NIA Order Handling

Figure 16 - Functional components involved in the IP Tunneling solution.

Second, the components responsible for discovering and negotiating the inter-domain paths are the *Network Capabilities Discovery* and the *NIA Order Handling* components. The role of the *Network Capabilities Discovery* component is to obtain from a remote INP running the IP Tunneling solution the list of its ingress points along with their capabilities and parameters. The role of the *NIA Ordering* component is to request from a remote INP the utilization of an ingress point previously discovered. The result of this request is an agreement (the NIA) which may contain guarantees such as bandwidth reservation. Both components have their counterparts which are the *Network Capabilities Advertisement* and the *NIA Order Handling* components. The *Network Capabilities Advertisement* component is responsible for advertising the list of the local ingress points to a requestor. The *NIA Order Handling* component is responsible for receiving a *NIA Order*, for checking that it is valid and feasible and for provisioning the necessary resources (and the reservations if required).

Third, the *NP Engineering* component is responsible for selecting and provisioning the end-to-end paths that will be used to forward the traffic. It relies on the constraints and network objectives received by the *CPA Order Handling* and *Business-based Network Development* components. The *NP Engineering* component is divided in subcomponents, each responsible for a specific set of functionalities. The *NP Monitoring* component is mainly used to measure the inter-domain traffic matrix, i.e. to determine the volume of each inter-domain flow. The *NP Mapping* component is used to check if there are local intra-domain paths (in existing Network Planes) that can be used to reach an egress router with given constraints. The *NP Resource Availability Checking* component is used to check if there is enough capacity available in a given *Network Plane* for a given traffic flow. The *NP Provisioning and Maintenance* component is used to setup and re-dimension Network Planes.

4.1.4.2 Handling a CPA Order

In order to clarify the role of each functional component, we show in Figure 17 how a *CPA Order* is handled. We will assume that this *CPA Order* contains a latency constraint for outgoing traffic sent from a source network *S* to a destination *D* in a remote INP. The latency of the requested end-to-end path must be lower or equal to *C*. The operation is similar for other kinds of constraints.

As explained earlier, the *CPA Order* is received by the *CPA Order Handling* component which checks that the user (operator/customer) has been granted the access for submitting *CPA Orders*. If so, the validity of the CPA Order is checked. This includes for instance checking that the source network belongs to the local INP. If these verifications succeed, the newly received *CPA Order* triggers the *NP Engineering* component.

Based on the destination prefixes mentioned in the *CPA Order*, the *NP Engineering* component is able to determine the remote INP that must be contacted. The *NP Engineering* component retrieves from the remote INP the list of remote ingress points *{RI}* that allow reaching the destination *D*. It also retrieves the list of local egress points *{LE}* that allow reaching each ingress in the set *{RI}*. Then, it optionally retrieves the volume of the traffic flow from *S* to *D* by invoking the *NP Monitoring* component. It then checks with the *NP Mapping* component if there are local Network Planes suitable for carrying this traffic between the source *S* and each egress *LE*. For each possible Network Plane, it checks if there is enough capacity for the new flow with the help of the *NP Resource Availability Checking* component. It ends up with a list of possible local Network Planes.

The *NP Engineering* component is therefore able to build a list of possible end-to-end paths, based on the local Network Planes and the inter-domain paths from the each *LE* to each *RI*. This set can be pruned from the paths that already do not allow to meet the flow constraint, i.e. only the paths with a latency which is lower than *C* are kept. The *NP Engineering* then runs an optimization process to select the end-to-end paths that will be used to forward the constrained flow, while meeting the other constraints and the global network objectives. The outcome of this optimization is a single path *(S, LE, RI, D)*.

The last task of the NP Engineering component consists in setting up the path and ensuring it carries the flow. This involves two main steps. First, the remote INP must be contacted in order to inform it that the path from *RI* to *D* will be used to forward the given flow. In addition, a reservation might have to be performed in the remote INP. These two steps are delegated to the *NIA Ordering* component. Second, provisioning must be performed in the local INP. The *NP Provisioning* component might have to re-dimension the selected Network Plane. In addition, the *RE* ASBR must be configured so as to serve as a tunnel head-end for the traffic flow. Finally, the PE that connects the customer might have to be configured in order to direct the traffic flow in the selected NP and towards the *RE* ASBR.

4.1.4.3 Handling a NIA Order

In this section, we describe how the remote INP handles a *NIA Order*. We assume that the NIA Order concerns incoming traffic that will be received at an ingress router *LI* and destined to the local network *D*. This scenario is illustrated in Figure 18. This *NIA Order* is received by the *NIA Order Handling* component that will first check if the requestor is allowed and if the destination network belongs to the local INP. If the *NIA Order* is accepted, it is forwarded to the *NP Engineering* component.

The first action then is to check if this request can be mapped to an existing Network Plane. This is the role of the *NP Mapping*. If a suitable Network Plane is found, the *NP Resource Availability Checking* component verifies that the request can be accommodated in this Network Plane. A minimum bandwidth might optionally be provided within the NIA Order. In this case, the *NP Resource Availability Checking* component must check that the requested bandwidth amount can be allocated If there is not enough capacity, the *NP Provisioning and Maintenance* component is triggered in order to try to re-dimension the Network Plane. If the Network Plane cannot be re-dimensioned, the requesting INP is notified. Otherwise, the requesting INP is notified of the success (there is an agreement) and is informed of the parameters to be used (marking).

We have described in the above paragraphs how a *NIA Order* for incoming traffic is handled. It is also possible that a remote INP requests a NIA for traffic going in the reverse direction (outgoing traffic). This kind of *NIA Order* would typically be issued by an INP that wants to balance the load of its incoming traffic. In this case, the *NIA Order* would specify the remote ingress *RI* to be used and the destination *D* in the remote INP. The NIA Order could also specify the local egress *LE* to be used, or it could leave this choice free. We illustrate in Figure 19 a request for a NIA concerning outgoing traffic and where no egress is specified.

The *NIA Order* is handled in the same manner than for incoming traffic by the *NIA Order Handling* component. The *NP Engineering* component is then triggered and it will process the request as follows. First, it will get the list of local egresses *{LE}* that allow to reach the specified *RI*. For each possible LE, it will then obtain from the *NP Mapping* component the list of suitable Network Planes and it will further check that these Network Planes can accommodate the request through the *NP Resource Availability Checking* component or if they can be re-dimensioned (*NP Provisioning and Maintenance*). It ends up with a list of possible local Network Planes.

The *NP Engineering* component is therefore able to build a list of possible paths from the source *S* to the remote ingress *RI*, based on the local Network Planes and the inter-domain paths from the each *LE* to the *RI*. This set can be pruned from the paths that already do not allow to meet the flow constraint (if any). The *NP Engineering* then runs an optimization process to select the end-to-end paths that will be used to forward the constrained flow, while meeting the other constraints and the global network objectives. The outcome of this optimization is a single path *(S, LE, RI)*, or a set of paths if load-balancing over these paths is considered.

The necessary resources are then provisioned (and optionally reserved) in the local INP thanks to the *NP Provisioning and Maintenance* component and the requesting INP is notified of the success.

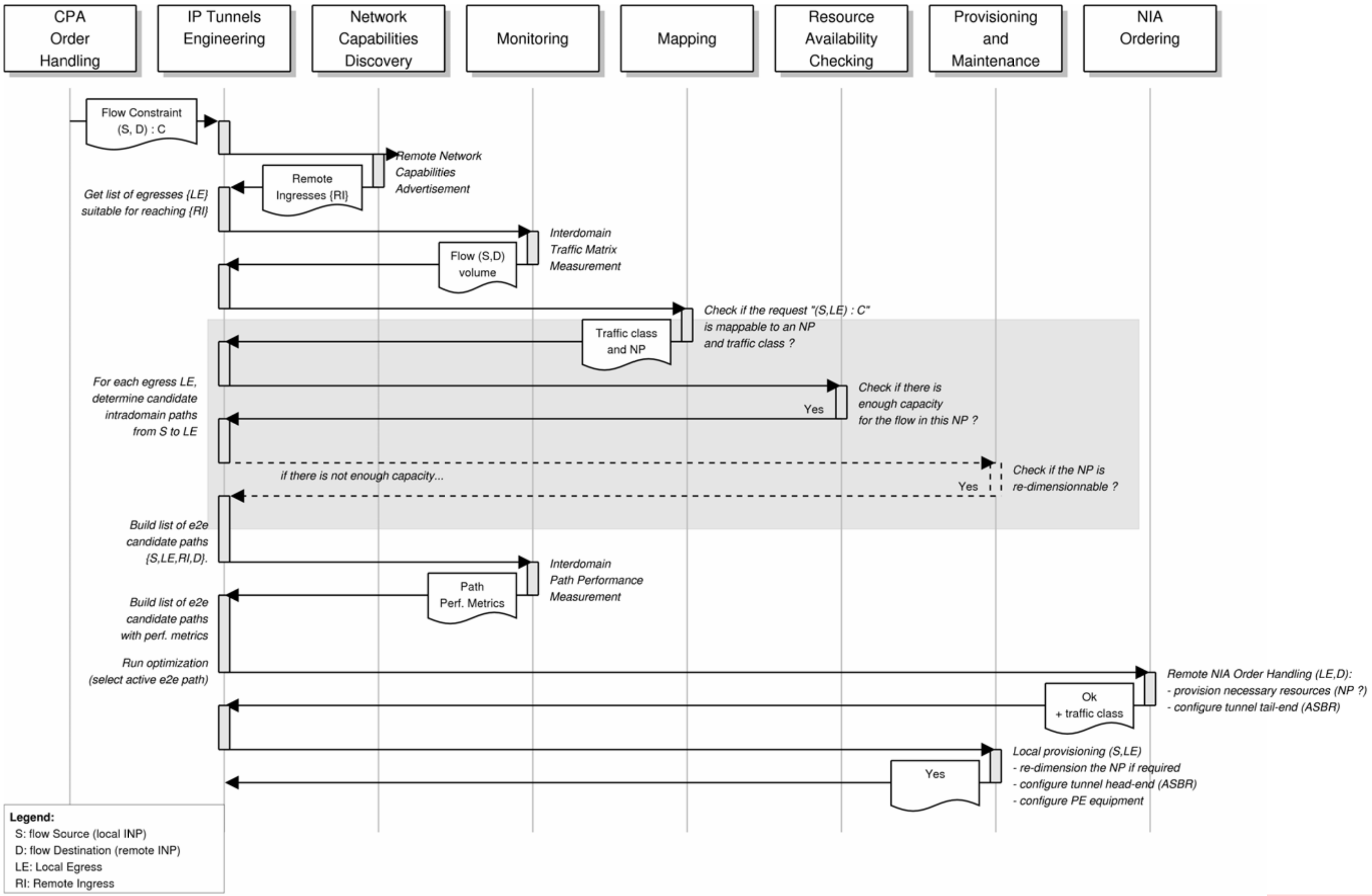

CPA Order Handling
IP Tunnels Engineering
Network Capabilities Discovery
Monitoring
Mapping
Resource Availability Checking
Provisioning and Maintenance
NIA Ordering
Flow Constraint (S, D) : C
Remote Network Capabilities Advertisement
Remote Ingresses {RI}
Get list of egresses {LE} suitable for reaching {RI}
Interdomain Traffic Matrix Measurement
Flow (S,D) volume
Check if the request "(S,LE) : C" is mappable to an NP and traffic class ?
Traffic class and NP
For each egress LE, determine candidate intradomain paths from S to LE
Check if there is enough capacity for the flow in this NP ?
Yes
if there is not enough capacity...
Check if the NP is re-dimensionnable ?
Yes
Build list of e2e candidate paths {S,LE,RI,D}.
Interdomain Path Performance Measurement
Path Perf. Metrics
Build list of e2e candidate paths with perf. metrics
Run optimization (select active e2e path)
Remote NIA Order Handling (LE,D):
- provision necessary resources (NP ?)
- configure tunnel tail-end (ASBR)
Ok + traffic class
Local provisioning (S,LE)
- re-dimension the NP if required
- configure tunnel head-end (ASBR)
- configure PE equipment
Yes
Legend:
S: flow Source (local INP)
D: flow Destination (remote INP)
LE: Local Egress
RI: Remote Ingress

Figure 17 - Flowchart of the handling of a CPA order for outgoing traffic.

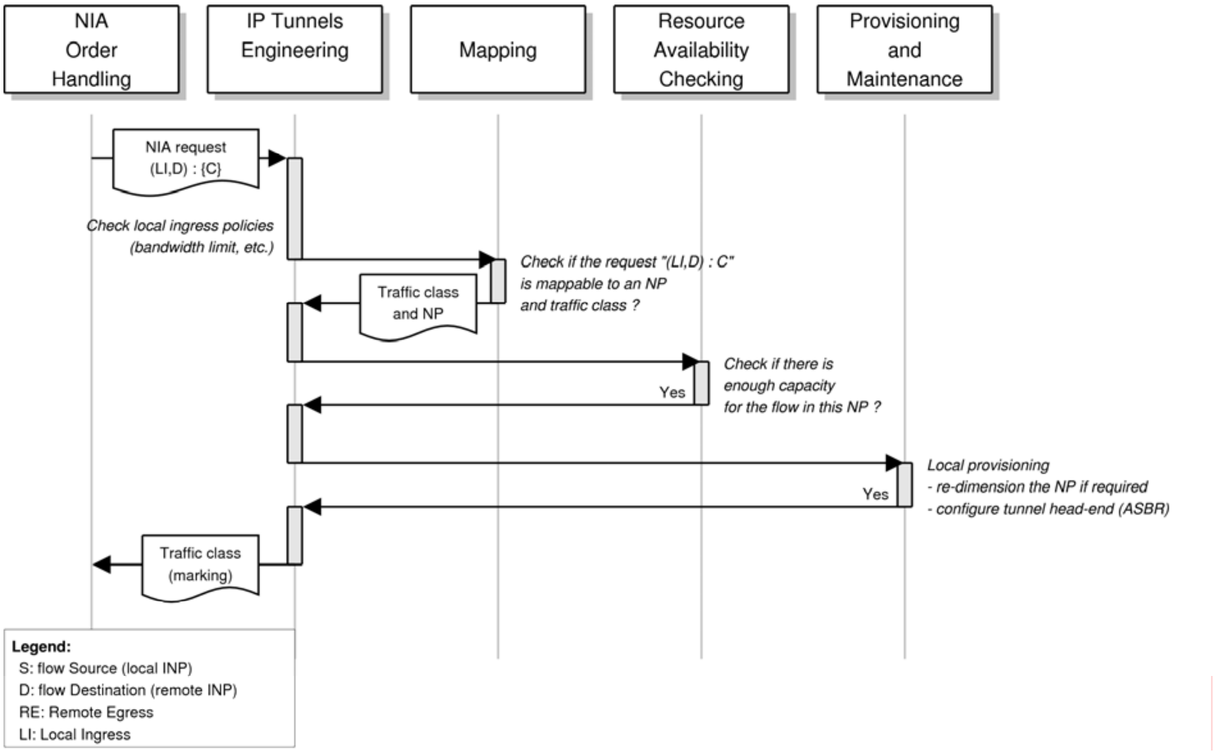

NIA Order Handling
IP Tunnels Engineering
Mapping
Resource Availability Checking
Provisioning and Maintenance
NIA request (LI,D) : {C}
Check local ingress policies (bandwidth limit, etc.)
Check if the request "(LI,D) : C" is mappable to an NP and traffic class ?
Traffic class and NP
Check if there is enough capacity for the flow in this NP ?
Yes
Local provisioning
- re-dimension the NP if required
- configure tunnel head-end (ASBR)
Yes
Traffic class (marking)
Legend:
S: flow Source (local INP)
D: flow Destination (remote INP)
RE: Remote Egress
LI: Local Ingress

Figure 18 - Flowchart of the handling of an NIA order for incoming traffic.

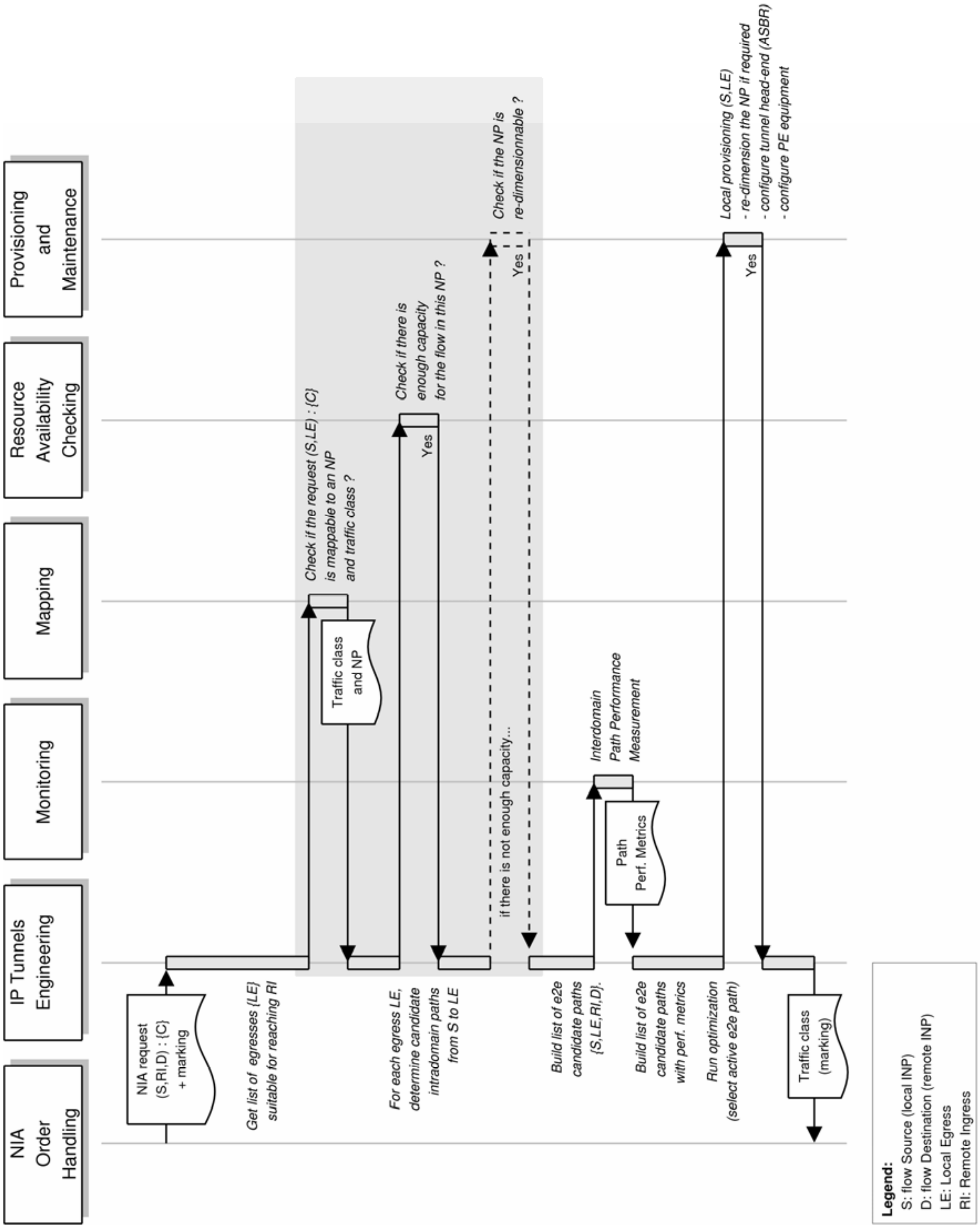

NIA Order Handling
IP Tunnels Engineering
Monitoring
Mapping
Resource Availability Checking
Provisioning and Maintenance
NIA request (S,RI,D) : {C} + marking
Get list of egresses {LE} suitable for reaching RI
Check if the request (S,LE) : {C} is mappable to an NP and traffic class ?
Traffic class and NP
For each egress LE, determine candidate intradomain paths from S to LE
Check if there is enough capacity for the flow in this NP ?
Yes
if there is not enough capacity...
Check if the NP is re-dimensionnable ?
Yes
Build list of e2e candidate paths {S,LE,RI,D}.
Interdomain Path Performance Measurement
Path Perf. Metrics
Build list of e2e candidate paths with perf. metrics
Run optimization (select active e2e path)
Local provisioning (S,LE)
- re-dimension the NP if required
- configure tunnel head-end (ASBR)
- configure PE equipment
Yes
Traffic class (marking)
Legend:
S: flow Source (local INP)
D: flow Destination (remote INP)
LE: Local Egress
RI: Remote Ingress

Figure 19 - Flowchart of the handling of a NIA order for outgoing traffic.

4.1.5 System architecture

The system architecture (see Figure 20) relies on dedicated servers running in the network of each participant, which we call the *Tunneling Service Controllers* (TSC). These TSCs are responsible for discovering the available inter-domain paths, selecting which paths will be used for forwarding the traffic and provisioning the network accordingly. A TSC can be a standalone workstation (such as a PCE [RFC4655] or a RCP [FEAM04]) or it can be run as a different process in the control plane of a router. The TSCs interact with other components such as a global TSC directory that allows each site to advertise its TSCs along with the network they serve and their capabilities. Each site has a configuration and policy database where the users (typically, the network operators) define the constraints they want to put on the traffic flows (CPAs), and the global objectives they want to achieve.

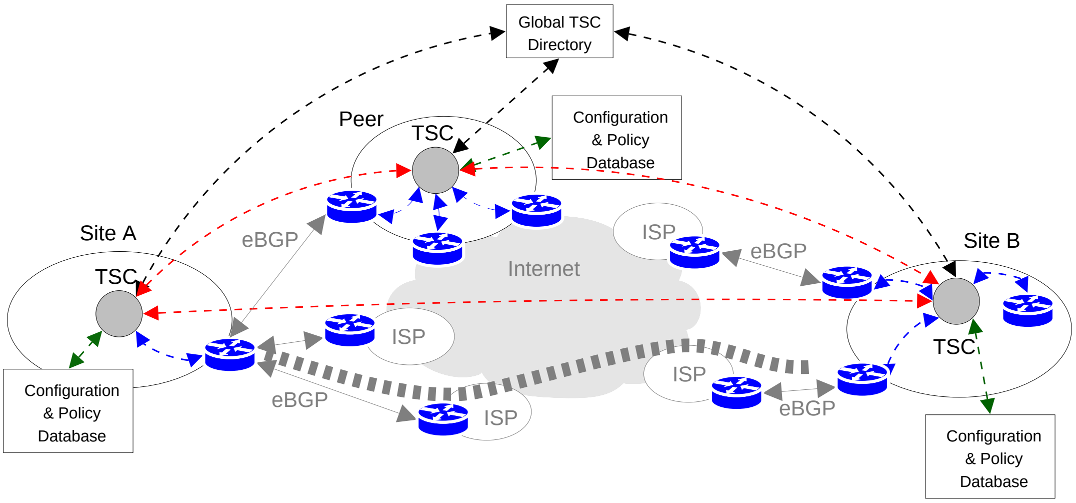

Global TSC Directory
Peer
TSC
Configuration & Policy Database
Site A
TSC
eBGP
Internet
ISP
eBGP
Site B
ISP
TSC
Configuration & Policy Database
eBGP
ISP
ISP
eBGP
Configuration & Policy Database

Figure 20 - Overview of the system components involved in the IP Tunneling solution.

Basically, a TSC starts by discovering the participants that handle networks that are the source or the destination of flows on which the network operators have defined constraints. A TSC might also be interested in contacting those TSCs that are the main sources of its incoming traffic, for the purpose of moving these sources and balancing the load. We describe, in the following paragraphs, the system architecture from the point of view of a single TSC. We show in Figure 21 a graphical representation of the system components and their interactions.

First, the *Policies and Configuration Database* is used to store flow constraints, global network objectives and system configuration parameters such as ingress/egress constraints. By *flow constraints*, we understand a maximum bound on the latency or a minimum bound for the available bandwidth associated to a flow. The availability of an end-to-end backup path is another possible constraint. A flow would typically be identified by source and destination prefixes, source and destination ports, transport protocol and ToS value (note that all “fields” are not mandatory and that “wildcards” could be used). By *global network objectives*, we understand balancing the traffic load across the access links or minimizing the peering cost. The definition of these policies is the role of the network operators, the customer networks operators and the service providers. Receiving, validity checking and storing the flow constraints is the role of the *CPA Order Handling* component. In addition, the *Business-based Network Development* block aims at receiving the network objectives from the network operator, checking their validity and storing them in the database. We envision that the network global objectives and the system configuration are items that can only be modified by the network operators.

Then, the *Network Capabilities Advertisement and Discovery* component is responsible for determining the capabilities of the remote INPs. It is divided in two parts. First, the *TSC Advertisement and Discovery* component is responsible for advertising to other domains the TSCs available in the local domain. This component is also responsible for discovering the TSCs in remote domains. Note that TSCs need not to communicate directly in order to discover themselves ***(1)***. Second, the *Ingresses Advertisement and Discovery* component is responsible for requesting from remote domains (through their TSCs) the list of their ingress points and their capabilities. This block is also responsible for answering the requests from remote domains. This block relies on a communication between the TSCs ***(2)***. The list of ingresses that can be announced to other domains and the parameters of these ingresses are found in the network *Policies and Configuration* database ***(3)***.

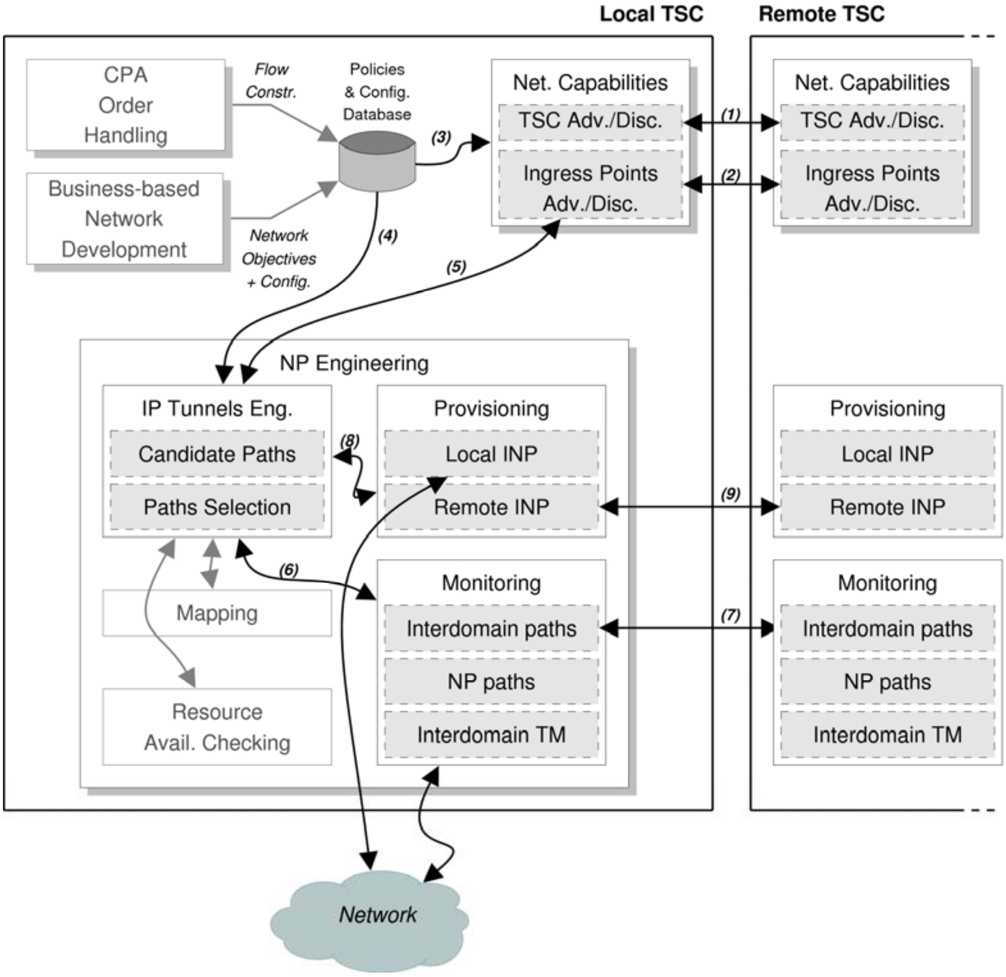


Figure 21 - Detailed view of the TSC system components and their interactions.

The *Monitoring* component is responsible for measuring the inter-domain traffic matrix and the paths performances. The inter-domain traffic matrix contains the amount of traffic that crosses the INP boundaries. In particular, it contains the amount of traffic that belongs to each flow specified in the *Policies and Configuration Database*. The paths performance measurement is done separately for the intra-domain and inter-domain parts of the paths. The inter-domain paths performance measurement requires exchanges of information with the remote TSC ***(7)***.

This is the responsibility of the *IP Tunnels Engineering* component to select the end-to-end paths that will be used to forward the traffic. For this purpose, this block relies on the constraints and network objectives found in the *Policies and Configuration Database* ***(4)***. Depending on the flow constraints and on the network objectives, the *IP Tunnels Engineering* component will need to contact one or more TSCs in remote domains in order to find alternative inter-domain paths (this is done through the *Network Capabilities Advertisement and Discovery* component) ***(5)***. These inter-domain paths are then combined with the local (intra-domain) paths to form candidate end-to-end paths. If required, the component will also trigger measurements of the inter-domain and intra-domain parts of the paths performed by the *Monitoring* component ***(6)***. The *IP Tunnels Engineering* component also obtains the inter-domain traffic matrix from the *Monitoring* component ***(6)***. Once end-to-end paths have been selected, the *IP Tunnels Engineering* notifies the *Provisioning* component which is responsible for “implementing” the paths ***(8)***.

Finally, the *Provisioning* block is used to setup the networking equipment according to the selection of paths provided by the *IP Tunnels Engineering* component. The setup of the local intra-domain path (from the source to the egress) is performed depending on the local INP Network Plane implementation. The remote intra-domain part of the selected path (from the ingress to the destination) and the configuration of the tunnel tail-end are negotiated with the remote domain ***(9)***. When the requested resources are available, the remote domain answers with a confirmation and the required parameters (for instance, the DSCP value to be used to enter the requested Network Plane in the remote domain). This confirmation message is the implementation of the *Network Interface Agreement* (NIA) described in the functional architecture. This part of the *Provisioning* component is also responsible for handling the requests issued by the remote domain ***(9)***, for checking if the requests can be fulfilled by contacting the *NP Engineering* block (*Resource Availability Checking*) and for later committing the changes requested by the remote TSC.

We describe each system component in more details in the following sections. We will not further detail the following blocks: *CPA Order Handling*, *Business-based Network Development*, *NP Mapping* and the *NP Resource Availability Checking*.

4.1.5.1 Policies and Configuration Database

The purpose of the *Policies and Configuration Database* is to store flow constraints, network objectives and system configuration parameters such as ingress/egress constraints. We describe in more depth the information required for each database element. We do not define in this document how the constraints and the configuration should be defined by the user (operator/customer). We will assume that these policies are available and stored in a database accessible by the various TSCs. Standards or proposed standards such as COPS [RFC2748], PIBs [RFC3318] and DEN-ng [STRA03] might be envisioned for specifying these policies. This is however out of the scope of the IP Tunneling solution specification. For this reason, we will not specify in this section how this database is structured, but only define what information it should contain. For the same reason, we will not specify the *CPA Order Handling* and *Business-based Network Development* blocks which are responsible for filling this database.

4.1.5.1.1 Flow Identification

In order to specify traffic flow constraints, one needs to be able to identify a particular flow. This is the aim of the *Flow Identification*. A traffic flow will typically be specified based on its *source prefix*, *destination prefix*, *source port*, *destination port, transport protocol (TCP/UDP)* and *ToS* value [RFC2475]. For the purpose of readability, we will only specify flows with the source and destination prefixes and note a *Flow Identification* as *(s,d)*.

In a *Flow Identification*, all fields are not mandatory. We can imagine that fields for which there is no restriction can be filled with a "wildcard", meaning that they would match any value. For example, the source port in a *Flow Identification* containing multiple source addresses should probably not be mentioned since it is very unlikely that the source port of two different source hosts will be equal.

4.1.5.1.2 Flow Constraints

A *Flow Constraint* is an association between a *Flow Identification* and a traffic constraint. Typical traffic constraints are defined in terms of latency, bandwidth or resilience. We will note these objectives as shown in Table 1. If a *Latency Constraint* is associated to a flow that means that the given flow must not be subject to latency higher than the specified threshold *D*. Similarly, a *Bandwidth Constraint* mentions that the given flow must have an available bandwidth at least equal to *B*.

In addition to pure performance constraints, it is also possible to associate a flow with a resilience constraint. In that case, in addition to a path that satisfies the performance constraints associated to the flow, the system must provision a backup path that can also satisfy these constraints. Note: the backup path should be disjoint from the primary path at least on the access links of both domains. It should also be possible to make them disjoint inside each domain. Finally, if possible the paths could additionally be made disjoint at the AS-Path level.

Latency constraint	$d_{s,d} \leq D$
Bandwidth constraint	$b_{s,d} \geq B$
Backup	*End-2-end / Hot standby*

Table 1: Summary of typical Flow Constraints.

4.1.5.1.3 Global Network Objectives

In addition to *Flow Constraints*, the database must allow to store *Global Network Objectives*. These objectives specify what must be globally optimized in the network. Typical objectives are summarized in Table 2. Here, *T* and *R* denote the inter-domain traffic and routing matrices respectively. The *T* matrix gives the amount of traffic sent between two prefixes for a given class of traffic while *R* indicates the inter-domain path (and therefore the peering link) used by each prefix pair. We assume that the cost $c_k(T,R)$ of a peering *k* can be computed based on these matrices. Similarly, we assume that the traffic load $l_k(T,R)$ of a peering *k* can be computed based on these matrices. C_k represents the capacity of the peering link *k*.

Peering Cost Minimization	$\min\left(\sum_k c_k(T,R)\right)$
Load-Balancing	$\min\left(\sum_k \left\lvert\frac{l_k(T,R)}{C_k} - 1\right\rvert\right)$

Table 2: Summary of possible global objectives.

Note: The database must also contain the specification of the peering cost functions, $c_k(T,R)$.

4.1.5.1.4 Access link constraints

In addition to the above policies (*Flow Constraints* and *Network Objectives*), the database is also used to store the system configuration. This section defines the policies and parameters related to the utilization of the local access links by other participants (summarized in Table 3). These policies should cover the availability of access links for other participants (controlled by access lists), bandwidth limits for each access links, relative preferences, and so on. The configuration parameters cover the supported tunneling mechanisms of each access interface (along with the tunneling parameters).

Access list	List of ASes that are allowed or denied the service through this access link.
Bandwidth limit	Fraction of bandwidth that can be used (reserved) by remote domains.
Traffic Conformance Policies	Shaping/policing [optional]

Supported Tunneling Mechanisms (and parameters)	Depends on the available hardware on ASBRs. Can be inferred from network inventory.
Preference	

Table 3: Summary of access link constraints.

4.1.5.1.5 System Access Constraints

In order to control which users are allowed to change the *Policies and Configuration Database*, this database should also contain user access lists. A typical configuration would allow the network operators to define the network global objectives and the system configuration (access link constraints and system access constraints), while the customer network operators and service providers would only be allowed to specify *Flow Constraints*. In addition, it would be interesting to prevent a customer to associate *Flow Constraints* with a *Flow Identification* where the customer owns neither the source and destination prefixes.

This implies that the *Policies and Configuration Database* should contain a list of users with their roles and credentials. Specifying such a database is out of scope.

4.1.5.2 Network Capabilities Advertisement and Discovery

The *Network Capabilities Advertisement and Discovery* functional component is responsible for discovering the ingress points of remote INPs along with the capabilities of these ingresses. This functionality can be further refined in two parts.

The first part consists in discovering the INPs that are willing to cooperate and offer the tunneling service to other INPs. It is also important to be able to find which remote networks are managed by which cooperating INP. Finally, it is also required to discover the TSCs used by the remote INPs in order to contact them. We describe in section 4.1.5.2.1 how the TSCs advertisement and discovery are performed.

The second part consists in discovering the ingress points of a remote INP, the network reachable through these ingresses and their parameters. This discovery is done by talking with one of the TSCs of the remote INP. We describe in section 4.1.5.2.2 what information is provided by a TSC about the ingress points of its domain.

4.1.5.2.1 TSCs Advertisement and Discovery

The aim of the *TSCs Advertisement and Discovery* function is to allow each participant domain to advertise its TSCs and their capabilities. We envision that this can be done in two manners. The first manner corresponds to what is done today, i.e. a preliminary business agreement is made between two INPs and the configuration (IP addresses of the TSCs, security parameters, etc) are exchanged manually by means of letter, phone call or e-mail. This information could also be stored in the Routing Assets Databases (RADb) maintained by the routing registries such as ARIN, RIPE and APNIC. However, this is a slow process and it can therefore be difficult to manually arrange lots of contracts in case the number of cooperating participants is high.

We propose a more dynamic automatic discovery of the participants by mean of a public global TSC directory service. Using a directory service for private use among a close group is just a particular case. In this directory, each participant will **register** the networks it owns. Each network is specified as a list of IP prefixes called a domain. It can be conceived that the size of these networks can range from a single IP address (of a particular server for instance) to the whole participant's IP space. A participant should be able to register more than one TSC per domain for robustness and load-balancing purposes.

On the other hand, a domain should also be able to **find** in the directory which domains participate and to **locate** the TSC(s) responsible for a particular domain.

A suitable, existing directory service is the Domain Name System (DNS) [RFC1035]. In this case, groups of IP addresses would typically be identified by a domain name. The main advantage of the DNS is its distributed nature which provides robustness to the failure of a single DNS server. A second advantage of the DNS is its wide deployment. Any network in the world that is connected to the Internet uses the DNS. In addition, the DNS can easily be extended: the DNS can store new resource records (RR) and it also supports various security-related resource records. It is for instance possible to associate a public key or a certificate with each TSC.

4.1.5.2.1.1 TSC Advertisement

One possible way for a participant to advertise in the DNS the TSCs of its own domain would be to rely on the DNS SRV resource record [RFC2782]. This resource record allows specifying the location of well-known services. A typical use of this RR is for advertising the SIP servers of a domain. The SRV RR allows to mention a service (identified by a well-known symbolic name), along with the underlying protocol's symbolic name and the domain it is running in.

The example shown below advertises two TSCs running in *domaina.net*. The first two lines specify classical DNS Aliases *tsc1* for host 10.0.0.1 and *tsc2* for 10.0.0.2 while the last two lines specify that a TSC is running on *tsc1* on port 1024, using TCP and another one on *tsc2*. The SRV specification allows defining a ranking among the different servers, using a priority field. A client must attempt to contact the lowest priority server first. In the example, the priority is 0 for both servers. In addition, a relative weight can be assigned to each server for load-balancing purpose. In the example, server *tsc1* should be sent 75% of the requests while server *tsc2* should be sent the remaining 25% of the requests.

```
tsc1.domaina.net. 86400 IN A 10.0.0.1
tsc2.domaina.net. 86400 IN A 10.0.0.2
_tsc._tcp.domaina.net 86400 IN SRV 0 75 1024 tsc1.domaina.net
_tsc._tcp.domaina.net 86400 IN SRV 0 25 1024 tsc2.domaina.net
```

Table 4: Example DNS SRV resource records identifying the TSCs in domaina.net.

Usually, the resource records for one domain (or zone) are defined in a statically configured file stored on the DNS server. At the time the DNS was designed, the zone files were not expected to change frequently. A TSC that needs to advertise itself through the DNS would need to change that zone file on its local DNS server. Fortunately, there is a standard way to dynamically update the resource records of a zone by using the DNS UPDATE message [RFC2136], an extension to the classical DNS protocol [RFC1035]. This message allows adding or deleting RRs from a specified zone. A TSC that needs to advertise itself for a domain would issue a DNS UPDATE message to the authoritative DNS server of the zone requesting to add the SRV resource record. The security of the DNS UPDATE messages is tackled by the DNSSEC protocol extensions [RFC4033, RFC4034, RFC4035].

4.1.5.2.2 Ingresses Advertisement and Discovery

The second functional block, named Ingress Advertisement and Discovery, is responsible for (1) advertising the local ingress points and their constraints/capabilities and (2) discovering the ingress points of a remote participant, i.e. possible tunnel tail-ends. The two functions (advertisement and discovery) rely on a communication between the TSCs in the involved domains. They exchange information about their respective ingress points through an *Ingress Information List* (IIL), composed of one *Ingress Information Item* (III) for each ingress point.

4.1.5.2.2.1 Ingress Information Item

Basically, an ingress point is the interface of a router that has a public, routable IP address. The typical information provided by a TSC for an ingress point would contain the following elements (as shown in Table 5). First, the *set of authorized interfaces* of the ingress point. Each interface is identified by an IP address. Second, each interface comes with a *Preference* level, an integer value administratively assigned by the TSC operator. The interfaces with the highest preference value are preferred and should be used by the requestor if possible. Third, a *Bandwidth Limit* is provided for each interface. The Bandwidth Limit specifies what amount of bandwidth the client is allowed to send through this interface. In addition to these parameters, the supported tunneling mechanisms supported by each interface are advertised by means of a *Tunnel Specification List* (TSL). Finally, the III also contains an enumeration of the IP domains reachable through this ingress, under the form of a *Reachability Information List* (RIL).

Field	***Description***	***Optional***
Interface	IP address of the ingress interface	
Preference	Preference ranking of this ingress.	X
Bandwidth Limit	Maximum bandwidth that can be allocated on this ingress.	X
Tunnel Specification List (TSL)	List of *Tunnel Specification Items* (TSIs) see Table 6	
Reachability Information List (RIL)	List of *Reachability Information Items* (RIIs)	

Table 5: Ingress Information Item (III).

A TSL is a list of *Tunnel Specification Items* (TSI). A TSI contains the following information. The first element is an identification of the tunneling mechanism. Examples of such mechanisms are IP-in-IP [RFC1853], GRE [RFC2784], L2TPv3 [RFC3931] or IPsec [RFC1825, FRAN01, DORA99]. Each tunneling mechanism has its own set of parameters that can be specified in a mechanism-dependant information structure as shown in Table 6.

The tunneling mechanisms listed above mainly differ by the security level they can provide. The most important threat for tunnel-based frameworks is probably packet spoofing. Indeed, the insertion of carefully forged malicious packets into the tunnel could allow an attacker to conduct DDoS attacks with packets that would appear as originated by the tunnel tail-end. Most tunneling mechanisms, except the simple IP-in-IP protocol, support means of quickly discarding forged packets. With GRE, the insertion of a *Key* field agreed between the two parties provides light protection against spoofing. With L2TPv3, the tunnel head-end must add a *Session ID* field chosen by the tail-end. In addition, a *Cookie* field might also be used. The value of the *Cookie* is agreed during the tunnel session establishment and is checked by the receiver of data packets. If the attacker is able to eavesdrop the tunnel traffic, it will be easy for him to forge packets with correct *Key* field for GRE tunnels or correct *Session ID* and *Cookie* in the case of L2TPv3 tunnels.

IPsec in tunnel mode relies on cryptography to provide higher security levels. There are two flavors of IPsec. The first one uses the *Authentication Header* (AH) and only offers authentication, integrity protection and an optional anti-replay protection. The second flavor uses the *Encapsulated Security Payload* (ESP) and can provide confidentiality in addition to authentication. In tunnel mode, the ESP also protects against traffic analysis in the sense that an eavesdropper cannot see which hosts are communicating. These increased security levels of course come at a higher computational cost. The establishment of a *security association* (SA) between two IPsec peers is performed by the *Internet Key Exchange* (IKE) protocol. Different IKE authentication methods are possible. The first one relies on a pre-shared secret key. A shared-secret has obvious drawbacks such as the need for exchanging it in a secure manner. The other IKE authentication methods rely on digital signatures or public key encryption. Using public key encryption requires a *Public Key Infrastructure* (PKI) to retrieve the certificates of the peers. One possibility for authenticating the TSCs would be to rely on certificates derived from those that will probably be issued in the near future by the IANA and the routing registries (RIPE, ARIN, APNIC…). Such certificates would bind one TSC to its AS and possibly the network prefixes it owns. See Section 4.1.5.2.2.2 for a more detailed discussion on these certificates.

If there is an agreement on what source will send traffic into a tunnel, this allows the automatic installation of IP address-based filters on the tunnel tail-end. In addition, the tunnel tail-end might also filter based on the destination address of the encapsulated packets. Packets that are not destined to a network advertised as reachable from this router are discarded. Note that such access control mechanisms are not part of the specification of IP-in-IP, GRE or L2TPv3. In the case of IPsec, there is preliminary work on a Security Policy Database (SPD), but nothing standardized yet.

Mechanism	***TSI parameters***	***Optional***
IP-in-IP	No additional parameters.	
GRE	An optional 4-bytes <u>*Key*</u> field can be used by the tail-end to authenticate the source of the packet.	X
L2TPv3	Optional <u>*Shared Secret*</u> (password) for authenticating and checking the integrity of control messages.	X
	Mandatory <u>*Session ID*</u>: a 4-bytes value (non-zero) identifying an L2TPv3 session.	
	Optional <u>*Cookie*</u> value used by L2TPv3 to check the association of a received data message with the session (identified by the above *Session ID*). The *Cookie* length can vary and be as long as 64bits.	X
IPSec (tunnel mode)	Flavor: AH (authentication only) or ESP (encryption + authentication).	
	Optional anti-replay protection.	X

Table 6: Parameters in the Tunneling Specification Item (TSI).

The *Reachability Information Item* (RII) contains an identification of the reachable IP space (under the form of an IP prefix). In addition, it may also contain an indication of the possible QoS guarantees (latency or available bandwidth) and resilience options. The real performance guarantees are only available after an NIA is obtained from the remote domain (see *Network Provisioning* block). This indication allows the Path Selection algorithm to early discard ingresses that will not allow fulfilling a set of flow constraints.

Field	***Description***	***Optional***
Reachable domain	Range of IP addresses	

Latency	Latency (measured or statically defined) from the ingress point to this domain	X
Bandwidth	Available bandwidth (measured or statically defined)	X
Resilience	Protection available (fast-reroute or end-to-end backup)	X

Table 7: Reachability Information Item (RII).

4.1.5.2.2.2 Communication Protocol

In order to get the list of ingresses of a remote participant and the related TSLs, a management protocols is needed. This protocol should mainly support two messages: an *Ingress Points Specification Request* message and the corresponding *Ingress Points Specification Response* message.

We propose to implement this request/response protocol would be to rely on *Web Services* [BOOT04], an interoperable messaging framework which can rely on the *Simple Object Access Protocol* (SOAP) [GUDG03] for performing RPCs into XML and carry these RPCs over HTTP. Note that this is also the communication means envisioned by the IPSphere consortium [IPSP06].

The main threat for the Inter-TSC communications is to allow a non authorized peer to access or request resources. Therefore, authentication is needed. This authentication can be provided by an SSL layer used under SOAP/HTTP. Using SSL would require a PKI for storing and retrieving certificates. An alternative could be to store public keys or certificates for each TSC in the DNS in addition to the SRV RRs. This would make possible for participants to authenticate each other. For example, the DNS CERT RR [RFC4398] allows storing certificates and certificates revocation lists in the DNS.

Another possibility could be to rely on *Resource Public Key Certificates* (RPKC) currently discussed at the IETF in the SIDR working group. One RPKC certifies that the certificate's subject is the current controller of a collection of IP addresses and AS resources (listed in the certificate's resource extension). This is based on X.509v3 certificates [RFC3280] and IP addresses and AS Number extensions [RFC3779]. Typically, IANA would issue certificates for large IP blocks to main Internet registries (ARIN, RIPE, APNIC, ...) that would subsequently delegate parts of these blocks to local registries and finally to Internet Service Providers. Validating the certificate of an ISP would therefore consist in finding a validated certification path from the ISP to IANA that would serve as a trusted authority.

4.1.5.3 Monitoring

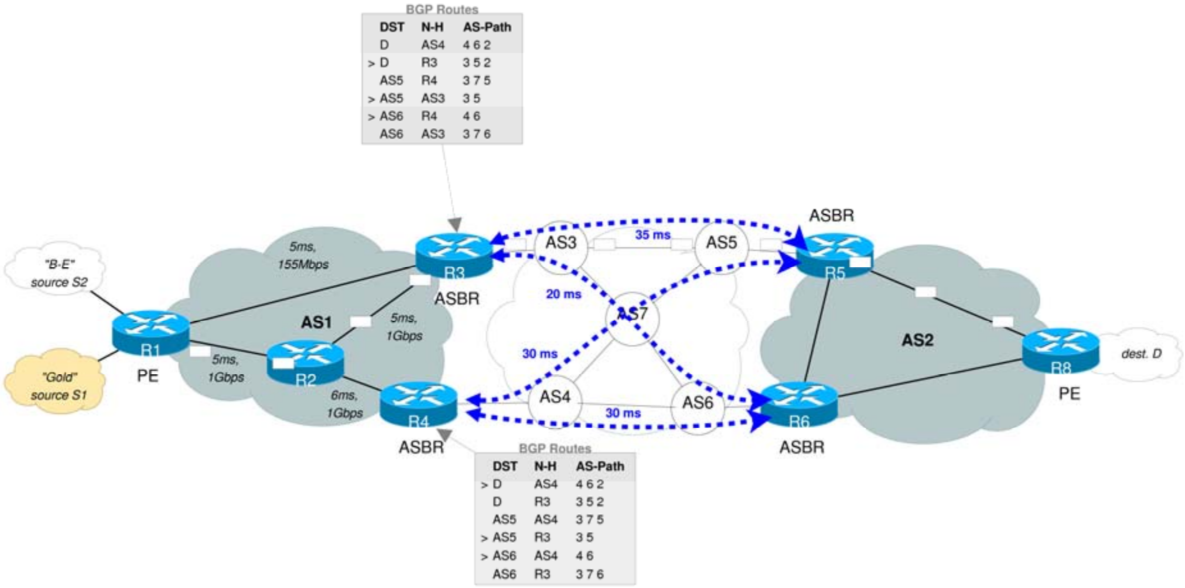
The purpose of the *Monitoring* block is to gather the network performance metric required for the paths selection algorithm. The roles of the *Monitoring* block are as follows. First, it must be able to measure the performance of the candidate end-to-end paths in order to compare them and to monitor the performance of the selected end-to-end paths in order to detect performance degradation. The metrics that need to be reported depend on the constraints that are put on the flows that would be forwarded along the paths. If there is a latency constraint, the latency of the paths must be measured. The end-to-end paths are composed of two different parts: an intra-domain part (one in the local INP and another one in the remote INP) and an inter-domain part. The intra-domain part is a path between the source or destination network and an ASBR. The inter-domain part is a path between two remote ASBRs. We detail the intra-domain paths performance measurement in Section 4.1.5.3.1 and the inter-domain paths performance measurement in Section 4.1.5.3.2.

The second role of the Monitoring component is the measurement of the inter-domain traffic matrix. We describe this part in Section 4.1.5.3.3.

4.1.5.3.1 Intra-domain Paths Performance Measurement

The intra-domain paths performance measurement consists in measuring the performance of the intra-domain part of candidate end-to-end paths in order for the paths selection algorithm to be able to compare them. By intra-domain part of the paths, we understand the paths between the local source or destination network, or the PE router to which it is connected, and a prospective egress/ingress router. The performance metrics that are required depend on the constraints that are put on the flows. Two main performance metrics should be supported: the latency (one-way delay) and the available bandwidth.

For example, in the topology shown in Figure 22, *AS1* would like to setup a path with better latency from the source network *S1* to the destination network located in *AS2*. *AS1* must be able to determine the performance of the intra-domain path from the PE router *R1* to each ASBR that can reach the remote ingress points. In this case, the remote ingresses of *AS2* are *R5* and *R6* and they are both reachable from *R3* and *R4* (they both have BGP routes for the prefixes containing *R5* ad *R6*). Therefore, the paths from *R1* to *R3* and the paths from *R1* to *R4* need to be measured. The performance metrics could be obtained by performing active probing in *AS1* [CISC04]. However, in case the intra-domain path to be measured belongs to a Network Plane offering strict performance guarantees, it is not necessary to perform measurement since the measured performance will be at least as good as the performance guarantees.



BGP Routes

	DST	N-H	AS-Path
	D	AS4	4 6 2
>	D	R3	3 5 2
	AS5	R4	3 7 5
>	AS5	AS3	3 5
>	AS6	R4	4 6
	AS6	AS3	3 7 6

BGP Routes

	DST	N-H	AS-Path
>	D	AS4	4 6 2
	D	R3	3 5 2
	AS5	AS4	3 7 5
>	AS5	R3	3 5
>	AS6	AS4	4 6
	AS6	R3	3 7 6

Figure 22 - Inter-domain paths performance measurement.

We assume that such path performance measurements are offered by the Network Plane engineering technique supported by each INP. Indeed, the methods that can be used for performing the performance measurements will depend on the Network Planes implementation.

4.1.5.3.2 Inter-domain Paths Performance Measurement

Measuring the performance of the inter-domain part of the paths is more difficult since equipment that we do not control is crossed by these paths. This poses problem for measuring the one-way delay for example. It is not possible to rely on *Round-Trip Time* (RTT) measurement since routing can be asymmetric and the probe packets could follow a different path. Therefore, such measurement requires the cooperation of the tail-end of the path. In this framework, we can assume that at least the remote site equipments are eager to cooperate for performing the measurements.

In addition to this, measuring the inter-domain paths that are not currently selected by the routing protocols might require configuration changes such as the establishment of tunnels, or the cooperation of the path endpoints. For example, in the case of Figure 22, it is possible to measure the best BGP routes towards the remote ingresses *R5* and *R6*. The AS-Paths of these routes are *(3 5)* and *(4 6)*. They are selected by *R3* and *R4* respectively. Measuring the other available (but not selected) routes *(3 7 6)* and *(4 7 5)* requires the ability to force probe packets to exit through an interface that might apparently have no route to reach the destination.

A lot of techniques are available for measuring the paths performance. The IETF IPPM Working Group has standardized an active probing technique for measuring the one-way delay: the One-way Delay Measurement Protocol (OWAMP) [RFC4656, KALI00]. It is also be possible to rely on the use of synthetic coordinates for predicting the latency based on a limited number of measurements [DABE04, LAUN05b]. For what concerns the available bandwidth measurement, there are also a lot of proposals such as PathChar [DOWN99], Sprobe [SARO02], Nettimer [LAI01], Pathload [JAIN02, JAIN02b]. To our knowledge, nothing has been standardized yet by the IETF IPPM Working Group concerning the available bandwidth measurement.

Finally, it is required to quickly detect when an active path, i.e. currently used to carry traffic, is broken. The IETF has standardized the *Bidirectional Forwarding Detection* (BFD) mechanism [KATZ06]. This measurement will take place during the second monitoring phase.

The comparison and the selection of performance measurement mechanisms is out of scope for the AGAVE project.

4.1.5.3.3 Inter-domain Traffic Matrix Measurement

We are interested in measuring the inter-domain traffic, i.e. the prefix-prefix matrix. One solution consists in relying on Netflow statistics [SOMM02] collected on the border routers. Collecting such statistics might still be an operational issue today for two main reasons [VARG04]. First, the size of a prefix-prefix matrix is significantly larger than a router-router matrix. The number of source and destination prefixes is on the order of 180,000 [HUST06]. Second, activating Netflow can put an important burden on the border routers. Finally, setting up such a measurement infrastructure requires a significant investment in configuration time and equipment. Consequently, Netflow will usually only be activated on the peering interfaces that carry a significant fraction of the traffic. In addition, Netflow sampling [CHOI05] is also used in order to decrease the volume of the collected statistics.

Since we focus on stub networks, we assume that there will be few border routers where Netflow measurements must be activated.

4.1.5.4 Paths Selection

The objective of the *Paths Selection* component is to select among a set of candidate paths a set of paths that best comply with the flow constraints and the global network objectives found in the Policies and Configuration database.

4.1.5.4.1 Building the Candidate Paths List

The first task of this block is to combine the list of candidate inter-domain paths obtained from the *Ingresses Discovery* function block with the local network configuration to build a set of possible paths. The local network configuration includes the possible egress points that can reach the remote ingresses and the intra-domain paths from the flow sources to the egresses. This combination step should take into account the tunneling mechanisms supported by the ingress and egress routers.

1) Gather from the ASBRs the routes available for reaching the discovered ingress points in the remote domain. This is typically done by examining the BGP routes available in each ASBR [BLUN06, SCUD05]. In the example of Figure 22, the ingresses advertised by *AS2* are *R5* and *R6*. By looking at the BGP routing tables of AS1's border routers, *R3* and *R4*, it is possible to determine that there are 4 possible inter-domain paths for reaching *AS2*. There are two paths for reaching *AS2* by *R5* which is

in the prefix advertised by *AS5* and two paths for reaching *AS2* by *R6* (in *AS6*). We also learn from the BGP routing tables that their AS-Paths are *(3 5)*, *(4 6)*, *(3 7 6)* and *(4 7 5)*. Note that we take into account all the BGP paths even those that are not currently selected for forwarding by the BGP routers in *AS1*. There are usually a large number of alternative paths available between multi-homed stubs [LAUN05, QUOI06].

2) Gather information from the *NP Engineering* block (*NP Mapping*, *Resource Availability Checking* and *Provisioning* sub-blocks) about the intra-domain paths available from the local prefix and the possible egresses discovered in step (1). In the example of Figure 22, the possible egresses are *R3* and *R4*. We have to look at the paths available between the PE router *R1* that connects the source network S1 and each egress. The number of available intra-domain paths will depend on the techniques deployed for providing Network Planes inside the local INP. For example, if M-ISIS [PRZY06] is deployed with two different virtual topologies, one for best-effort and one that minimizes the delay. We will have one path in each virtual topology for reaching each egress. That makes 4 different paths (Figure 24 illustrates this situation).

3) Build a list of candidate end-2-end paths by combining the intra-domain paths (or NPs) obtained in step (2), the intra-domain paths obtained in step (1) and the remote intra-domain paths advertised by the remote INP (see Section 4.1.5.2.2.1). For the example shown in Figure 22, and assuming that the local INP has deployed M-ISIS with 2 virtual topologies, the list of candidate paths would be as shown in .

Path	*Local intra.*	*Inter-domain*	*Remote intra.*
1	R1→R3 (TOS=0)	R3→R5	R5→R8
2	R1→R3 (TOS=1)	R3→R5	R5→R8
3	R1→R3 (TOS=0)	R3→R6	R6→R8
4	R1→R3 (TOS=1)	R3→R6	R6→R8
5	R1→R4 (TOS=0)	R4→R5	R5→R8
6	R1→R4 (TOS=1)	R4→R5	R5→R8
7	R1→R4 (TOS=0)	R4→R6	R6→R8
8	R1→R4 (TOS=1)	R4→R6	R6→R8

Table 8 - List of candidate end-to-end paths.

4.1.5.4.2 Paths Selection

The Paths Selection algorithm is responsible for selecting the best paths among the candidate end-to-end paths obtained in Section 4.1.5.4.1. The best paths are the paths that fulfill the individual flow constraints and the global network objectives. This is a multi-objective optimization problem. One possible way to solve such a problem is to rely on evolutionary computing [DEB01]. Examples of such algorithms have already been used for solving traffic engineering problems [UHLI03] and load-balancing [QUOI05, QUOI06].

A first heuristic might be as follows:

1. Initialize solution: for each constrained flow, assign the flow to a path that satisfies the constraint. If the flow cannot be assigned a path, report the operator that the constraint cannot be satisfied.
2. Optimize the solution: build a population that contains the initial solution as well as mutations

of the original solution obtained by shifting a flow to alternate paths that satisfy the flow constraints. Evolve the population by further mutating the solutions. Measure the solution with an objective function expressing the global objectives (load-balancing, cost-minimization). Put pressure in the objective function in favor of solutions involving the least number of path changes, the least number of tunnels to establish.

The Paths Selection algorithm must avoid oscillations that could be caused by resource rushes. A possible method would be to rely on a hysteresis such as in the RON framework [ANDE02]. Other methods for avoiding oscillations in Intelligent Route Control systems (IRC) were studied by Ruomei Gao, Dovrolis, Zegura [GAO06]. Their paper indicates that it is required (1) to take into account the impact of shifting traffic on available bandwidth measurement and (2) to de-synchronize measurements performed by different IRCs by, for example, introducing random delays between measurements.

The Paths Selection algorithm must also allow segregating from the global optimization the flows for which individual constraints were defined. That means that a flow which must be forwarded along the lowest latency would not be taken into account for the load-balancing objective or for the cost minimization objective. This is necessary if one wants to allow forwarding traffic along a path with a lower latency, even at the expense of increasing the peering cost for instance.

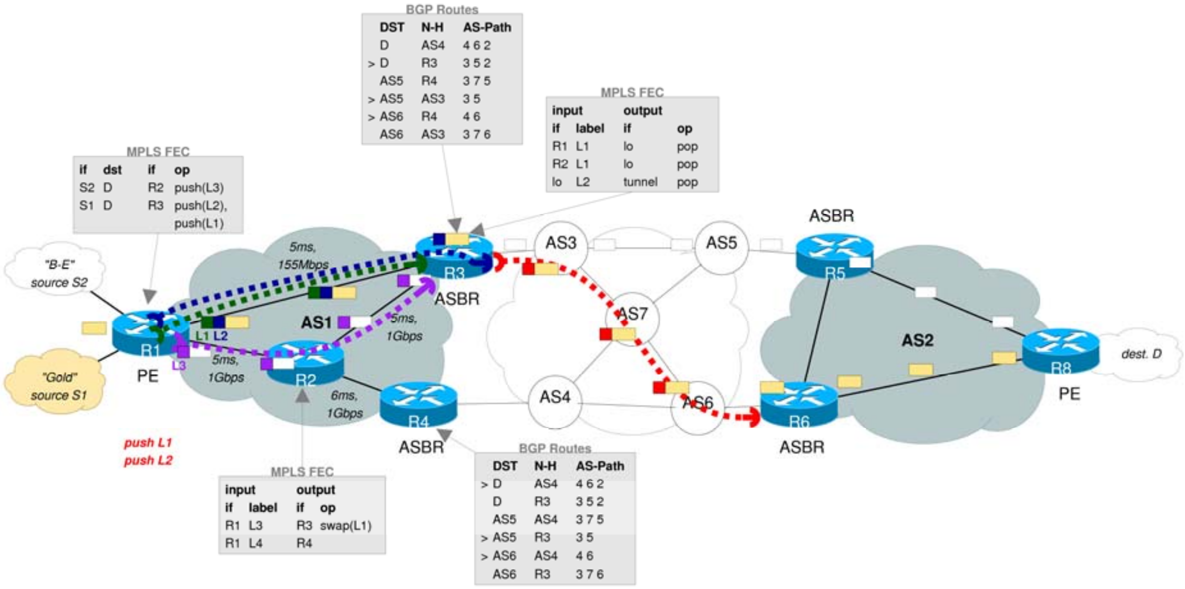
4.1.5.5 Provisioning

The objective of the *Provisioning* component is to configure the network resources in order to use the set of paths selected by the *Paths Selection* algorithm. This involves configuring the tunnel ednpoints and configuring the network routers with the necessary state for directing the flows on their assigned paths (and possibly inside tunnels). The network provisioning has to take place at two different places. Let's take the example of a path assigned to a flow with a local source and a remote destination. On one side, the local INP network must be configured so as to direct the traffic received from the source network (through its PE router) to the selected egress router. In addition, the egress router must be configured to direct this traffic into a tunnel headed at the remote INP's ingress. On the other side, the remote INP must be configured so as to forward the traffic received from its ingress to the destination. We discuss the local provisioning in Section 4.1.5.5.1 and the remote provisioning in Section 4.1.5.5.2.

4.1.5.5.1 Local Network Provisioning

The local network provisioning task will depend on the Network Plane mechanisms deployed in the local INP. In this section, we describe the necessary tasks for performing this provisioning in the case of MPLS-based Network Planes and multi-topology IGP-based Network Planes.

We describe an example relying on MPLS-based Network Planes in Figure 23. This example is based on the situation of Figure 22. We assume that the path assigned to the traffic flow *(S1,D)* is through the inter-domain IP tunnel from R3 to R6. A first provisioning step is to configure the ASBR *R3* as the head-end of the tunnel. This includes configuring the tunnel based on the mechanism and parameters obtained by the *Ingresses Advertisement and Discovery* component. Directing the traffic sent by *S1* to the egress *R3* is done through the Network Plane obtained from the *Mapping* component. In this case, we will consider that the path inside the assigned network plane is implemented by the MPLS LSP from *R1* to *R3*. The labels used to identify this LSP are typically distributed by the RSVP-TE protocol [RFC3209]. To direct the flow from *S1* into this LSP, the PE router *R1* is configured so as to push the MPLS label *L1* before the packets from *S1* and directed to *D*. Based on this label, the packets will be directed to *R3* where the label will be popped.



MPLS FEC

if	dst	if	op
S2	D	R2	push(L3)
S1	D	R3	push(L2), push(L1)

BGP Routes

	DST	N-H	AS-Path
	D	AS4	4 6 2
>	D	R3	3 5 2
	AS5	R4	3 7 5
>	AS5	AS3	3 5
>	AS6	R4	4 6
	AS6	AS3	3 7 6

MPLS FEC

input		output	
if	label	if	op
R1	L1	lo	pop
R2	L1	lo	pop
lo	L2	tunnel	pop

MPLS FEC

input		output	
if	label	if	op
R1	L3	R3	swap(L1)
R1	L4	R4	

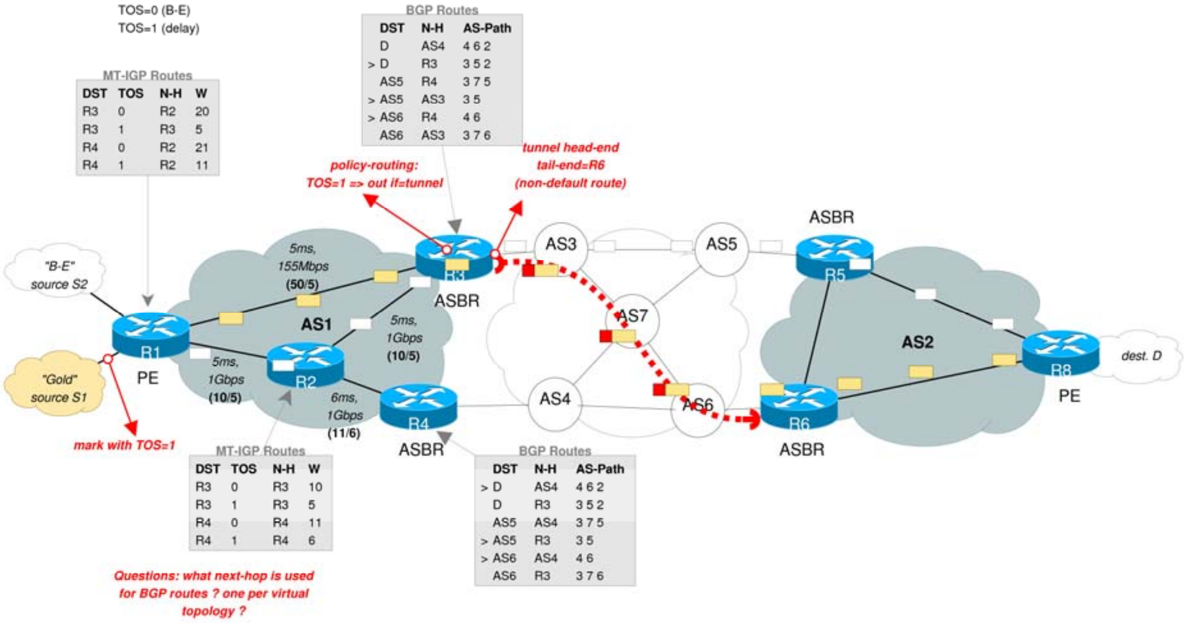
BGP Routes

	DST	N-H	AS-Path
>	D	AS4	4 6 2
	D	R3	3 5 2
	AS5	AS4	3 7 5
>	AS5	R3	3 5
>	AS6	AS4	4 6
	AS6	R3	3 7 6

Figure 23 -Example configuration if Network Planes are implemented using MPLS.

In addition to this, it is required to identify on *R3* the outgoing interface where the packets will be directed. In this case, the outgoing interface is the virtual interface associated with the tunnel. We need a double MPLS encapsulation to achieve this objective (as in MPLS-based VPNs [DAVI00, RFC4364]). In Figure 23, we use an additional label *L2* for this purpose. This label is pushed on the packets from *S1* to *D* before the label used for the intra-domain LSP. This second label might be distributed by BGP. When the packets arrive at *R3*, the first label (*L1*) is popped and the packet is delivered locally. Since there is a second label, a second lookup is performed in the FEC whose result is to pop the final label and direct the packets to the tunnel interface. The external label can be distributed by using MP-BGP.

In a second example, we consider a situation where the Network Planes are implemented using Multi-topology IGP Routing. This situation is illustrated in Figure 24. In this example, there are two virtual routing topologies defined in *AS1*. One is associated with *TOS=0* and is optimized for bandwidth while the second one is associated with *TOS=1* and is optimized for delay. We assume that we optimize the end-to-end latency and therefore we need to use the Network Plane with *TOS=1* for carrying the packets from *S1* to the egress *R3*. The source S1 must mark the packets destined for D that must benefit from the reduced latency with a DSCP value of 1. In this manner, they will be forwarded according to the IGP routing choices optimized for delay. One issue however is to distinguish the egress that must be used to reach *D* since it might be different from the default egress selected by BGP. One possible way to advertise the egress that should be used when the DSCP value is 1, is to leak the destination prefix *D* with the next-hop corresponding to the egress in the virtual topology associated with TOS=1. Another possibility is to use MP-BGP and use a special address family composed of the TOS value and the destination prefix.



MT-IGP Routes

DST	TOS	N-H	W
R3	0	R2	20
R3	1	R3	5
R4	0	R2	21
R4	1	R2	11

BGP Routes

	DST	N-H	AS-Path
	D	AS4	4 6 2
>	D	R3	3 5 2
	AS5	R4	3 7 5
>	AS5	AS3	3 5
>	AS6	R4	4 6
	AS6	AS3	3 7 6

MT-IGP Routes

DST	TOS	N-H	W
R3	0	R3	10
R3	1	R3	5
R4	0	R4	11
R4	1	R4	6

BGP Routes

	DST	N-H	AS-Path
>	D	AS4	4 6 2
	D	R3	3 5 2
	AS5	AS4	3 7 5
>	AS5	R3	3 5
>	AS6	AS4	4 6
	AS6	R3	3 7 6

Figure 24 - Example configuration if Network Planes are implemented using MTR.

4.1.5.5.2 Remote Network Provisioning

In the above section, we have not discussed of the provisioning in the remote INP. However, the provisioning steps are similar to those of the local INP. The difference lies in the fact that the remote provisioning is not triggered by the *Paths Selection* component, but by a remote INP, through the sending of a *Path Setup Request* message (PSReq).

The PSReq message is addressed to a remote INP in order to request a path between a remote ingress *RI* and the destination network *D* with some constraints. The PSReq message can also be sent to request a path in the reverse direction between a remote source network *S* and a local ingress router *LI*. In the later case, it is possible to force the remote egress router that must be used or this choice can be left free for the remote INP (in this case, this does not impede on the remote INP's traffic engineering objectives such as load-balancing or peering cost minimization).

When such a message is received by a TSC, it must first check that the request can be accommodated. Then, pre-reserve the resources and finally send an answer to the requestor under the form of a Path Setup Response message (PSResp). The PSResp message would typically contain an optional DSCP value identifying the Network Plane that must be used in the remote INP.

Source	S
Destination	D
Direction	S→D or D→S
Ingress router	Local ingress (LI) or remote ingress (RI) depending on the direction
Egress router	Only in case of reverse path request [optional]
Tunneling mechanism	As obtained from the *Ingress Advertisement and Discovery* component.
Latency constraint	[optional]
Bandwidth constraint	[optional]

Table 9 - Path Setup Request message parameters.

When PSReq messages are issued to remote domains, transactions might be required to first pre-reserve in the remote domains the required resources and if all the pre-reservations are successful, to commit all the reservations. The whole transaction is successful in this case. If one path request cannot be fulfilled in one of the remote domains, what would be the right action to take? (1). Cancel the whole transaction? (2). Commit the successful pre-reservations and re-trigger the Paths-Selection algorithm (with the established paths in a taboo list so that they cannot be moved)? (3). Other solution?

4.1.6 Conclusion

In this document, we have laid out the initial specification of the IP Tunneling solution. We have presented the components of a TSC, the main block of the solution. We have described the requirements of each component as well as its parameters and how to implement the IP Tunneling framework in presence of various Network Plane implementations.

4.2 q-BGP enhancement

One of the proposed methods to maintain and distribute routing information in inter-domain network planes in AGAVE is the use of a QoS-enhanced version of BGP, called q-BGP. The following section describes the work to be performed to adapt q-BGP for use in AGAVE and describe a range of processes, algorithms and protocol enhancements. q-BGP builds on BGP in that it includes two new attributes:

- A QoS Service Capability attribute, which signals, as part of the q-BGP OPEN message, that this message and following UPDATE messages are part of a QoS aware q-BGP session, and which network plane it belongs too.
- A QoS_NLRI attribute which expresses which network plane this message belongs to and the optional fields which describe the QoS attributes of the path expressed in the message.

q-BGP was significantly developed as part of the MESCAL project and is well documented in [MSCLD12] [MSCLD13]. Q-BGP is available as an Internet draft at [BOUC05].

To extend and investigate q-BGP further we will consider a sub-set of the following areas of research:

- QoS-Attribute types: The types of data conveyed in the the QoS_NLRI update messages
- QoS-Attribute calculation: How the QA values are obtained and what is presented in the QoS_NLRI fields.
- QoS-Attribute usage: QA could be generated and interpreted in a number of ways.
- Route selection policies: The method used to compare q-BGP UPDATE messages to choose which route is installed in the q-FIBs.
- Network plane optimisation: How a single plane is optimised globally based on local domain decisions and how multiple planes may interact assuming hard and softer partitioning.

4.2.1 QoS-attribute types

The first area of investigation is the actual information that is conveyed in the QoS_NLRI field. [BOUC05] specifies 3 types of information that can be conveyed, described briefly below, together with a few additional types which will form part of the research.

4.2.1.1 *Primitive types*

- **Average one-way delay** [BOUC05]: The average delay a packet can expect when following this path.
- **Minimum one-way delay** [BOUC05]: The minimum delay a packet can expect when following this path.
- **Available bandwidth** [BOUC05]: The available capacity a packet can expect when following this path. There is no specified method of calculating this figure, and could be the total available bandwidth to the given prefix from the AS sending the UPDATE message, or could be a fraction of that available bandwidth, which may be a more accurate approach since the total available bandwidth from an AS will obviously not be available to all ASes receiving the message. The method of calculating what is to be offered is a significant research topic in the work that will be carried out as it can have a big effect on route movements.
- **Packet Loss Rate** [BOUC05]: The expected rate of packet loss that can be expected when following this path.

4.2.1.2 *Derived types*

These are attribute types which have a useful purpose on their own but rely in part on first-order types, for example one way of defining jitter is as the variance or range of delay.

4.2.1.2.1 Jitter, or Inter-packet delay variation

Jitter is an important attribute [BOUC05] to many real time applications and could serve a useful purpose in planes which carry real-time traffic. Since jitter is commonly caused in the network by congestion and queuing it could also be used as a measure of congestion.

4.2.1.2.2 Traffic volatility

This is a measure of the change in available bandwidth along this route. Such a metric could serve as a sign of route instability or the availability of links in the path.

Derived types can also be of a variety which aren’t usually direct measurements but rather a calculation based on other values, for example a statistical metric of a primitive type. Two proposed types are listed below:

4.2.1.2.3 Confidence factor

This is a statistical measure of the accuracy that can be expected of the non-derived types. Depending on its usage it may serve a similar purpose to second-order types like traffic volatility.

4.2.1.2.4 Abstract Performance Metrics

Given that the routing behaviour for a given plane should be the same across a plane there is scope to investigate abstract metrics which express the suitability of the route to the purpose of the plane. Such metrics can then be compared directly in the choice of route.

4.2.2 QoS-attribution calculation

Now that we have seen the types and classes of QAs the question arises of how these values are obtained. The calculation of QA values *within* an AS has two purposes:

- To be used in *local decision making*.
- To be used in *q-BGP advertisements* to adjacent ASes.

The methods used to calculate the values for the two purposes above are usually the same, but are not required to be so. Note also that this applies to values within an AS, and is separate from the values that are received from an adjacent AS. Whatever their purpose the values can be either a static value, a periodically changing value, potentially based on live monitoring, and a semi-static value which is a value generated by some algorithm which could have as its input monitoring data. The three cases are described below:

4.2.2.1 Static values

This is where the values of QAs that are used are specified as static values in the q-BGP configuration, typically from the off-line TE. The problem with the use of non-changing values is that they do not represent current network conditions and the network doesn't have a chance to adapt since it is an open-loop system. What typically happens in the network is the phenomenon of "QA rush" where many ASes choose a good route and send traffic along it, causing congestion on the route and decreased levels of service [GRIF07].

4.2.2.2 Monitored values (dynamic values)

At the other extreme of volatility the values of QAs used are monitored live and decisions on near-real-time information is made. The benefit of this approach is that QA values now reflect actual network conditions and can lead to a better use of network resources, with, depending on how this is achieved, less of the "QA rush" described earlier. However, care must be taken not to advertise newly updated values too frequently as this can lead to repeated avalanches of q-BGP messages throughout the network and the inability to converge on a stable routing configuration.

4.2.2.3 Semi-static values

This is a middle ground between fully static and fully dynamic QA values. These could take the form of predefined values which are advertised when a certain condition is met, which is triggered by monitored live values. Alternatively an algorithm could make intelligent decisions on when and what to re-advertise based on monitored information. This will form a significant part of the research into q-BGP as it is required to avoid "QA rush" and to optimise network plane usage.

4.2.3 Route selection policies

Given a range of QA types that are made available to the q-BGP route selection process and that their calculation can be done in a number of ways, we now examine the process that actually makes the decision on which route to take, based on the above information.

4.2.3.1 Priority based route selection process

The route selection policy described in [BOUC05] specifies a scheme whereby the QoS attributes of incoming q-BGP UPDATE messages are compared based on a priority order scheme.

The highest priority QA type in each message is compared first, and the message with the better value (ie. higher value in the case of available bandwidth, lower value in the case of delay etc..) is chosen. Given an equality of absolute values the second priority QA type is compared and so on.

This however leads to situations where very similar values of QA are seen as totally different and most decisions are based on the first priority QA type. To increase the chance that second and further priority QA types are used as part of the decision process an equivalence margin is defined such that:

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

then the messages are considered equivalent in terms of this QA. Where QAmargin is the size of the equivalence margin and MessageA_QA and MessageB_QA are the QA values of the messages that are being compared.

The use of the priority based route selection process can be seen in [BOUC05] and the equivalency margin can be seen in [GRIF07].

4.2.3.2 Alternative route selection processes

Other than the priority based scheme described above there are other potential route selection processes:

4.2.3.2.1 Comparison based on convoluted metric

Here comparison of routes is performed based on a formula which attempts to normalise and collapse the QAs into a single numerical value, for example to find the weighted average:

$$\frac{\sum_{n} \alpha_n \frac{QAn_{inco\min g}}{QAn_{typical}}}{n}$$

Where: $QAn_{incoming}$ is the *n*th QA type (delay, bandwidth etc..) of the incoming message and $QAn_{typical}$ is a typical value for QoS attribute QAn, and αn is a weighting co-efficient.

$QAn_{typical}$ could also be the best value (i.e. lowest for delay, highest for bandwidth) of all those in the RIB, so then the above equation becomes the average of normalised QoS attributes.

Such schemes may be prone to routing loops, but such a case is removed by examining the AS_PATH at the input message filter.

αn is potentially a source of programmability in q-BGP, or could be specified in the network plane definition. Such comparison logic will be investigated, especially in an attempt to prevent route oscillations when using dynamical monitored values.

4.2.3.2.2 Ranked comparison

This is where all incoming advertisements are ranked in comparison to all others in the RIB and route selection is then based on the ranks of each QA type. This is different to the plain priority based scheme because it ignores the absolute differences in QAs and rather considers how good they are in comparison to all that are available.

4.2.3.3 QoS attribute usage

A further area of investigation is how exactly the q-BGP process uses the information gained in the UPDATE messages. It is possible to throttle incoming messages, or use hysteresis to prevent the propagation of large avalanches of messages and causing large scale instability. Such methods will be investigated, especially when dynamically monitored QAs are being used in advertisements.

4.2.3.4 Re-advertisement of q-BGP UPDATEs and QA values

In a related way the conditions under which q-BGP UPDATE messages are re-advertised will be investigated. Oscillation dampening is the most significant driving factor for investigating this aspect of q-BGP and solutions to be examined include the use of delayed or rate-limited (here the rate is the number of messages per second) re-advertisements, or the use of weighted-moving-average to make any large changes in QA values smaller and hopefully prevent avalanches of messages.

4.2.4 Single plane optimisation

The optimisation function which is implicitly encoded in the QA types, the route selection process and the re-advertisement rules has already been in part investigated for a single network plane which is hard partitioned from other planes in [GRIF07]. It was demonstrated that there isn’t always an obvious correlation between the factors being locally optimised for and the effect across the entire network.

4.2.4.1 Local decisions and global results

In the original BGP where AS Path length was used as one of the more significant comparison metrics and attempted to create routes which follow the short path between the source and destination, but we are now using more metrics to choose a route. This was seen in [GRIF07] where a selection of q-BGP route selection policies was compared and demonstrated that making a local decision to optimise for a certain attribute, say available bandwidth, didn't necessarily achieve a global optimisation on bandwidth, rather to achieve the best use of bandwidth a combination of bandwidth and delay were used. We will investigate such phenomena and the impact on network plane performance, and how local optimisation can be designed to achieve differentiated qualities of services across the network and how these policies would map to network planes.

4.2.5 q-BGP and the co-existence of planes

All investigations into q-BGP have been with a single plane and assumed that multiple planes would act similarly, given the same resources, and assuming a hard partitioning of network resources. We will investigate how q-BGP reacts when the partitioning is not hard and changes in one plane will affect the other planes in terms of available bandwidth, delay and other metrics which would cause a series of re-advertisements to be triggered. The complexity of the problem is further increased when advertisements are formed from dynamic values, and the multiple network planes form a closed-loop feedback system which could potentially be very unstable.

It is proposed that resources, specifically the inter-domain bandwidth available to each network plane, are not just specified as a single value, but as a maximum and minimum value which then forms the limits on the bandwidth usage by the q-BGP process per AS per network plane.

This direction of research potentially creates a very complex problem because of the many feedback terms that are seen between the layers and without very careful dampening and network control the network may not settle to a stable routing state. It is proposed that this is investigated further.

4.3 Inter-domain resilience issues

Some customer's applications such as the ones selected in AGAVE use case (VoIP and VPN) typically have high availability requirements ([AHMA01]). For example, for VoIP, the typical requirement for the media flows is a Loss of Connectivity (LoC) of less than 200ms.

On the other hand, BGP -the protocol currently used for inter-domain routing- has a slow convergence speed, typically between 1 and 100 seconds depending on the failure, the number of routes involved, and the topology etc. Typically it's not possible for the network operator to guarantee a LoC below 5 seconds, even with best hardware, software and engineering rules.

To address this issue, waiting for hardware improvement thanks to the Moore Law is not an option since hardware already hardly follow inter domain routing route growth. So if we add the increasing availability requirements from customers and new applications the situation is not expected to improve by itself.

4.3.1.1 Introduction

The BGP protocol is heavily used by INP networks. For resiliency purposes, most of the IP network operators deploy redundant routers and BGP sessions to minimize the risk of BGP session breakdown towards their customers, providers or peers. In a context where an INP wants to upgrade or remove a particular router, line card or external link that maintains one or several BGP sessions, our requirement is to avoid customer or peer traffic loss as much as possible. As the failure is known ahead of time, it should be made possible to reroute the customer or peer traffic before the maintenance operation occurs and BGP session is torn down. This requires BGP to be able to advertise "future" non urgent events and not only "past" events.

Currently, the BGP specification does not include any operation to prevent traffic loss in case of planned maintenance. A successful approach of such mechanism should indeed minimize the loss of traffic in most foreseen maintenance situations. It should be easily deployable and if possible, provide backward compatibility. In other word, it should be lightweight.

4.3.1.2 Problem statement

Currently, when one (or many) BGP session needs to be shut down BGP breaks the existing path and then informs its peers about the failure. This generates packets loss.

As an example, let's take this very simple above topology where a customer (AS A) is dually connected to its provider (AS B):

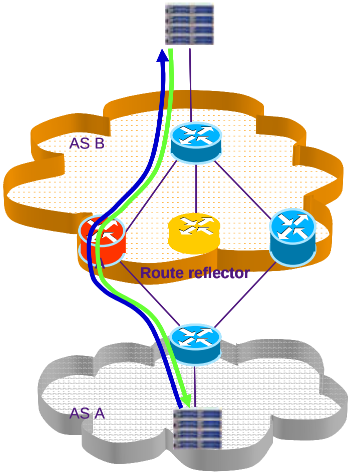


Figure 25: Topology creating LoC during BGP PM

During the planned maintenance, the router called "C12C" -in red in Figure 25- needs to be upgraded and hence shutdown. It can be directly reloaded with the "reload" command which is more or less graceful for the network. Or the INP can first shutdown the BGP sessions to warn the peers. But in both cases, as shown in [DUBO04], during the BGP convergence, packets are lost for a few seconds in both directions (green for downstream, blue for upstream):

Case 1: 100 BGP routes need to be rerouted.

Case 2: 100 000 BGP routes need to be rerouted.

Reload of C12C

(shut the whole BGP router)

Clear ip BGP C12C

(shut one eBGP session)

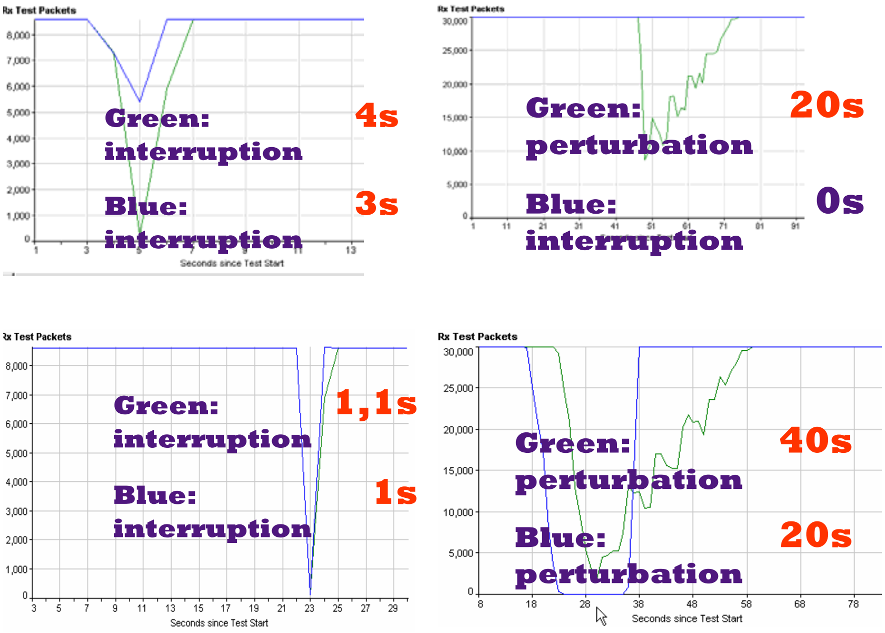


Figure 26: LoC during BGP convergence

This result is very preliminary because only one BGP topology is tested and results have not been analyzed. For example, in the upper right case, the upstream flow in blue is not interrupted during the reload which is surprising. We suspect that with the "reload" command, the Cisco GSR router -which is highly distributed- reloads its control plane card (GRP) but doesn't explicitly reload its line card. With the recent implementations of the NSF (Non Stop Forwarding) and GR (Gracefull Restart) features, we suspect the line cards keep running and forwarding even with their head (control plane) cut. So AS A has still two forwarding paths and the blue flow has no interruption even if AS "A" takes time to detect the failure, performs a BGP convergence and updates its FIB.

One of the reasons for this LoC is the use of BGP Route Reflectors which may hide some alternative paths. Hence some routers and typically the ASBR C12C don't have an alternate route. When the nominal path is shutdown, the ASBR starts dropping packets and advertise the failures to its neighbors. The peers try to find an alternate route but this may requires some additional BGP message exchange.

This behavior is not satisfactory in a maintenance situation because customer's (or peer’s) traffic that was directed towards the removed next-hops is lost until the end of BGP convergence. As it is a planned operation, a make before break solution should be made possible.

As maintenance operations are frequent in large networks, the global availability of the network is significantly impaired by the BGP maintenance issues. For example, in a tier-1 European ISP, planned maintenance operations account for 50% of the routers failures. As another example, in a major VPN SP, planned maintenance account for 80% of PE failures and are responsible for 46% of the PE unavailability.

Addressing planned maintenance operations is not a BGP specific issue but a generic signaling and routing issue. Some routing or signaling protocols are already addressing it, for example MPLS-TE in [VASS01], GMPLS in [ALI01], IS-IS TE in [VASS02], link state IGP (OSPF or IS-IS) in [FRAN05] and [FRAN06]…

4.3.1.3 *Requirements for the BGP solution*

The planned maintenance solution should be lightweight to minimize the modifications to BGP protocol. It should be incrementally deployable, at least on a per AS basis but preferably on a per router increment. It should brings improvement incrementally as a solution requiring a full scale deployment before any improvement is likely to never be deployed especially when independent (so selfish) networks are concerned. It should also be applicable to the multi-protocol extensions of BGP to also be applicable to others address families (eg IPv4, IPv6, multicast, labeled, MPLS VPN…)

Both steps of the planned maintenance should be covered: when the router / eBGP link is shutdown and when the router / eBGP link is brought back online.

The solution should work with different forwarding paradigm:

- IP (pervasive iBGP)
- MPLS (BGP free core)
- BGP/MPLS VPNs

The solution should be applicable to all common BGP topologies and especially the following ones which are the most used.

4.3.1.3.1 eBGP topologies

The eBGP topology refers to the inter-domain topology at the AS level: how many links between the ASes, how many ASBR involved, how many ASes involved.

The solution should be applicable to a customer, peers or provider dually connected to one or two ASBRs:

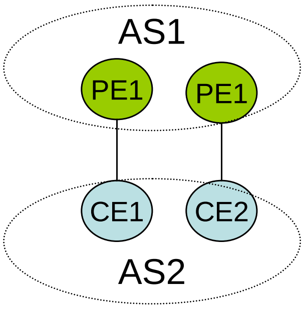


Figure 27: eBGP topology 2PE-2CE

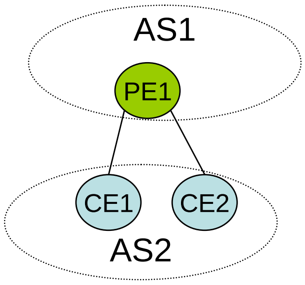


Figure 28: eBGP topology PE-2CE

But given the above requirements, an Internet wide convergence is out of scope:

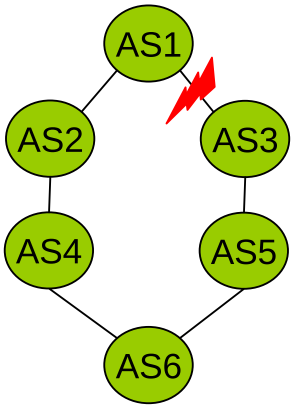


Figure 29: eBGP Internet wide topology

4.3.1.3.2 iBGP topologies

The solution should be applicable different iBGP topologies such as full mesh, route reflectors, hierarchical route reflectors and centralized route reflectors:

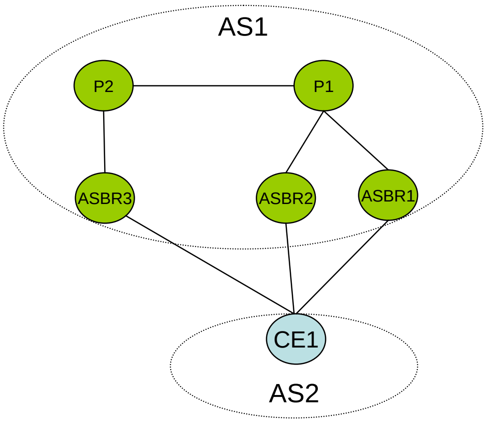


Figure 30: iBGP full mesh topology

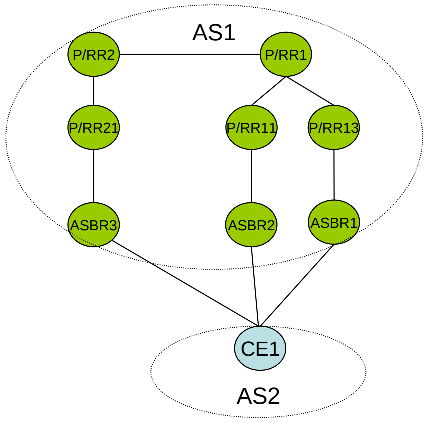


Figure 32: iBGP hierarchical RR topology

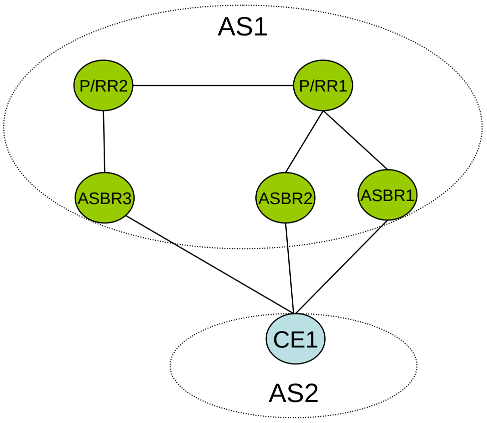


Figure 31: iBGP RR topology

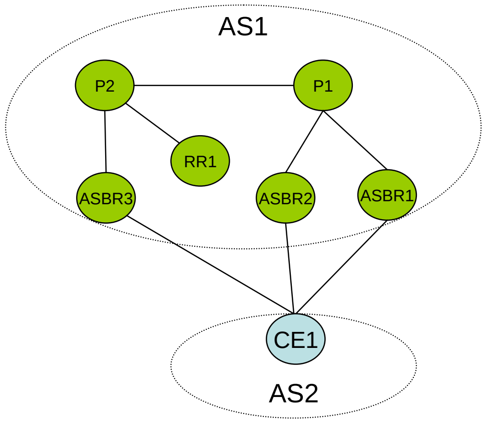


Figure 33: iBGP centralized RR topology

In the above figures, the solid lines are IP links. The iBGP sessions are not represented and are inferable from the name of the topology (eg full mesh implies a full mesh of iBGP sessions between all routers of the AS) and the name of the router (eg RR are Route Reflectors which centralizes iBGP sessions).

4.3.1.4 Solution

This work will be described in D3.2.

4.3.2 ASBR protection with RSVP-TE egress fast reroute

4.3.2.1 Background and Motivations

This work can be used to protect the inter-connection of NPs (implemented with MPLS) across multiple domains against inter-domain link or ASBR failures. Therefore, this work can result in robust PIs.

Some mission critical services such as VoIP require a deterministic fast recovery under 100ms upon link or node failure. The MPLS-TE Fast Reroute (MPLS FRR) technology defined in [RFC4090], allows guaranteeing such recovery performances, and is widely deployed today. It relies on a local protection of primary TE-LSPs, with local backup TE-LSPs that are established before the failure. Backup LSPs are setup between the node upstream to the protected element, called PLR (point of local repair) and a node downstream to the protection element, called Merge Point (MP) where the primary and backup LSP merge. During failure the upstream node (ie the PLR) updates its MPLS forwarding table so that the traffic received on the protected LSP is forwarded within the backup LSP. This procedure does not imply any path computation or signalling during the failure, and backup routes are pre-installed within the MPLS Forwarding Table, which allows guaranteeing deterministic sub-50ms recovery upon failure [ROUX04].

There are various MPLS FRR deployments strategies: Link protection can be ensured by setting up one-hop primary TE-LSPs protected by a backup TE-LSP that avoids the protected link, while node protection can be ensured by a full mesh of TE-LSPs between Edge Routers, protected by backup TE-LSPs that avoid the protected nodes. MPLS FRR allows protecting links and transit nodes of a TE-LSP. In return it does not allow protecting Ingress and Egress LSRs. Ingress LSR protection can be ensured by an IP FRR protection, realized by the router upstream to the Ingress LSR. The upstream router detects the failure and redirects the traffic towards an alternate Ingress LSR. In return, in the state of the art, Egress LSR protection cannot be ensured by the LSR upstream to the failure, it can only be performed by the Ingress LSR and this does not allow achieving sub-50ms recovery.

To ensure fast recovery upon link and node failures, operators deploy a mesh of TE-LSPs between their Edge routers. This allows ensuring fast protection of intra-AS traffic, but does not protect inter-AS traffic against inter-AS link and ASBR failures.

Inter-AS link protection and ASBR node protection is a key requirement for mission critical inter-AS communications such as the interconnection of VoIP gateways of distinct network operators.

Inter-AS link protection can easily rely on a one-hop TE-LSP setup on the inter-AS link, protected by a local backup TE-LSP that avoid the protected inter-AS link, and the eBGP session can be setup on top of this one-hop TE-LSP. This design scales well and requires a few configurations on ASBRs.

In return, the only mechanism today to ensure ASBR node protection consists of deploying end-to-end inter-AS MPLS-TE LSPs (see [RFC4216]) from PEs to PEs, that are locally protected with backup TE-LSPs. While really powerful, this mechanism faces obvious scalability limitations (the number of LSPs is the squared number of PEs), and requires strong coordination between operators. Also it requires that all operators along the inter-AS chain support RSVP-TE.

We define here an alternative mechanism called RSVP-TE Egress Fast Reroute (*Egress FRR*) allowing to protect ASBRs in a scalable way. *Egress FRR* is a new MPLS-TE Fast Reroute mechanism that allows protecting the Egress LSR of a TE-LSP. With such a mechanism a TE-LSP has two destinations, one primary and one backup Egress LSR. In nominal situation the penultimate LSR forwards the traffic to the primary egress, while during failure the traffic is forwarded to the backup egress. An Edge Router that learns, via BGP, a prefix reachable through two Egress ASBRs, installs this prefix within a TE-LSP that has for primary and backup destination these two Egress ASBRs. Upon failure the penultimate LSR forwards the traffic to the backup Egress LSR. This allows ensuring sub-50ms recovery upon ASBR failure.

An overview of the solution is provided in section 4.3.2.2. Section 4.3.2.3 defines the RSVP-TE *Egress FRR* mechanism. Finally section 4.3.2.4 defines extended BGP next-hop resolution procedures so as to support ASBR protection with RSVP-TE *Egress FRR*.

4.3.2.2 Solution Overview

The Fast Reroute Extensions to RSVP-TE for LSP Tunnel mechanism defined in [RFC4090] does not allow for fast protection of TE-LSP Egress LSRs. Upon failure the failover cannot be ensured by the LSR upstream to the failure. It is ensured by the Ingress LSR, which does not allow achieving sub-50ms protection.

The only way to ensure sub-50ms protection actually requires performing the failover on the node directly upstream to the failed element. For that purpose we define here a new MPLS-TE FRR mechanism called *Egress FRR,* that allows protecting the Egress LSR of a point-to-point TE-LSP. A backup Egress LSR is defined in advance, to protect a TE-LSP primary Egress LSR. A backup TE-LSP is setup between the penultimate LSR and the backup Egress LSR. Upon Egress LSR node failure or Penultimate LSR - Egress LSR link failure, the penultimate LSR redirects the traffic received on the protected TE-LSP, onto the backup TE-LSP, towards the backup Egress LSR. The backup route is preinstalled within the penultimate LSR forwarding table, which allows guaranteeing sub-50ms deterministic recovery upon egress LSR failure.

To ensure Egress ASBR protection, the BGP selection process on the Ingress Edge router is modified: For each prefix learned via BGP, reachable through two Egress ASBRs, the Ingress LSR installs this prefix within an *Egress FRR* protected primary TE-LSP whose primary and backup egress LSRs are these two Egress ASBRs. Note that when a route reflector is used, only one next-hop is advertised to Edge routers for a given prefix, so a BGP extension is required here so as to distribute several next-hops. This could rely on the mechanism defined in [BHAT06]

On the Backup Egress ASBR, there is a context specific IP forwarding table (aka IP Forwarding Information Based, FIB) for traffic received on the *Egress FRR* backup TE-LSP. This requires Penultimate Hop Popping (PHP) to be deactivated on the *Egress FRR* backup TE-LSP. In this context specific IP forwarding table, the Primary Egress LSR is not considered as a next hop and the traffic directly leaves the AS. Such context specific forwarding on the backup Egress ASBR allows avoiding the traffic to be redirected to the failed Egress ASBR.

For the sake of illustration, in Figure 34 below, there are two Egress ASBRs, R4 and R6 in AS1, to reach 1.1/16. An *Egress FRR* protected TE-LSP T1 is setup on R1, with R4 as primary Egress LSR and R6 as backup Egress LSR. A backup TE-LSP T2 is setup from the penultimate LSR R3 to the backup Egress LSR R6. On R1, T1 is selected to route traffic towards 1.1/16. R3 maintains two outputs within its forwarding table for the protected LSP, a primary output towards R4 and a backup output within T2 towards R6. Upon R4 failure, R3 reroutes the traffic within T2 towards R6, and on R6 the traffic is looked up in a context specific FIB that avoids R4.

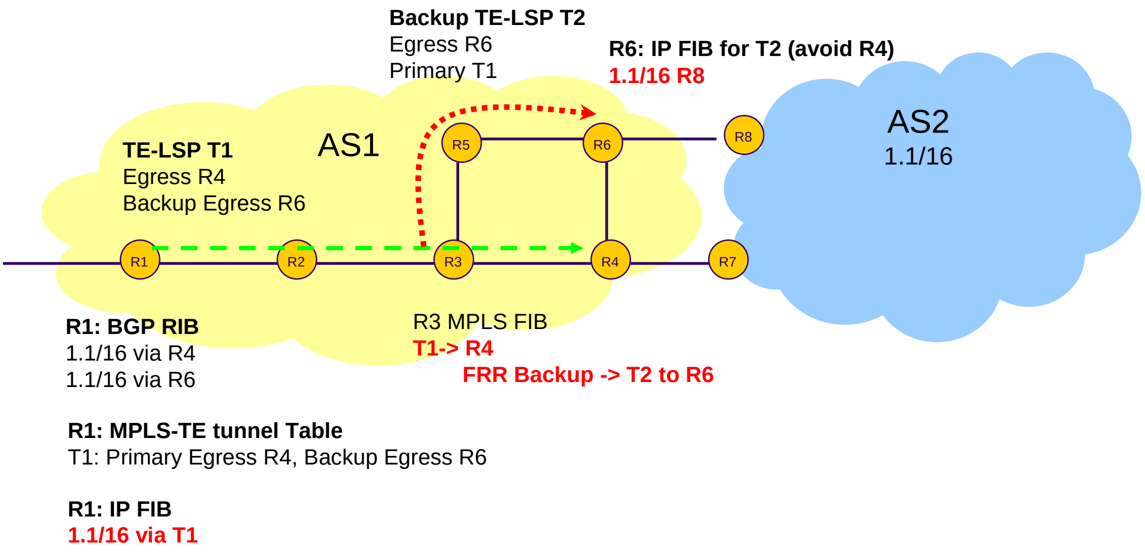


Figure 34 Egress ASBR protection with RSVP-TE Egress FRR

Similarly, to ensure Ingress ASBR protection, a one hop primary TE-LSP is setup on the Inter-AS link, protected by *Egress FRR* with a backup LSP towards a secondary Ingress ASBR.

For instance, in Figure 35 below the Ingress ASBR R7 is protected by an *Egress FRR* protected one-hop TE-LSP from R4 to R7 with a backup LSP from R4 towards the backup Ingress ASBR R8.

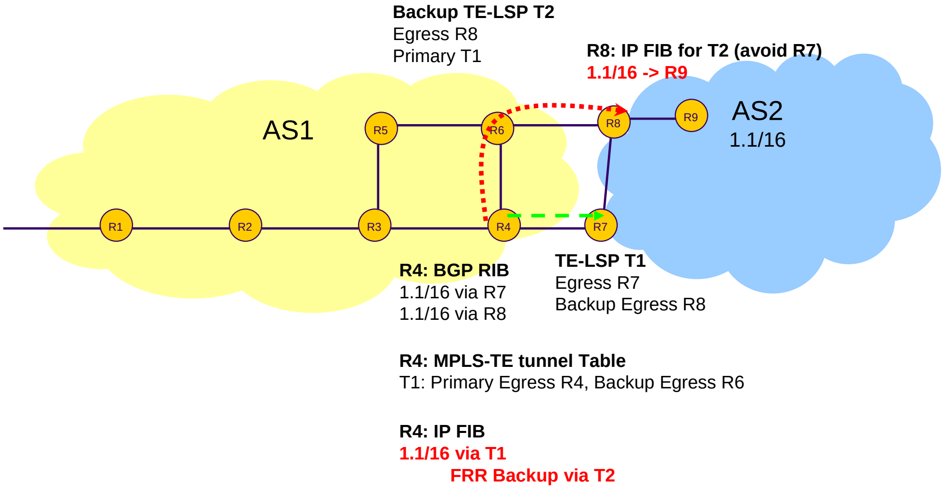


Figure 35 Ingress ASBR protection with RSVP-TE Egress FRR

Note: An alternative to ensure Ingress ASBR and inter-AS link protection consist of having an LSP down to the Ingress ASBR in the neighbouring domain. In this case the Egress ASBR is protected by a standard NNHOP bypass LSP.

For instance, in Figure 36 below the Ingress ASBR R7 and inter-AS link are protected by an *Egress FRR* protected TE-LSP from R1 to R7 with a backup LSP from R4 towards the backup Ingress ASBR R8. The Ingress ASBR is protected by a standard Fast Reroute backup LSP from R3 to R7. Note that this requires similar extensions to the BGP selection process on the Ingress LSR R1 but this requires here that the BGP next-hop self feature is not activated on R4 and R6.

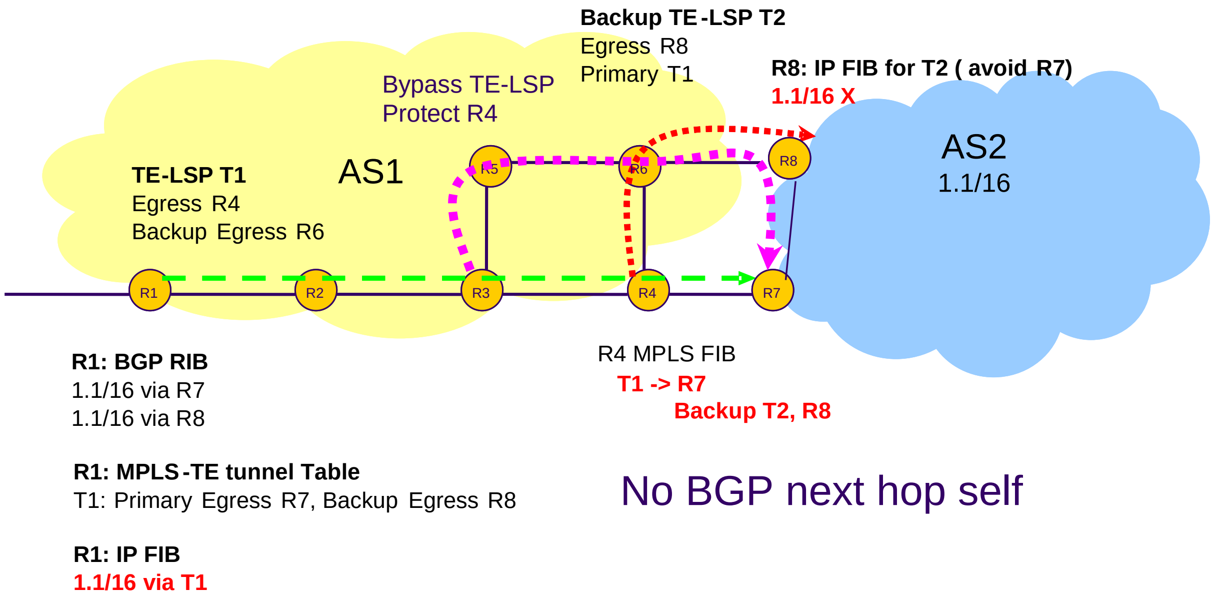


Figure 36 Egress ASBR, Ingress ASBR and inter-AS link protection with RSVP-TE Egress FRR

4.3.2.3 RSVP-TE Egress Fast Reroute

This section describes in details the RSVP-TE Egress Fast Reroute (*Egress FRR*) mechanism.

4.3.2.3.1 Egress FRR Terminology

The *Egress FRR* system described in Figure 37, to protect the Egress LSR of a primary TE-LSP in an MPLS-TE network, comprises:

(1) An MPLS-TE Network = A set of LSRs that support the RSVP-TE protocol defined in [RFC3209].

(2) A primary TE-LSP established with RSVP-TE.

(3) A backup TE-LSP established with RSVP-TE with as ingress LSR, the penultimate LSR of the primary TE-LSP, and as Egress LSR the Backup Egress LSR.

(4) The primary TE-LSP Ingress LSR (PIL).

(5) A set of transit LSRs of the primary and backup TE-LSPs.

(6) The Primary Egress LSR (PEL) = The Egress LSR of the primary TE-LSP.

(7) The Backup Egress LSR (BEL) = The Egress LSR of the backup TE-LSP.

(8) The PenUltimate LSR of the primary LSP (PUL) , in charge of setting up the backup LSP. During ultimate link failure or Primary Egress LSR failure, this router detects the failure and redirects the traffic towards the backup LSP.

-(1): MPLS-TE Network = A set of LSRs that support the RSVP-TE protocol defined in [RFC3209]
-(2): Primary TE-LSP established with RSVP-TE
-(3): Backup TE-LSP established with RSVP-TE
-(4): Primary TE-LSP Ingress LSR (PIL)
-(5): Transit LSRs of the primary and backup TE-LSPs
-(6): Primary Egress LSR (PEL) = The Egress LSR of the primary TE-LSP
-(7): Backup Egress LSR (BEL) = The Egress LSR of the backup TE-LSP
-(8): PenUltimate LSR of the primary LSP (PUL)

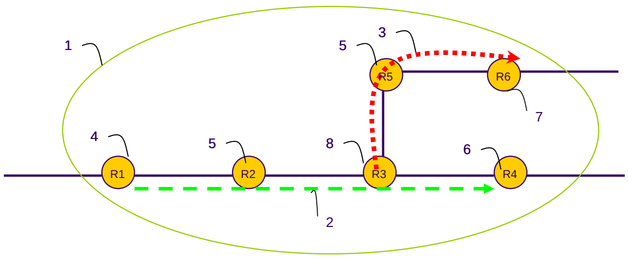


Figure 37 Egress FRR System

Note: An Ingress LSR can also be the penultimate LSR (case of one-hop primary TE-LSP).

4.3.2.3.2 RSVP-TE Signalling extensions

The *Egress FRR* mechanism requires extensions to RSVP-TE signalling defined in [RFC3209] and RSVP-TE Fast Reroute defined in [RFC4090].

New elements need to be carried within RSVP-TE *Path, Resv and PathErr* messages. Section 4.3.2.3.2.1 describes the required information, and section 4.3.2.3.2.2 proposes, for the sake of illustration, a way to encode this information in RSVP-TE.

4.3.2.3.2.1 Required information

The RSVP-TE *Path* message to setup the primary TE-LSP, needs to include the following information, in addition to the information defined in RFC3209:

- The indication whether *Egress FRR* is desired or not
- The IP address of the Backup Egress LSR
- Optionally the backup TE-LSP path, from the penultimate LSR to the backup egress LSR.

The RSVP-TE *Resv* message, for setting up the primary TE-LSP, needs to include the following additional information:

- The indication whether *Egress FRR* is available or not
- The indication whether *Egress FRR* is in use or not.

The RSVP-TE *Path* message, for setting up a *Egress FRR* backup TE-LSP must include the additional following information:

- The LSP Type = *Egress FRR* Backup LSP
- The primary Egress LSR address
- The primary LSP identifiers
- The indication whether *Egress FRR* protection is in use or not

The RSVP-TE *PathErr* message that is sent by the penultimate LSR when *Egress FRR* is triggered, must include a new error code "*Egress FRR* in use".

4.3.2.3.2.2 Information Encoding

We propose here, for the sake of illustration a way to encode within RSVP-TE, the required information defined above. Note that an IETF draft defining these fields and requesting for IANA code points will be submitted for the Q1 2007 IETF meeting.

- **The indication whether Egress FRR is desired or not:** Can be encoded within a new flag of the "Attribute Flags" TLV, carried within the RSVP object "LSP_ATTRIBUTE" defined in [RFC4420].
- **The IP address of the Backup Egress LSR:** Can be encoded within a new RSVP object called "Backup Egress Object".
- **The Backup LSP Path:** Can be encoded within the SERO RSVP object defined in [BERG06].
- **The indication whether Egress FRR is available or not:** Can be encoded with a new bit of the "RRO Attributes" sub-object of the RRO object, defined in [RFC4420].
- **The indication whether Egress FRR is in use or not:** When used in a *Resv* message it can be encoded within a new bit of the "RRO Attributes" sub-object of the RRO object. When used in a *Path* message, it can be encoded with a new bit of the "Attribute Flags" TLV carried within the "LSP_Attributes" object.
- **The indication of the LSP Type "Egress FRR Backup LSP":** Can be encoded with a new bit of the "Attribute Flags" TLV carried within the RSVP "LSP_Attributes" object.
- **The IP address of the protected primary Egress LSR:** Can be encoded with a new RSVP object called "Primary Egress Object".
- **The identifier of the protected primary TE-LSP**: Can be encoded within a new object, called "Primary LSP" that includes the Session objects and Sender Template objects of the protected TE-LSP.
- **The Egress FRR trigger notification:** Can be encoded using a new error value "Local Repair *Egress FRR* in use" of the RSVP error code "Notification" (code 25).

4.3.2.3.3 RSVP-TE Procedures

Standard RSVP-TE procedures to setup a TE-LSP, defined in [RFC3209], apply here unless explicitly specified below. They are not repeated here.

4.3.2.3.3.1 Procedures on the Primary TE-LSP Ingress LSR (PIL)

4.3.2.3.3.1.1 Procedure before failure

During the establishment of the primary TE LSP, the Ingress LSR includes in the RSVP-TE *Path* message, in addition to parameters defined in [RFC3209], the following parameters: The indication that *Egress FRR* is desired, the IP address of the backup Egress LSR, and optionally the path of the backup LSP from the Penultimate LSR (PUL) to the Backup Egress LSR (BEL). This path can be explicitly configured by the operator on the ingress LSR, or it can be dynamically computed by the ingress LSR.

On receipt of a *Resv* message for the TE-LSP, that indicates that *Egress FRR* is available, the Ingress LSR can determine that the requested *Egress FRR* mechanism is ready.

A primary TE-LSP protected by *Egress FRR* is used on the Ingress LSR to route IP and/or MPLS traffic. When an IP route is installed within an LSP protected by *Egress FRR*, the Ingress LSR must ensure that the route can be reached both via the Primary Egress LSR and via the Backup Egress LSR, and that the Backup Egress LSR does not rely on the Primary Egress LSR to forward the traffic to the destination (see also section 4.3.2.4).

An LSP protected with *Egress FRR* can be used for static routing, IGP routing (autoroute announce) or BGP routing (see also section 4.3.2.4).

A primary TE-LSP protected with *Egress FRR*, is configured on the Ingress LSR, directly by the operator, or indirectly via a network management system (NMS). The configuration includes, in addition to classical MPLS-TE parameters, the following parameters:

- The desire for *Egress FRR* protection
- The Backup Egress LSR IP address
- Optionally the explicit path of the *Egress FRR* backup LSP towards the backup Egress LSR.

4.3.2.3.3.1.2 Procedure during failure

Upon Primary Egress LSR failure, or ultimate LSP link (i.e. PUL-PEL link) failure, the Ingress LSR receives a *PathErr* message with an error code 25 (Notification) and a new error value "*Egress FRR* in use" sent by the penultimate LSR. It also receives a *Resv* message that indicates that *Egress FRR* is in use, and with a modified RRO, including the backup path between the penultimate LSR and the backup Egress LSR.

On receipt of these messages, the Ingress LSR still forwards traffic within the LSP. After expiration of a timer, if the *Egress FRR* is still in use, the Ingress LSR can optionally perform the following procedures:

- It can put the LSP metric to INFINITY, so that routing protocols that use the LSP (IGP or BGP), reroute the traffic within another LSP towards another Egress LSR (potentially but not necessarily the backup Egress LSR), in a make before break manner.
- It may, after the expiration of a configured timer, delete the LSP.

4.3.2.3.3.1.3 Primary LSP deletion

To delete a primary TE-LSP the ingress LSR sends an RSVP *PathTear* message as defined in [RFC3209]. This deletion must trigger the deletion of the corresponding *Egress FRR* backup LSP, on the penultimate LSR.

4.3.2.3.3.1.4 Reversion

When the failure of the primary Egress LSR or the ultimate link is repaired, a reversion, i.e. a switchover on the primary path, can be performed in two ways:

- This can be done directly by the penultimate LSR, in which case the Ingress LSR is not implicated. The Ingress LSR wil receive a *Resv* message that indicates that the *Egress FRR* mechanism is no longer in use.
- This can be done by the Ingress LSR, which establishes a new primary LSP towards the primary Egress LSR, potentially protected by a backup Egress LSR, and then redirects the traffic towards this new LSP before to delete the old LSP, if it is still alive.

4.3.2.3.3.2 Procedures on the Penultimate LSR (PUL)

4.3.2.3.3.2.1 Primary and backup TE-LSP setup

Upon reception of an RSVP-TE *Path* message for a new LSP, including the indication that *Egress FRR* is desired, an LSR checks if it is the penultimate LSR on the path, by computing the number of hops to the destination. If it is one hop to the destination, the LSR is the PenUltimate LSR (PUL) on the path, and it must perform the following operations:

(1) the *Path* message must be forwarded to the primary Egress LSR, following RFC3209 procedures, and without modifying the *Egress FRR* indication. The backup Egress LSR address must be kept, and the potential backup path must be removed.

(2) A backup TE-LSP must be setup, towards the backup Egress LSR. For that purpose the PUL sends an RSVP *Path* message that includes none exhaustively:

- A session object with, as destination the Backup Egress LSR, and as tunnel id, a locally generated id.
- A sender-template object with, as source address a PUL address, and as LSP-id a locally generated id.
- An explicit route carried within an Explicit Route Object (ERO), which can be computed dynamically by the PUL, or partially/entirely specified in the Path message for the primary LSP. The path followed by the backup LSP must not traverse the primary Egress LSR.
- The LSP Type = *Egress FRR* Backup LSP.
- The IP address of the Primary Egress LSR.
- The identifier of the protected primary TE-LSP.

Note that backup LSP parameters (including bandwidth, affinities and priorities) and primary LSP parameters, can be equal or can differ. This is a local decision on the PUL.

On receipt of a *Resv* message for the backup LSP, indicating that the backup LSP is established, the PUL sends a *Resv* message for the primary LSP (provided it already received a *Resv* message for the primary LSP), towards the Ingress LSR, indicating that the *Egress FRR* procedure is available.

When the PUL is a transit LSR, it maintains in its MPLS Forwarding Table, two outputs for the incoming label of the protected primary LSP:

- A primary output, which points to the outgoing interface towards the primary Egress LSR.
- A fast reroute backup output which points to the backup LSP interface and label.

When the PUL is also an Ingress LSR (case of one-hop primary LSP), it maintains in its IP Forwarding table two outputs for each IP prefix routed within the protected primary LSP:

- A primary output, which points to the outgoing interface towards the primary Egress FRR.
- A fast reroute backup output which points to the backup LSP interface and label.

In nominal situation (ie Primary Egress LSR up) the backup output is not active.

4.3.2.3.3.2.2 Procedure during failure

The PUL detects the failure of the ultimate link or the failure of the primary Egress LSR, thanks to a layer 2 alarm such as an SDH alarm (e.g. AIS, RDI), or thanks to a heart beat mechanism such as the BFD (Bidirectional Forwarding Detection) protocol. Upon failure detection, the PUL, immediately updates its IP/MPLS forwarding table, the primary output is deactivated and the backup fast reroute output is activated. At that time the traffic is redirected on the backup LSP towards the backup Egress LSR.

At the same time the PUL sends a *PathErr* message towards the Ingress LSR, for the primary TE-LSP, with the error code Notification (= code 25) and error value *"Egress FRR* in use". It also sends a Path message for the backup TE-LSP towards the Backup Egress LSR, indicating that *Egress FRR* is in use.

It also sends a *Resv* message towards the Ingress LSR, for the primary TE-LSP, indicating that *Egress FRR* is in use, and with a modified RRO including the path between the PUL and the backup Egress LSR.

The RSVP Path State Block (PSB) for the impacted primary LSP is maintained. The refresh timer for the RSVP Resv State Block (RSB) of the impacted primary LSP (i.e. the refresh timer for *Resv* sent by the failed Egress LSR), is deactivated, and the PUL works as if it were still receiving *Resv* message refreshes from the primary Egress LSR. Particularly, it still refreshes upstream Resv states.

An implementation may use Backup LSP *Resv* refresh messages as primary LSP *Resv* refreshers.

4.3.2.3.3.2.3 Primary LSP Deletion

On receipt of a *PathTear* message or a *ResvTear* message for the primary LSP, the PUL needs to delete the backup LSP as well. It has to send a *PathTear* for the backup LSP, towards the backup Egress LSR.

4.3.2.3.3.2.4 Reversion

When the failed element is repaired, the PUL can locally start again refreshing the primary LSP towards the primary egress LSR, by sending a *Path* message. It can then reactivate the primary output in its forwarding table and redirect the traffic towards the primary Egress LSR.

When the reversion is performed, a *Resv* message is sent towards the Ingress LSR, indicating that the *Egress FRR* procedure is no longer in use, and with an RRO including the direct path towards the primary Egress LSR. A *Path* message is also sent towards the Backup Egress LSR, indicating that the *Egress FRR* procedure is no longer in use.

4.3.2.3.3.3 Procedures on the Backup Egress LSR (BEL)

The backup Egress LSR has to switch traffic received in the backup LSP in the context of the failed primary Egress LSR, so as not to forward traffic back to this failed LSR.

For that purpose penultimate hop popping must be deactivated for the backup LSP, that is the backup Egress LSR must send a label >=16 within the *Resv* message for the backup LSP. As such, the Egress LSR knows that if traffic received on this LSP this means that the primary Egress LSR has failed and that it must notforward traffic to this Egress LSR.

The Backup Egress LSR must maintain one context specific FIB per protected primary Egress LSR. The route selection process to populate a context specific FIB for a protected primary Egress LSR, is such that routes that traverses the primary Egress LSR are not taken into account and not installed.

On receipt of a *Path* message for a new *Egress FRR* Backup LSP, the backup Egress LSR, allocates a label and installs the label in its MPLS Forwarding table. This label mapped to the context specific FIB for the corresponding primary Egress, identified in the *Path* message. It then replies with a *Resv* message carrying the allocated label.

4.3.2.3.4 Make before break procedure

The primary and backup LSP may be re-optimized independently or simultaneously.

RSVP-TE LSP reoptimization is performed in a make before break manner: A new LSP following a better path is setup, it shares resources with the old LSP, then the Ingress LSR redirect traffic on this new LSP and the old LSP is finally torned down. This allows tunnel reoptimization with minimu impact on the traffic. RSVP-TE make before break procedures are detailed in [RFC3209].

Practically, reoptimization occurs when there is a better path in the network (new link/node added, metric change, bandwidth released), or after a local fast reroute operation (global reoptimization). The frequency actually depends on the frequency of the above events. Note that the reoptimization can be event driven or timer driven. The timer driven approach is recommended for stability reasons.

Upon backup LSP reoptimization, the new backup LSP shares resources with the old backup LSP following make before break procedures defined in [RFC3209].

Upon primary LSP reoptimization, if the PUL is modified, a new backup LSP will be setup and the old backup LSP is deleted. The old primary LSP and the new primary LSP share protection resources, following make before break procedures defined in [RFC3209].

The new primary LSP can also share resources with the old backup LSP. The association of these two LSPs is ensured thanks to the identifier of the protected primary LSP, carried within the *Path* message for the backup LSP.

4.3.2.3.5 Protection resources sharing

Two *Egress FRR* backup LSPs that protect distinct primary Egress LSRs, can share bandwidth, as they will normally not be activated simultaneously (assuming only single failure scenario). In such case the reserved bandwidth is not the sum but the maximum of the two LSP bandwidths.

4.3.2.3.6 Example

Figure 38, Figure 39, Figure 40, and Figure 41 below illustrate with an example, the setup of a primary LSP from R1 to R4, protected with *Egress FRR*. The backup Egress LSR is R6 and the penultimate LSR is R3.

The primary LSP, LSP1, is configured by the operator on the Ingress LSR R1, directly or thanks to a NMS. The configuration includes, in addition to basic MPLS-TE parameters: the request for *Egress FRR* and the IP address of the backup Egress LSR, R6.

R1 computes a primary path and an *Egress FRR* backup path that respect the TE constraints (bandwidth, affinities…), and then starts RSVP-TE signalling, by sending a *Path* message that includes, in addition to basic RSVP-TE objects, the request for Egress FRR, the IP address of the backup Egress LSR R6, and the *Egress FRR* backup LSP path (R3-R5-R6).

On receipt of this *Path* message the Primary Egress LSR R4 sends a *Resv* following normal RSVP-TE procedures.

On receipt of this *Path* message, the LSR R3 detects that it is the PUL by checking the remaining hops in the ERO. It starts the setup of a backup LSP, LSP2, towards the backup Egress LSR R6. For that purpose, it sends a *Path* message with, as ERO, the backup path included in the received *Path* message for the primary LSP. This *Path* message also includes the indication that this is an *Egress FRR* Backup LSP, along with the address of the primary egress LSR R4 and the identifier of the protected primary LSP LSP1 (see Figure 38).

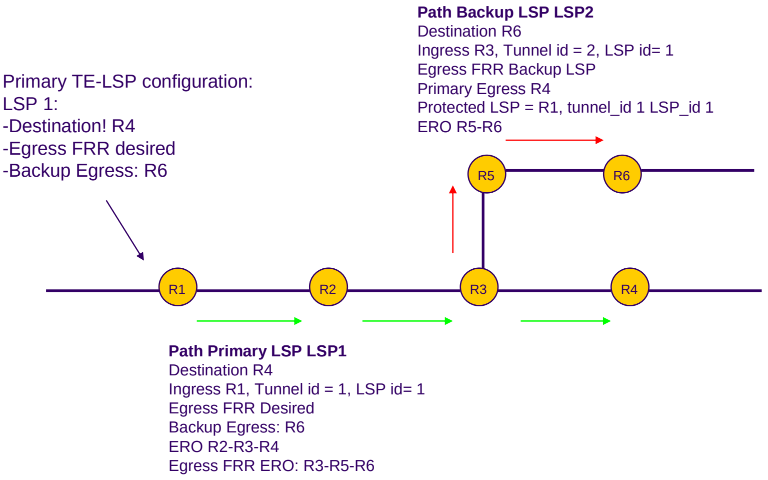


Figure 38 Signalling of primary and backup LSPs: Path message

The Backup Egress LSR R6, allocates a non null label, (32 in this example), for the backup LSP, and sends a *Resv* message on the backward direction towards the PUL. R6 installs this label within its MPLS Forwarding table and it is mapped to a context specific FIB, that avoids the protected primary Egress LSR R4 (see Figure 39 and Figure 40, "FIB (avoid R4)"). This FIB is built from the IP RIB; RIB routes whose next hop is LSR R4 are not taken into account when building this context specific FIB.

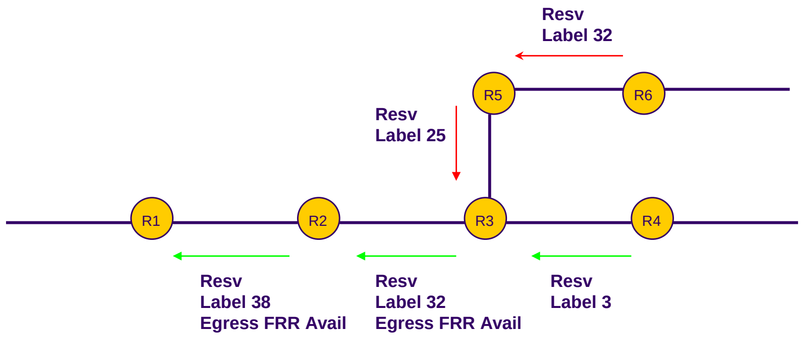


Figure 39 Signalling of primary and backup LSPs: Resv message

On receipt of the *Resv* messages for the primary and backup LSPs, the PUL sends a *Resv* message towards the Ingress LSR, indicating that *Egress FRR* is available.

Figure 40 illustrates the content of the IP and MPLS forwarding tables, and the forwarding of IP/MPLS packets towards 1.1/16, reachable via the primary Egress LSR R4 and the backup Egress LSR R6, before R4 failure.

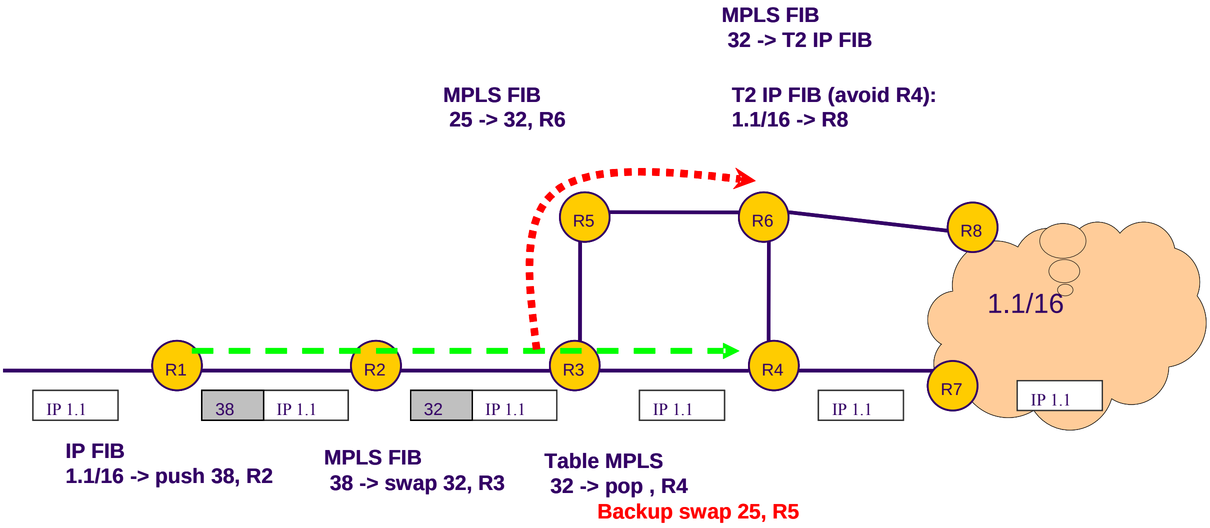


Figure 40 Packet forwarding before failure

The MPLS table on R3 includes two outputs for the primary LSP label:

- A primary output towards the primary Egress LSR R4.

- A backup output within the backup LSP, towards the backup Egress LSR R6.

Figure 41 illustrates the content of the IP and MPLS forwarding tables, and packet forwarding during R4 failure. The PUL is redirecting traffic from the primary LSP LSP1 to the backup LSP LSP2, towards the backup Egress LSR R6.

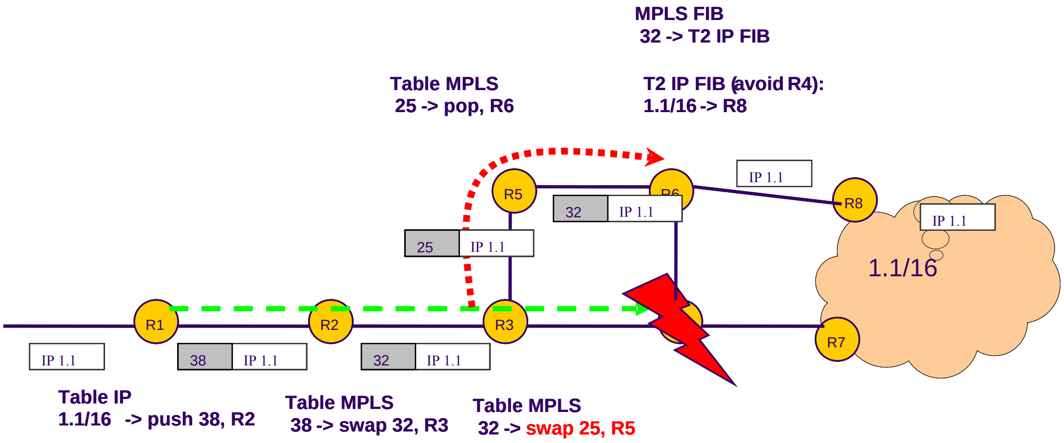


Figure 41 Packet forwarding during failure

On R6, packets are forwarded within the context specific FIB that avoids R4, they are forwarded directly to their destination.

4.3.2.4 ASBR Protection with RSVP-TE Egress Fast Reroute

The RSVP-TE *Egress FRR* protection mechanism can be used to ensure ASBR protection. This requires extensions to the BGP next-hop resolution process.

To protect a given prefix P against failure of the downstream ASBR on the path, an ASBR or Edge router (running iBGP) needs to learn at least two routes for the prefix, that is two downstream ASBRs through which the prefix is reachable. This is natively supported if no Route Reflector is used. Else, this requires BGP extensions such as those proposed in [MULTI-NEXTHOP].

The BGP next-hop resolution process on an edge router E, for a prefix P reachable via a set S of at least two downstream ASBRs must be extended as follows:

- Find the best next-hop B1 in S following standard BGP selection procedures.
- Find the best next-hop B2 in S minus {B1} following standard BGP selection procedures.
- Find a TE-LSP LSP1 protected by *Egress FRR* whose Primary and Backup Egress LSRs are B1 and B2
- Install P within LSP1

In nominal situation the traffic towards P traverses the ASBR B1.

Upon B1 failure the traffic is fast rerouted towards the backup downstream ASBR B2 within 50ms.

After a few seconds, the BGP session with the failed protected ASBR B1 is deleted (after the expiration of the BGP deadtimer (BGP keepalives no longer received from B1)), and the routes advertised by this ASBR are removed. Also the TE-LSP metric may be set to INFINITY. In case of any of these two events, the BGP next hop should be changed in a make before break manner. A new next hop B' (not necessarily B2) is selected as best nexthop and a TE-LSP LSP3 towards B' (potentially *Egress FRR* protected) is selected for P. The route for P in the FIB is then replaced in an atomic manner (LSP1 replaced by LSP3), so as to minimize traffic disruption.

This approach can be used to protect Egress ASBRs and Ingress ASBRs. In case of Egress ASBR protection, a mesh of *Egress FRR* protected TE-LSP is setup between Ingress and Egress ASBRs (see figure 1).

In case of Ingress ASBR protection a one hop *Egress FRR* protected TE-LSP is setup between the Egress ASBR and the downstream Ingress ASBR (see figure 2).

4.3.2.5 Conclusion

Sub-100ms recovery upon link and node failure is a key requirement for mission critical services such as VoIP or Telemedicine [ROUX06]. The MPLS Fast Reroute mechanism defined in [RFC4090] is a powerful tool that allows for sub-50ms recovery upon link or node failure. It is widely deployed today for intra-AS protection. Protection against ASBR failures requires today and end-to-end inter-AS LSP [RFC4216], and this does not scale very well for a large number of ASBRs. To overcome these scalability limitations we define here a new FRR mechanism that does not requires end-to-end inter-AS TE LSPs. It relies on a new RSVP Fast Reroute mechanism called Egress FRR that allows protecting the Egress LSR of a TE-LSP. A backup LSP from the penultimate LSR to a backup Egress LSR is setup and upon primary Egress LSR failure, the penultimate LSR redirects the traffic within the backup LSP towards the backup Egress LSR. On the backup Egress LSR a context specific forwarding is performed so as to avoid the traffic to be redirected to the primary Egress LSR. To protect against ASBR failures, an upstream installs a prefix reachable via two downstream ASBRs, within an Egress FRR protected LSP whose primary and backup Egress LSRs are these downstream ASBRs. This allows ensuring sub-50ms recovery upon ASBR node failures and inter-AS link failures in an inter-AS path.

4.3.3 Robust Egress point selection

4.3.3.1 Introduction

Inter-domain Outbound Traffic Engineering (TE) [FEAM03, BRES03] aims to control traffic exiting a domain by assigning the traffic to the best egress points (i.e. routers or/and links). Since inter-domain links are the most common bottlenecks in the Internet [BRES03], optimizing their resource utilization is a key objective of outbound TE. In the literature, several outbound TE approaches have been proposed [BRES03, HO04]. These proposals, however, have neglected the detrimental impact of inter-domain link failure on the achieved TE performance. In fact, the network performance under failure conditions should ideally be optimized by considering failure as part of the outbound TE optimization.

Failure occurs as part of daily network operations [NUCC03, SRID05]. Inter-domain failures are typically caused by: (1) *physical failures* such as inter-domain link fibre cut and equipment failure, or (2) *logical failures* such as router CPU overload, operation systems problem and maintenance. A recent study [BONA05] discovered that logical inter-domain link failures are common events and are usually transient in nature. When a failure happens on an egress point (EP), traffic is shifted to another available EP in accordance to the BGP route selection policies. However, if a large amount of traffic is shifted, congestion is likely to occur on these new serving EPs. This problem has not been considered in the existing outbound TE proposals. An intuitive approach to minimize this congestion is to redirect the traffic to another EP by adjusting BGP routing policies in an online manner until the best available EP has been found. Such online trial-and-error approach may cause router misconfiguration, unpredicted traffic disruption and flooding of BGP route advertisements, leading to route instability and slow convergence. As a result, a systematic outbound TE approach that produces optimal performance under both normal and failure scenarios so as to minimize online and unpredictable route changes is highly desirable.

Hence, we propose an offline outbound TE approach that enhances the robustness of the existing NPs which use the BGP protocol for inter-domain routing. More specifically, our approach is not used to design a specific NP, but it can be "replicated" to individual NPs that apply the BGP routing protocol. Note that, our approach is expected to achieve reasonably good traffic engineering performance under both Normal State (NS, i.e. no inter-domain link failure) and Failure States (FS, i.e. single inter-domain link failure).

4.3.3.2 *Overview of the Objective and Design*

The purpose of this section is to explain our objective and describe the overall design (i.e. inputs and outputs) of our problem. Our robust egress point selection problem is an optimization problem that aims to determine a primary and a secondary egress point for each destination prefix such that this egress point selection minimizes the maximum inter-domain link utilization under NS and the average of maximum inter-domain link utilization across all FSs. Note that since *single* link failure is the predominant form of failure in communication networks [NUCC03], we therefore only compute a primary and a secondary egress point per destination prefix (i.e. no need to compute a tertiary egress point since the primary and secondary egress points would not fail simultaneously).

To achieve our objective, the NP provisioning and maintenance functional block encompasses an offline inter-domain outbound TE optimizer component. The task of this component is to optimize the primary and secondary egress point selection.

In this section we specifically address the outbound TE problem by only taking into account traffic optimisation across inter-domain links. A more general scenario will be described in section 4.3.4 where both intra- and inter-domain topologies will be considered. In this section, since our objective is to demonstrate the principle of robust outbound TE, we apply our work to the single egress selection case and on a general network model where each EP is composed of an egress router attached to a single inter-domain link. In this case, EP failure and EP utilization, in fact refer to inter-domain link failure and inter-domain link utilization respectively.

According to the above explanations, the offline inter-domain outbound TE optimizer component requires three inputs: (1) the physical inter-domain topology that contains information on ASBR connections and inter-domain link capacities (2) inter-domain traffic matrix based on the subscribed CPA and NIA demands (from the NP mapping and NIA order handling functional blocks), (3) remote destination prefixes and their reachability information. The outputs of this component are: (1) a set of primary egress points (PEP) that determine the egress points under Normal State (NS, i.e. no inter domain link failure) and (2) a set of secondary egress points (SEP) that determine the egress points under Failure States (FS, i.e. single inter-domain link failure).

4.3.3.3 *Problem Formulation*

NOTATION	DESCRIPTION
K	A set of destination prefixes, indexed by k
L	A set of egress points, indexed by l
S	A set of states $S=\{\varnothing U (\forall\ l \in\ L)\}$, indexed by s
I	A set of ingress points, indexed by i
$t(k,i)$	Bandwidth demand of traffic flows destined to destination prefix $k \in K$ at ingress point $i \in I$
$Out(k)$	A set of egress points that have reachability to destination prefix k
c_{inter}^{l}	Capacity of the egress point l
x_{sk}^{l}	A binary variable indicating whether prefix k is assigned to the egress point l in state s
u_{s}^{l}	Utilization on non-failed egress point l in state s. Its value is zero when $s=l$
$U_{max}(s)$	maximum egress point utilization in state s
U_{Ave}^{FS}	Average of maximum egress point utilization across all failure states

Table 10. Notation used for the robust egress point selection problem

In this section, we present our robust egress point selection optimization problem formulation. Table 10 shows the notation used in this paper.

Each element of the inter-domain TM, $t(k,i)$, represents the total volume of traffic from ingress point i towards destination prefix k. Due to the increasing use of multihoming, a prefix usually can be reached through multiple EPs, thereby allowing outbound TE to select the best EP for the traffic. Given an inter-domain topology, destination prefixes together with their reachability information and an inter-domain TM, the goal of our optimization problem is to determine, for each destination prefix, both a PEP under NS ($s=\varnothing$) and a SEP that will serve the traffic when the PEP has failed (i.e. under FS). The optimization objective is to minimize both the maximum EP utilization under NS and the average maximum EP utilization across all FSs. Recall that each FS corresponds to a single EP failure. The number of FSs is hence equal to the number of inter-domain links $|L|$. By adding the NS, the total number of states $|S|$ is $|L| + 1$. The computational complexity of our problem is thus an increasing function of the total number of states. To reduce this complexity, one may take the idea in [SRID05] of performing the TE only on a small subset of FSs whose failures have significant impact on network performance. This set of EPs is referred to as *critical* EPs but we leave this as future work. The maximum EP utilization under state s can be calculated as:

$$\forall s \in S : Minimize\, U_{\max}(s) = Minimize \underset{\forall l \in L/\{s\}}{Max} (u_s^l) = Minimize \underset{\forall l \in L/\{s\}}{Max} \left(\frac{\sum_{k \in K}\sum_{i \in I} x_{sk}^{l} t(k,i)}{c_{\text{inter}}^{l}}\right) \qquad (1)$$

Under any FS s, the term $x_{sk}^{l}t(k,i)$ consists of flows which are assigned to EP l as their PEP and also flows which are assigned to EP l as their SEP. Clearly, under NS ($s=\varnothing$), the term only includes the former.

Since our optimization objective is to minimize the maximum EP utilization under both NS and FSs simultaneously, a bi-criteria optimization problem is formed. However, the two optimization objectives conflict with each other and hence we resort to a weighted sum approach to transform them into a single-criterion optimization problem, which is simpler to solve. The optimization objective function is thus:

$$Minimize\ F=(1\text{-}w)U_{max}(\varnothing)+wU_{Ave}^{FS}\ ,\ 0 \le w \le 1 \qquad (2)$$

$$\text{where } U_{Ave}^{FS} = \underset{\forall s \in S/\{\varnothing\}}{Ave} (U_{\max}(s)) = \frac{\sum_{s \in S/\{\varnothing\}} U_{\max}(s)}{|S|-1} \qquad (3)$$

subject to the following constraints:

$$\forall l \in L, k \in K, s \in S \quad if \quad x_{sk}^{l} = 1 \quad then \quad l \in Out(k) \tag{4}$$

$$\forall k \in K, s \in S : \sum_{l \in Out(k)} x_{sk}^{l} = 1 \tag{5}$$

$$\forall l \in L, k \in K, s \in S : x_{sk}^{l} \in \{0,1\} \tag{6}$$

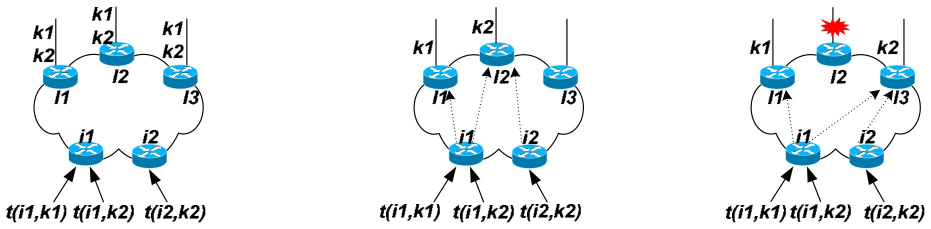
$$\forall l \in L, k \in K \quad if \quad x_{\varnothing k}^{l} = 1 \quad then \begin{Bmatrix} x_{sk}^{l} = 1 & \forall s \in S / \{l\} \\ x_{sk}^{l} = 0 & \forall s = l \end{Bmatrix} \tag{7}$$

By varying weight *w* and re-solving *F*, one can generate a trade-off curve between the two objectives using the weighting method of multi-objective programming [COHO78]. If we solve the problem with *w=0*, the problem is simply reduced to the PEP selection problem. If *w=1*, the problem then completely ignores the performance under NS. We present results for *w=0.5* (i.e. equal weight to the objectives optimized under NS and FS), which allows us to achieve significant performance improvement for SEP selection with only a small performance degradation for the PEP selection. Constraint (4) ensures that if prefix *k* is assigned to EP *l* under either NS or any of the FSs, then this prefix is reachable through EP *l*. Constraints (5) and (6) ensure each destination prefix is assigned to only one PEP under NS ($s=\varnothing$) and only one SEP under FSs. Constraint (7) ensures that if prefix *k* is assigned to EP *l* under NS, then this prefix remains on *l* for all the FSs except when the current FS is the failure on *l*.

According to [BRES03], the primary (single) egress point selection problem considering the inter-domain link capacity constraint has been proven to be NP-hard by reducing it to the Generalized Assignment Problem (GAP), which is itself NP-hard. Considering our problem, if we set either the number of FSs or the weighting parameter to zero, our optimization problem is reduced to the uncapacitated version of the primary outbound TE problem in [BRES03]. As a result, our optimization problem is an extension version of [BRES03] and therefore is NP-hard. Hence, we resort to using a heuristic approach to solve the problem.

4.3.3.4 The Primary and Secondary Egress Point Selection Example

For better understanding of our robust egress point optimization problem, we provide an example in Figure 42(a)-(c). Figure 42(a) shows all the inputs to the problem, which includes ingress points *i1* and *i2*, egress points *l1*, *l2* and *l3*, traffic demands *t(i1,k1)*, *t(i1,k2)* and *t(i2,k2)* and destination prefixes *k1* and *k2* that can be reached through all the three egress points. Recall that the task of our optimization problem is to determine for each destination prefix, *both* an EP as its PEP so that inter-domain traffic (independent from any ingress point) will exit the domain from that point under NS *and* an EP as its SEP so that inter-domain traffic (independent from any ingress point) will exit the domain from that point when its PEP has failed (i.e. under FS). Figure 42(b) shows a potential solution of PEP selection, having *k1* and *k2* been assigned to egress points *l1* and *l2* respectively. As a result, the PEP for all the traffic demands destined to *k1* is *l1* and to *k2* is *l2* i.e. $PEP_{t(i1,k1)}$ → *l1*, $PEP_{t(i1,k2)}$ → *l2* and $PEP_{t(i2,k2)}$ → *l*2. In addition, Figure 42(c) illustrates a potential solution of SEP selection when EP *l2* has failed. As shown, *k2* has been re-assigned to egress point *l3* as its SEP. As a result, the SEP for all the traffic demands destined to *k2* are assigned to *l3*, i.e. $SEP_{t(i1,k2)}$ → *l3* and $SEP_{t(i2,k2)}$ → *l3*. Note that the traffic demand headed towards the unaffected destination prefix (i.e. *k1*) has remained intact.


k1
k2
k1
k2
k1
k2
l1
l2
l3
i1
i2
t(i1,k1) t(i1,k2) t(i2,k2)
k1
k2
l1
l2
l3
i1
i2
t(i1,k1) t(i1,k2) t(i2,k2)
k1
k2
l1
l2
l3
i1
i2
t(i1,k1) t(i1,k2) t(i2,k2)

(a) (b) (c)

Figure 42. (a) Outbound TE inputs, (b) PEP Selection and (c) SEP Selection for *k2*

4.3.3.5 Proposed Tabu Search Heuristic

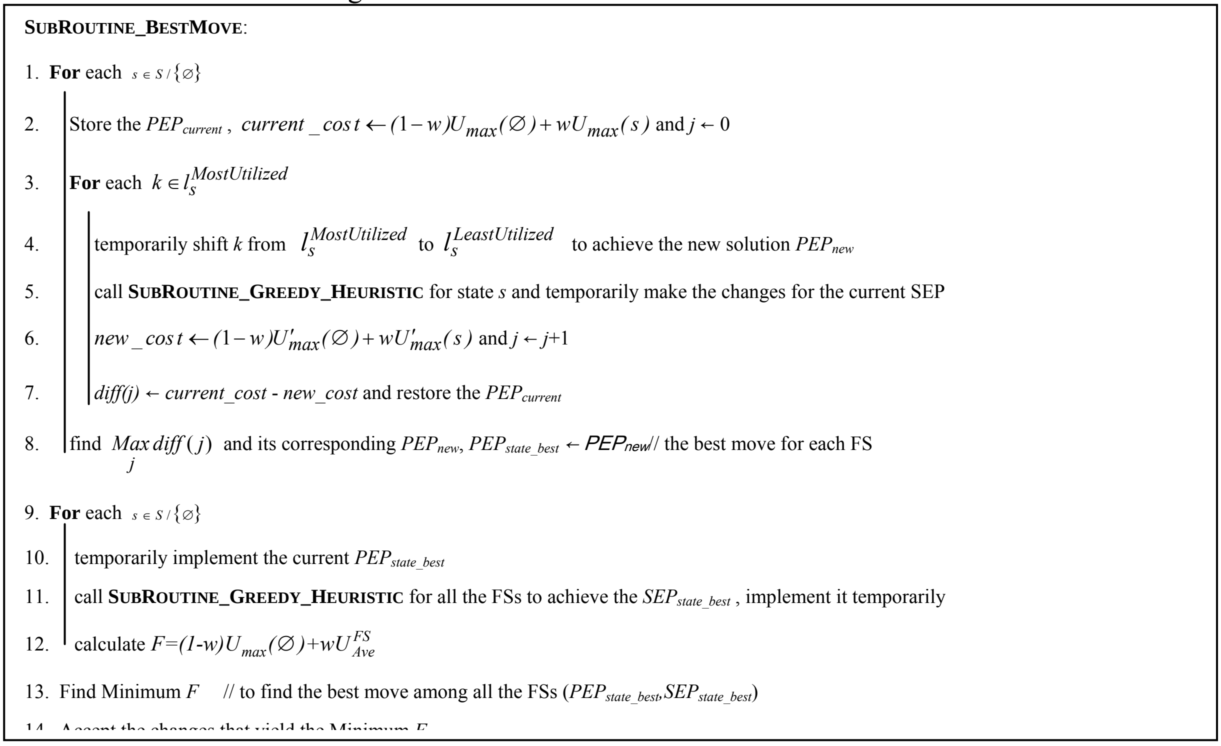
The Tabu Search (TS) methodology [GLOV97] guides local search methods to overcome local optimality and attempts to obtain near-optimal solutions for NP-hard optimization problems. Due to space limitations, the reader is referred to [GLOV97] for an overview of TS. In general, our proposed TS heuristic first requires initial PEP and SEP selection solutions, and then proceeds to obtain neighbor solutions by using a neighborhood search strategy in order to gradually enhance the quality of the initial solution.

4.3.3.5.1 Non-TE initial solution

We obtain initial PEP and SEP selection solutions by randomly selecting EPs for the destination prefixes while satisfying constraints (4) to (7). These initial solutions can be regarded as non-TE (i.e. non-optimized) solutions. The rationale of using such initial solutions is to demonstrate the effectiveness of the proposed TS heuristic in producing good performance from poorly performing initial solutions.

4.3.3.5.2 Neighborhood Search Strategy

A *move* transforms the current (initial) solution into a neighbor solution. To perform a move, we apply the SUBROUTINE_BESTMOVE heuristic shown in Figure 43, to first identify the best move for each FS and then select the best one among all the FSs.



```
SUBROUTINE_BESTMOVE:
1. For each $s \in S/\{\varnothing\}$
2.   Store the $PEP_{current}$, $current\_cost \leftarrow (1-w)U_{max}(\varnothing)+wU_{max}(s)$ and $j \leftarrow 0$
3.   For each $k \in l_s^{MostUtilized}$
4.     temporarily shift $k$ from $l_s^{MostUtilized}$ to $l_s^{LeastUtilized}$ to achieve the new solution $PEP_{new}$
5.     call SUBROUTINE_GREEDY_HEURISTIC for state $s$ and temporarily make the changes for the current SEP
6.     $new\_cost \leftarrow (1-w)U'_{max}(\varnothing)+wU'_{max}(s)$ and $j \leftarrow j+1$
7.     $diff(j) \leftarrow current\_cost - new\_cost$ and restore the $PEP_{current}$
8.   find $\underset{j}{Max}\, diff(j)$ and its corresponding $PEP_{new}$, $PEP_{state\_best} \leftarrow PEP_{new}$// the best move for each FS
9. For each $s \in S/\{\varnothing\}$
10.  temporarily implement the current $PEP_{state\_best}$
11.  call SUBROUTINE_GREEDY_HEURISTIC for all the FSs to achieve the $SEP_{state\_best}$ , implement it temporarily
12.  calculate $F=(1-w)U_{max}(\varnothing)+wU_{Ave}^{FS}$
13. Find Minimum $F$    // to find the best move among all the FSs ($PEP_{state\_best}$,$SEP_{state\_best}$)
14. Accept the changes that yield the Minimum F
```

Figure 43. SUBROUTINE_BESTMOVE

The following steps explain how to identify the best move for each FS:

Step 1. Store the currently assigned PEP for all prefixes in $PEP_{current}$. Calculate the *current_cost*, i.e. the weighted sum of the maximum EP utilization under both NS and the current FS (Figure 43 line 2). List all the prefixes in $PEP_{current}$ assigned to the Most Utilized EP under the current FS ($l_s^{MostUtilized}$)[1].

[1] $l_s^{MostUtilized}$ is the link that has $\underset{\forall l \in L/\{s\}}{Max}\ u_s^l$

Consider each prefix at a time in the list and apply steps 2 to 4 until all the destination prefixes in the list have been considered (Figure 43 lines 3 to 7).

Step 2. Shift the prefix's PEP from $l_s^{MostUtilized}$ to the Least Utilized EP ($l_s^{LeastUtilized}$)[2] (the goal of this move is to attract traffic towards the $l_s^{LeastUtilized}$ and potentially to reduce the load on the $l_s^{MostUtilized}$). This results in a new solution for the PEP selection, which is denoted by PEP_{new}.

Step 3. Reassign the SEPs for the destination prefixes that have been assigned to the failed EP by using the **SubRoutine_Greedy_Heuristic** algorithm. The algorithm works as follows: (a) Sort all the destination prefixes on the failed EP by descending volume of traffic. (b) Take the first of these ordered prefixes and select as its SEP the available EP with the minimum utilization. (c) Repeat step (b) for the rest of the destination prefixes in order.

Step 4. Calculate the *new_cost* in the same way as the *current_cost* for the latest solution (Figure 43 line 6). Then calculate the difference between the *current_cost* and *new_cost* (i.e. *diff* = *current_cost-new_cost*). Restore the $PEP_{current}$.

Step 5. Identify the prefix that produces the largest value of *diff* (i.e. largest difference between the *current_cost* and *new_cost*). Consider the PEP_{new} that corresponds to this prefix as the best move for the current FS. Store this PEP_{new} in PEP_{state_best}.

Step 6.Repeat steps 1 to 5 for each FS and identify their PEP_{state_best} until all the FSs have been considered (Figure 43 lines 1 to 8).

After identifying the best move for each FS, we now identify the best of the best moves for all FSs by the following steps:

Step 1. For the best move for each FS, reassign the SEPs (SEP_{state_best}) for the corresponding PEP_{state_best} by using the **SubRoutine_Greedy_Heuristic** algorithm for all the FSs. (this calls the subroutine *s* times, once for each FS). Calculate objective function (2). Repeat step 1 for the best move of the next FS until all the FSs have been considered (Figure 43 lines 9 to 12).

Step 2. For all the FSs evaluated in step 1, choose the best move (i.e the PEP_{state_best} and its corresponding SEP_{state_best}) that yields the minimum objective value (Figure 43 lines 13-14).

4.3.3.5.3 Tabu List

The tabu list is a memory list that memorizes the most recent moves, operating as a first-in-first-out queue. As suggested in [GLOV97], the size of the tabu list depends on the size and characteristics of the problem. Since in our algorithm the attributes of a move are the highly and lightly utilized EPs, and shifted destination prefixes, the size of the tabu list is determined by the number of destination prefixes. We define the size of the tabu list to be *total number of destination prefixes / |L|*.

4.3.3.5.4 Diversification

The goal of diversification is to prevent the searching procedure from indefinitely exploring a region of the solution space that consists of only poor quality solutions. It is a modification of the neighbourhood searching strategy and is applied when there is no obvious performance improvement after a certain number of iterations. For diversification, a group of highly and lightly utilized EPs are chosen for shifting destination prefixes under a FS. We define the threshold of obvious performance improvement to be 10% of the best visited solution and the number of iterations to be 10% of the maximum iteration mentioned below.

4.3.3.5.5 Stopping Criterion

Many stopping criteria can be developed depending on the nature of the problem. The most common criterion, used in this paper, is to define a maximum number of iterations. However, we do not

[2] $l_s^{LeastUtilized}$ is the link that has $\underset{\forall l \in L/\{s\}}{Min} u_s^l$

arbitrary select the number of maximum iterations since the performance of the TS heuristic mainly depends on how many times the PEPs and SEPs are reassigned. We found that setting the maximum iteration number to be 5 times the number of destination prefixes gives us sufficiently good results.

4.3.3.6 Alternative Strategies

Our proposed TS heuristic is only one of several approaches in solving the robust egress point selection problem. In this section, we present three alternative approaches. For these approaches, **OPTIMAL-AWARE HEURISTIC** is used for the PEP selection and the three alterative approaches only differ in their SEP selection. We remark that the **OPTIMAL-AWARE HEURISTIC** is our best attempt in solving our PEP selection problem, as no algorithm for solving the problem with objective function (1) has been proposed in the literature. The **OPTIMAL-AWARE HEURISTIC** works as follows:

Step 1: Calculate the mean utilization by dividing the total traffic volume by the total capacity of all EPs. We regard this mean utilization as the theoretical optimal (i.e. the most load balanced) utilization targeted for each EP to achieve. However, this theoretical result is not a valid solution because it allows arbitrary traffic splitting over any EP, violating constraints (5) and (6). Nevertheless, it is used as an "NS lower bound" solution[3] for comparing performance with other strategies.

Step 2: To ensure that each EP does not exceed the theoretical optimal utilization, set the mean utilization as a capacity constraint on each EP.

Step 3: Sort the destination prefixes in descending order according to the amount of traffic they carry and choose one at a time in order.

Step 4: Select the EP with the minimum utilization as the PEP of this destination prefix if it satisfies the capacity constraint, if not proceed to the next prefix. Repeat this step until all the destination prefixes have been considered.

Step 5: If there exist unassigned destination prefixes because of capacity constraint violation, re-run step 4 without considering the capacity constraint.

4.3.3.6.1 Random Reassignment Strategy

In the Random Reassignment (**RandomR**) strategy, when an EP fails, the destination prefixes on the failed EP are re-assigned to other available but *randomly* chosen EPs. This strategy can be regarded as an approach that ignores the impact of failure on outbound inter-domain TE performance. We illustrate an example of the **RandomR** in Figure 44. In this example there are three EPs (*l1*, *l2* and *l3*) with egress link capacity 200, 100, 150 Mbps respectively and an ingress point *i*. The input traffic flows and their traffic volume are shown in Table 11. Figure 44(a) shows a solution of the PEP selection, which can be generated by the **OPTIMAL-AWARE HEURISTIC**. The solution has the best load balancing over all the EPs (i.e. $u_{\varnothing}^{l1} = \frac{80+10+10}{200} = 0.5$, $u_{\varnothing}^{l2} = \frac{40+10}{100} = 0.5$ and $u_{\varnothing}^{l3} = \frac{60+10+10}{150} = 0.533$). Figure 44(b) shows the solution of the SEP selection under EP *l1* failure produced by the **RandomR**. The figure demonstrates that when EP *l1* is assumed to fail, destination prefixes *k1*, *k4* and *k6* are then randomly assigned to EP *l2* and *l3* as their SEPs. This random assignment, however, causes heavy load on EP *l2* which could easily lead to congestion (e.g. $u_{l1}^{l2} = \frac{40+10+80+10}{100} = 1.4$, $u_{l1}^{l3} = \frac{60+10+10+10}{150} = 0.6$). Therefore, the **RandomR** performs poorly under any FS since no optimization is taken into account for FSs. Nevertheless, since only the affected destination prefixes are reassigned, the level of traffic disruption is minimized (i.e. only prefixes *k1*, *k4* and *k6* are disrupted when EP *l1* fails).

4.3.3.6.2 Global Reassignment Strategy

In the Global Reassignment (**GlobalR**) strategy, for any EP failure, the **OPTIMAL-AWARE HEURISTIC** is reapplied to perform PEP selection from scratch by excluding the failed EP. Such network-wide computation can be regarded as the best approach with respect to performance but possible large traffic disruption because the PEPs for most of destination prefixes are likely changed. We use the

[3] We can define the "FS lower bound" in a similar fashion. First for each FS we calculate the total volume of traffic divided by the capacity of all EPs excluding the failed one, and then choose the maximum (i.e. the worst case) as the FS lower bound.

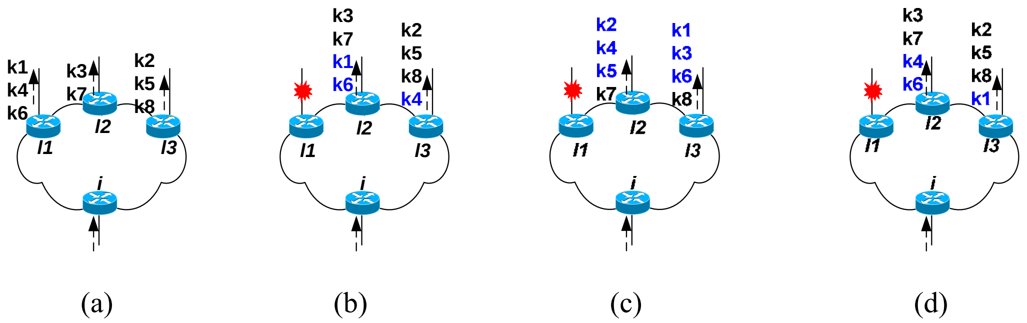
GlobalR as a reference point for evaluating the performance of other strategies. Figure 44(c) shows the result of the **GlobalR** based on the PEP selection solution shown in Figure 44(a). As can be seen, when EP *l1* fails, some prefixes are reassigned away from their original EPs even though failure has occurred on another EP. For example, *k2* and *k5* are shifted from EP *l3* to *l2* while *k3* is shifted from EP *l2* to *l3*. Nevertheless, the utilization upon any EP failure is optimal (i.e. $u_{l1}^{l2} = \frac{60+10+10+10}{100} = 0.9, u_{l1}^{l3} = \frac{80+40+10+10}{150} = 0.933$).

4.3.3.6.3 Greedy Reassignment Strategy

In the Greedy Reassignment (**GreedyR**) strategy, for any EP failure, only the destination prefixes assigned on the failed EP are re-assigned by a greedy heuristic as follows: the destination prefix that carries the largest amount of traffic is reassigned to the available EP that has the lowest utilization. This step repeats for the rest of the affected prefixes. The GreedyR strategy can be regarded as a simple approach of handling failures that might be taken by ISPs. Figure 44(d) shows the result of the **GreedyR** based on the PEP selection solution shown in Figure 44(a). As can be seen, the greedy reassignment of prefixes can provide a better load balancing compared to the random reassignment however, not as good as the **GlobalR** (i.e. $u_{l1}^{l2} = \frac{40+10+10+10}{100} = 0.7, u_{l1}^{l3} = \frac{60+10+10+80}{150} = 1.06$). Also, regarding traffic disruption this strategy performs identical to the **RandomR** which keeps the disruption to a minimum (i.e. only prefixes *k1*, *k4* and *k6* are disrupted when EP *l1* fails).

TRAFFIC FLOW	TRAFFIC VOLUME(MBPS)
t(k1,i)	80
t(k2,i)	60
t(k3,i)	40
t(k4,i)	10
t(k5,i)	10
t(k6,i)	10
t(k7,i)	10
t(k8,i)	10

Table 11. Input traffic flows



(a) (b) (c) (d)

Figure 44. Different algorithms for destination prefix assignment

4.3.4 Intra-/inter-domain interactions in terms of resilience

4.3.4.1 Introduction

In general, TE is used to optimize the INP operational network performance. In fact, it is a set of techniques that route the traffic on a path other than the shortest path chosen by the standard routing methods to achieve load balancing and optimize the overall network performance.

Due to the rapid increase in the number of domains and the amount of traffic across domains, it is important for each INP to optimize its TE performance not only within its own domain but also between its domain and its neighbours. In other words, an INP needs to consider both intra- and inter-domain TE optimization [HO06]. In intra-domain TE, the INP controls traffic routing within its own domain by either optimizing the IGP link weights or setting up edge-to-edge LSPs. The common intra-domain TE objectives are to minimize the total resource consumption and/or achieve load balancing within a domain. On the other hand, in inter-domain TE, the INP controls traffic entering and exiting its domain through the BGP policies to achieve load balancing over its inter-domain links.

In the intra- and inter-domain TE interactions the concept of *hot-potato* routing [AGAR05] is important. Hot-potato occurs when there are multiple “equally good egresses” (i.e. BGP routes with identical *local-pref* and *AS path length*) to reach a remote destination. In this case, the ingress router selects the closest egress point, based on the IGP link weights. Any change in the IGP link weights due to traffic-engineering, link failures or planned maintenance would cause hot-potato routing changes (i.e. change of egress points). This change can have significant performance impacts such as: (i) traffic disruption, (ii) a large traffic shifts that may cause congestion and performance degradation within and between the domains and also (iii) transient packet delay and loss while the routers re-compute their forwarding tables and etc. As a result, an intra-domain link failure may lead to detrimental hot-potato routing changes consequences. Moreover, as mentioned in section 4.3.3 inter-domain link failure can cause huge TE performance degradation. Therefore, due to the fact that intra-/inter- domain link failures are common and transient events [NUCC03, BONA05], investigating their impacts to minimize their consequences and achieve reasonably good TE performance is an important issue.

Prior robust intra-domain TE methods [NUCC03, SRID05, FORT02] have computed a set of IGP link weights that is robust to any single intra-domain link failure only within their domain and have ignored the detrimental failure impacts on their inter-domain TE performance. Also, as mentioned in section 4.3.3.1 existing inter-domain TE methods [BRES03, HO04] have not considered the failure impacts in their optimization methods. Since all these methods have optimized either pure intra-domain or inter-domain TE objectives, therefore, in case of any intra or inter-domain link failure, their overall TE performance may be suboptimal or even very poor. However, [HO06] has investigated the interactions between intra- and inter-domain TE and proposed a joint optimization and also [AGAR05] has evaluated the behaviour of hot-potato routing during their IGP link weight optimization. Nevertheless, none have investigated the impacts of intra-domain link failure on inter-domain TE performance and inter-domain link failure on intra-domain TE performance.

Hence, we propose an offline joint intra-/inter-domain TE approach that enhances the robustness of the existing NPs which use the IGP/BGP protocols, including their multi-topology extensions, for both intra- and inter-domain routing. In fact, similar to section 4.3.3, our approach is not used to design a specific NP, but it can be “replicated” to individual NPs that apply the IGP and BGP routing protocol. In fact, our approach investigates the potential failure impacts of intra- and inter-domain links and is expected to minimize the impacts while achieving reasonably good TE performance under both Normal State (NS, i.e. no intra- or inter-domain link failure) and Failure States (FS, i.e. single intra- or inter-domain link failure).

[[Please note that detailed information of on the overview of the objective and design has been suppressed in the public version of this document but will be published in a later stage.]]

5 SERVICE ENGINEERING

5.1 VoIP service engineering

The AGAVE project has identified VoIP service as a use case to drive the project IP Connectivity related specifications and to check the validity of proposed solutions. Thus, the project has identified a set of requirements to be met by both INPs and VoIP Service Provider. This effort has been documented in [D1.1]. One of the important conclusions of the analysis of these requirements is that in order to be able to offer robust, QoS-able and highly available VoIP Services, both network layer and service layer should be synchronised. New Service-based Traffic Engineering means should be implemented by the VoIP Service Providers (e.g. IP telephony routing, telephony load balancing, etc.).

The effort of designing, specifying and validating these Service-based TE techniques is out of scope of the AGAVE project. Nevertheless, examples to illustrate the role of such techniques may be enclosed in D3.2.

5.2 VoIP monitoring

This section specifies the framework for monitoring mechanisms destined to conversational services (mainly Voice over IP and Videotelephony over IP) within the context of the AGAVE functional architecture as defined in WP1 [D1.1] and using the monitoring architecture.

In the remaining part of the document, the terms VoIP and Conversational Services are used interchangeably. Indeed, the following discussion is valid for Voice and Video.

5.2.1 Objectives

As defined in WP1 [D1.1], VoIP monitoring should meet the following requirements:

- (R1) Customers should have the ability to verify the fulfilment of the SLA they subscribed to. *Indicators such as availability of the service, success rate of placed calls, number of failures that happened over the last period, Voice quality could be correlated with billing tickets*;
- (R2) A VoIP service provider should have means to monitor the usage of each SIA and whether a service peer meets its contractual commitments. After prior agreement between two service peers about *monitoring methodology, templates and data, indicators such as availability of the service, success rate of placed calls, number of failures that happened over the last period, Voice quality and network transmission parameters such as delay, jitter and loss could be exchanged between them. Billing tickets can then be correlated to the monitored indicators values. Network transmission indicators apply to the transmission of VoIP traffic beyond the SP boundary interface to the end destinations when the SP is responsible for media flow guarantees (see [D1.1], section 4.3.2.2). The transmission of VoIP traffic across the inter-SP link is the responsibility of the intermediate INPs (if such exist), and should be verifiable in the context of the established CPAs (see bellow).*
- (R3) In addition to these requirements, VoIP monitoring architecture should also allow VoIP Service Providers to verify the fulfilment of provisioning agreements they have subscribed to with IP Network Providers (i.e. CPA agreements). *As indicated in AGAVE WP1 (refer to Figure 24 of [D1.1]), CPA agreements might be local, i.e. through a direct connection to the IP infrastructure of the INP or remote, i.e. without direct connection to the infrastructure of the INP. The VoIP monitoring architecture should also manage both local and remote CPA associations.*

5.2.2VoIP monitoring implementations

Depending on the interface where the monitoring point is located, different sets of metrics might be monitored. The signalling specific metrics (call success rate, post dialling delay, and premature call release) will be monitored at the SLI and SII interfaces. The call quality metrics (end to end delay and transmission quality) will be monitored at SLI, SII and CPI interfaces.

Passive monitoring will only make use of passive monitoring points at SLI, SII and CPI interfaces, so that only real traffic generated by VoIP end users will be monitored. Several issues might be encountered:

- Many end user VoIP flows have to be monitored at the same time. The complexity of the monitoring function will increase with the number of flows to monitor.
- End user VoIP flows might cross several VoIP SP domains, and it might not be possible to get from the media flow the different SP domains it has crossed before a given domain boundary, so that if some VoIP flows are monitored as degraded at one boundary, it is difficult to identify if this media flow has been degraded between two given Service Providers. A correlation between different media flows using the route extracted from signalling information could be used for that purpose, but this need synchronization between media and signalling monitoring.

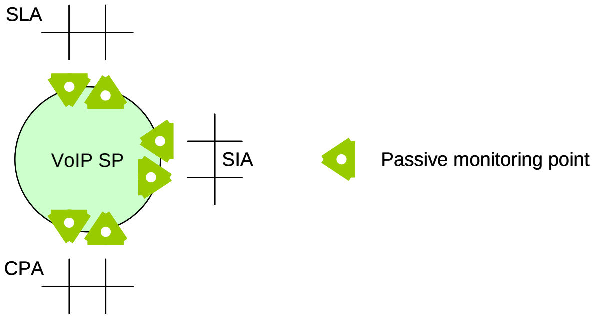


Figure 45 Passive VoIP monitoring

In order to encompass passive monitoring issues, an active monitoring could be used if VoIP agree to use compatible active monitoring points at SII or CPI interfaces sending and receiving VoIP test traffic. The benefit is to be able to monitor the IP segment between the two VoIP service providers independently form the end to end real VoIP flow paths. Several issues might also be encountered:

- Active monitoring generates additional VoIP traffic that does not generate revenues, so that it should only be used when the signalling and media resource availability is sufficient.
- Active monitoring points have to be configured so as to distinguish VoIP test traffic from real VoIP traffic.
- Active monitoring monitors the quality of a VoIP test flow assimilated to the real VoIP flows, so that it is as representative as possible, using the same packet length, type, sending rate and periodicity, and especially going the same IP path with the real VoIP flows.
- In order to evaluate the amount of real VoIP flows that might be impacted by a degradation encountered by VoIP test traffic, the number of VoIP active calls between the two VoIP Service Providers has to be monitored at the same time.

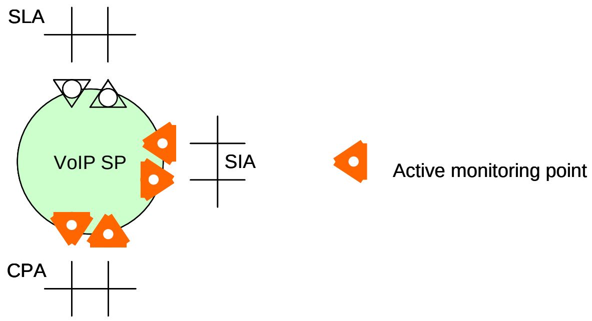


Figure 46 Active VoIP monitoring

5.2.3 Other considerations

As IP paths across different INP domains might not be the same for the outward and for the return from one VoIP SP to another VoIP SP, monitoring of the round trip time metric might not be sufficient to statute which party is involved in its degradation. The measurement of a one way delay metric by VoIP Service Providers could help to answer this but needs a proper synchronization between them.

The correlation between media flow degradation and premature call release cannot be done without synchronization between media flow and signalling monitoring. The granularity of media flow monitoring (per media flow monitoring or per set of media flows between a two VoIP Service Providers) should require further investigation in order to solve this issue.

6 SUMMARY

This deliverable presents the algorithms and mechanisms for implementing Network Planes within individual IP Network Providers (INPs), and also for binding the Network Planes across multiple domains for end-to-end service differentiation purposes. The proposed routing mechanisms include MRDV, multi-topology routing, overlay routing, IP tunnelling and q-BGP. Resilience requirements are also addressed for both services that need high QoS availability and robust traffic engineering in NP-aware domains. Specifically, description on MPLS fast rerouting and BGP based egress point selection algorithms are presented in this document in case of network failures. Finally, this deliverable describes the service provisioning paradigms at the application level, mainly on service level monitoring for Quality of Service (QoS) assurance to end users.

The final description on the proposed algorithms and mechanisms in WP3 will be provided in D3.2 towards the end of the project. The validation and evaluation of these work items will be performed in Work Package 4 (WP4).

7 REFERENCES

[AGAR05] Agarwal, S. et al, *Measuring the Shared Fate of IGP Engineering and Inter-domain Traffic*, in Proc. of IEEE ICNP, Boston, November 2005

[AHMA06] Ahmad, Z., Decraene B. , Le Roux JL, *High Availability in MPLS Networks*, in proceedings of the MPLS 2006 Conference, October 2006.

[AKEL04] Akella, A. et al, *A Comparison of Overlay Routing and Multihoming Route Control*, Proc. ACM SIGCOMM 2004

[ALI06] Ali, Z., Vasseur, J.-P., *Graceful Shutdown in GMPLS Traffic Engineering Networks*, draft-ietf-ccamp-mpls-graceful-shutdown-01.txt, October 2006

[ANDE01] Anderson, D. et al, *Resilient Overlay Networks*, Proc. 18th ACM SOSP, Banff, Canada, October 2001

[ANDE02] Andersen, D. et al. *Resilient Overlay Networks*. ACM SIGCOMM Computer Communication Review, Volume 32, Numero 1, January 2002.

[BERG06] Berger, Bryskin, Papadimitriou, Farrel, *GMPLS Segment Recovery*, http://www.ietf.org/internet-drafts/draft-ietf-ccamp-gmpls-segment-recovery-02.txt

[BHAT06] Bhatia, M., Joel, M., and Jakma, P., *Advertising Multiple NextHop Routes in BGP*, draft-bhatia-bgp-multiple-next-hops-01.txt, August 2006

[BLUN06] Blunk, L. Karir, M. and Labovitz, C., Internet draft, draft-ietf-grow-mrt-03, work in progress, June 26, 2006.

[BONA05] Bonaventure, O. et al, *Achieving Sub-50 Milliseconds Recovery Upon BGP Peering Link Failures*, in Proc. of ACM CONEXT, France, October 2005.

[BOOT04] Booth, D. et al, *Web Services Architecture*. W3C, February 2004.

[BOUC05] Boucadair, M. *QoS-Enhanced Border Gateway Protocol*, Internet-Draft, draft-boucadair-qos-bgp-spec-01.txt, July 2005

[BRES03] Bressound, B. et al, *Optimal Configuration for BGP Route Selection*, in Proc. of IEEE INFOCOM, April 2003

[CALL05] Callejo-Rodríguez, et al: *A Decentralized Traffic Management Approach for Ambient Networks Environments*, 16th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM) 2005, 145-156. Springer.

[CHOI05] Choi, B. Y. and Bhattacharyya. S., *Observations on CISCO sampled NetFlow*. In Proceedings of ACM SIGMETRICS Workshop on Large-Scale Network inference (LSNI), June 2005.

[CISC04] CISCO Systems. *CISCO IOS Service Assurance Agent*. 2004. http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/saang_ds.pdf

[COHO78] Cohon. J.L. Multi-objective Programming and Planning. Academic Press, New York

1978

[CRIS03] G. Cristallo and C. Jacquenet. Providing *Quality of Service Indication by the BGP-4 Protocol: the QOS_NLRI attribute*. Internet draft, work in progress. Draft-jacquenet-qos-nlri-05, June 2003.

[D1.1] Boucadair, M. et al., *Parallel Internets Framework*, AGAVE Deliverable D1.1, 8 September 2006.

[D2.1] Mykoniati, E. et al., *Initial Specification of the Connectivity Service Provisioning Interface Components and Implementation Plan*, AGAVE Deliverable D2.1, 2006

[DABE04] Dabek, F. et al, *A decentralized network coordinate system*. In Proceedings of ACM SIGCOMM, August 2004.

[DAVI00] Davie, B., and Rekhter, Y., *MPLS: technology and applications*. Morgan Kaufmann, 2000.

[DEB01] Deb, K., *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, June 2001.

[DORA99] Doraswamy, N. and Harkins, D., *IPSec; The New Security Standard for the Internet, Intranets, and Virtual Private Networks*. Prentice Hall, Internet Infrastructure Series, 1999.

[DOWN99] Downey, A. B., *Using pathchar to estimate Internet link characteristics*. In Proceedings of ACM SIGCOMM, October 1999.

[DUBO01] Dubois, N., Decraene B., Fondeviole B., *Graceful Shutdown of BGP Sessions*, Internet Draft draft-dubois-bgp-planned-maintenance-00.txt, June 2004.

[DUBO02] Dubois, N., et al, *Requirements for planned maintenance of BGP sessions* Internet Draft draft-dubois-bgp-pm-reqs-02.txt, July 2005.

[DUBO03] Dubois, N., et al, *Requirements for planned maintenance of BGP sessions*, in proceeding of IETF 63 IDR WG August 2005.

[DUBO04] Dubois, N., et al, *Graceful Shutdown of BGP Sessions*, in proceeding of IETF 60 IDR WG August 2004.

[ENNS06] Enns, R., *NETCONF Configuration Protocol*. Internet draft, work in progress. Draft-ietf-netconf-prot-12, February 2006,

[FEAM03] Feamster, N. et al, *Guidelines for Inter-domain Traffic Engineering*, in Proc. of ACMSIGCOMM CCR, October 2003

[FEAM04] Feamster, N. et al, *The Case for Separating Routing from Routers*. In Proceedings of the ACM SIGCOMM FDNA Workshop, August 2004.

[FORT02] Fortz, B. et al, *Optimizing OSPF/IS-IS Weights in a Changing World*, in Proc. of IEEE JSAC, May 2002

[FRAN01] Frankel, S., *Demystifying the IPsec Puzzle*. Artech House Publishers,Computer Security Series, 2001.

[FRAN05] François, P., Bonaventure, O., *Avoiding transient loops during IGP convergence in IP networks* in proceeding of IEEE INFOCOM 2005, March 2005, Miami, Fl., USA.

[FRAN06] François, P. et al, *Loop-free convergence using oFIB,* Internet Draft draft-francois-ordered-fib-02 October 2006.

[GAO06] Gao, R., Dovrolis, C. and Zegura E., *Avoiding Oscillations due to Intelligent Route Control Systems*. In Proceedings of IEEE INFOCOM, April 2006.

[GEOR01] Georgatos, F. et al, *Providing Active Measurements as a Regular Service for ISP's*. In Proceedings of PAM. April 2001.

[GLOV97] Glover, F. et al, *Tabu Search.* Kluwer Academic Publisher, Norwell MA 1997

[GRAH98] Graham, I. D., et al, *Nonintrusive and Accurate Measurement of Unidirectional Delay and Delay Variation*. In Proceedings of INET, July 1998.

[GRIF07] D. Griffin et al., *Interdomain Routing through QoS-class Planes*, to appear in IEEE Communications Magazine special issue on Quality of Service based Routing Algorithms for Heterogeneous Networks Feb 2007.

[GUDG03] Gudgin, M., et al, *Simple Object Access Protocol (SOAP) v1.2 Part 1: Messaging Framework.* W3C Recommendation, June 2003.

[HAN05] Han, J. et al, *Topology Aware Overlay Networks*, Proc. IEEE INFOCOM 2005

[HO04] Ho, K. et al, *Multi-objective Egress Router Selection Policies for Inter-domain Traffic with Bandwidth Guarantees*, in Proc. of IFIP Networking, Greece, May 2004

[HO06] Ho, K. et al, *Joint Optimization of Intra- and Inter-Autonomous System Traffic Engineering*, in Proc. of IEEE NOMS, Canada, April 2006

[HUFF02] Huffaker, B., et al, , *Delay Metrics in the Internet*. In Proceedings of IEEE international Telecommunications Symposium (ITS), September 2002.

[HUST06] Huston, G., *The CIDR Report*. January 2006. http://www.cidr-report.org.

[IPSP06] IPsphere Forum Work Program Committee. *Creating a Commercialy Sustainable Framework for IP Services*. White Paper v1b.0, May 2006. http://www.ipsphereforum.org.

[JAIN02] Jain, M. and Dovrolis. C., *End-to-End Available Bandwidth: Measurement Methodology, Dynamics and Relation with TCP Throughput*. In Proceedings of ACM SIGCOMM, August 2002.

[JAIN02b] Jain, M. and Dovrolis. C., *Pathload: A Measurement tool for end-to-end available bandwidth*. In Proceedings of PAM, 2002.

[KALI00] Kalindi, S., Zekauskas M., and Uijterwaal, H., *Comparing two implementations of the IETF IPPM One-way Delay and Loss Metric*. PAM Workshop, April 2000.

[KATZ06] Katz, D. and Ward, D., *Bidirectional Forwarding Detection (BFD)*. Internet draft, work in progress. Draft-ietf-bfd-base-05, June 2006.

[LAI01] Lai, K. and Baker, M., *Nettimer: A Tool for Measuring Bottleneck Link Bandwidth*. In Proceedings of USENIX USITS, March 2001.

[LAUN05] C. de Launois, B. Quoitin and O. Bonaventure. *Leveraging network performance with IPv6 multihoming and multiple provider-dependent aggregatable prefixes*. Computer Networks, Volume 50, Numero 8, June 2006.

[LAUN05b] C. de Launois, S. Uhlig and O. Bonaventure. *Scalable Route Selection for IPv6 Multihomed Sites*. In Proceedings of IFIP Networking, LNCS3462, May 2005.

[LI04] Li, Z. et al, *QRON: QoS-aware Routing in Overlay Networks*, IEEE Journal on Specific Areas in Communication (JSAC), Vol. 22, No. 1. 2004

[MSCLD12] Howarth, M. et al. MESCAL Deliverable D1.2 *Initial Specification of Protocols and Algorithms for Inter-domain SLS Management and Traffic Engineering for QoS-based IP Service Delivery and their Test Requirements.*

[MSCLD13] Wang, N. et al. MESCAL Deliverable D1.3 *Final specification of protocols and algorithms for inter-domain SLS management and traffic engineering for QoS-based IP service delivery.*

[NUCC03] Nucci, A. et al, *IGP Link Weight Assignment for Transit Link Failures*, in Proc. of ITC, Berlin, September 2003

[PRZY05] Przygienda, T. et al, “M-ISIS: Multi Topology (MT) Routing in IS-IS,” IETF Internet-draft, draft-ietf-isis-wg-multi-topology-11.txt, October 2005.

[PSEN06] Psenak, P. et al, “Multi-Topology (MT) Routing in OSPF,” IETF Internet-draft, draft-ietf-ospf-mt-06.txt, February 2006.

[QUOI05] Quoitin, B. and Bonaventure, O., *A Cooperative Approach to Inter-domain Traffic Engineering*. In Proceedings of EuroNGI, April 2005.

[QUOI06] Quoitin, B., *BGP-based Inter-domain Traffic Engineering*. PhD Thesis, August 2006.

[RFC1035] Mockapetris, P, *Domain names – implementation and specification*. RFC1035, November 1987.

[RFC1363] Partridge, C., *A Proposed Flow Specification*. RFC1363, September 1992.

[RFC1771] Rekhter, Y. and Li. T., A *Border Gateway Protocol 4 (BGP-4)*. RFC1771, March 1995.

[RFC1825] Kent, S. and Atkinson, R., *Security Architecture for the Internet Protocol*. RFC1825, November 1998.

[RFC1853] Simpson, W., *IP in IP Tunneling*. RFC1853, October 1995.

[RFC2136] Vixie, P. et al, *Dynamic Updates in the Domain Name System (DNS UPDATE)*. RFC2136, April 1997.

[RFC2475] Blake, S. et al, *An Architecture for Differentiated Services*. RFC2475, December 1998.

[RFC2679] Almes, G. et al, *A One-way Delay Metric for IPPM*. RFC2679, September 1999.

[RFC2702] Awduche, Malcom, Agogbua, 0'Dell, MacManus "Requirements for Traffic Engineering over MPLS", RFC 2702

[RFC2748] Boyle, J. et al, *The COPS (Common Open policy Service) Protocol*. RFC2748, January 2000.

[RFC2782] Gulbrandsen, A. et al, *A DNS RR for specifying the location of services (DNS SRV)*. RFC2782, February 2000.

[RFC2784] Farinacci, D. et al, *Generic Routing Encapsulation (GRE)*. RC2784, March 2000.

[RFC2858] Bates, T. et al, *Multiprotocol Extensions for BGP-4*. RFC2858, June 2000.

[RFC3209] Awduche, D., et al, *RSVP-TE: Extensions to RSVP for LSP Tunnels*. RFC3209, December 2001.

[RFC3280] Housley, R. et al, *Internet X.509 Public Key Infrastructure Certificate and Certificate Recovation List (CRL) Profile*. RFC3280, April 2002.

[RFC3318] Sahita, R. et al, *Framework Policy Information Base*. RFC3318, March 2003.

[RFC3779] Lynn, C. et al, *X.509 Extensions for IP Addresses and AS Identifiers*,RFC3779, June 2004.

[RFC3931] Lau, J. et al, *Layer Two Tunneling Protocol – Version 3 (L2TPv3)*. RFC3931, March 2005.

[RFC4033] Arends, R. et al, *DNS Security Introduction and Requirements*. RFC4033, March 2005.

[RFC4034] Arends, R., et al, *Resource Records for the DNS Security Extensions*. RFC4034, March 2005.

[RFC4035] Arends, R. et al, *Protocol Modifications for the DNS Security Extensions*. RFC4035, March 2005.

[RFC4090] Pan, Swallow, Atlas, *Fast Reroute extensions to RSVP-TE for LSP LSPs*, RFC4090, May 2005

[RFC4364] Rosen, E. and Rekhter, Y., *BGP/MPLS IP Virtual Private Networks (VPNs)*. RFC4364, February 2006.

[RFC4398] Josefsson, S. Storing *Certificates in the Domain Name System (DNS)*. RFC4398, March 2006.

[RFC4420] Farrel et al. *Encoding of attribute for RSVP-TE LSPs*, RFC4420, February 2006

[RFC4655] Farrel, A., Vasseur, J.-P. and Ash J., A *Path Computation Element (PCE)-Based Architecture*. RFC4655, August 2006.

[RFC4656] Shalunov, S. et al, *One-way Active Measurement Protocol (OWAMP)*. RFC4656, September 2006.

[RFC792] Postel, J., *Internet Control Message Protocol (ICMP)*. RFC792, September 1981.

[ROUX04] Le Roux, *Evaluation of MPLS Fast Reroute Architectures to protect an MPLS VPN network*, proceedings of the MPLS World Congress 2004.

[ROUX06] Le Roux et al. *Protection MPLS FRR pour la télémédecine*, French RNRT VTHD project, http://www.get-telecom.fr/archive/156/MPLS_Fiche_OK.pdf

[SARO02] Saroiu, S., Gummadi, P. K. and Gribble S. D., *SProbe: A Fast Technique for Measuring Bottleneck Bandwidth in Uncooperative Environments*. Submitted for publication, 2002, http://sprobe.cs.washington.edu/sprobe.ps.

[SCUD05] Scudder, J., BGP Monitoring Protocol. Internet draft, draft-scudder-bmp-00, work in progress, August 2005.

[SOMM02] Sommer, R. and Feldmann, A., *Netflow: Information loss or win ?* In Proceedings of ACM Internet Measurement Workshop, November 2002.

[SRID05] Sridharan, A. et al, *Making IGP Routing Robust to Link Failures*, in Proc. of IFIP Networking, Canada, May 2005

[STRA03] Strassner, J., *Policy-Based Network Management: Solutions for the Next Generation*. Morgan Kaufmann Publishers, August 2003.

[SUBR05] Subramanian, L. et al, *HLP: A Next Generation Inter-Domain Routing Protocol*. In Proceedings of ACM SIGCOMM, August 2005.

[TELK02] Telkamp, T. *Traffic Characteristics and Network Planning*, NANOG 26 Meeting, October 2002.

[UHLI03] Uhlig, S., Bonaventure, O. and Quoitin, B., *Inter-domain Traffic Engineering with Minimal BGP Configurations*. In Proceedings of the 18th International Teletraffic Congress (ITC), September 2003.

[VARG04] Varghese, G. and Estan, C., *The measurement manifesto*. ACM SIGCOMM Computer Communications Review, Volume 34, Numero 1, 2004.

[VASS01] Vasseur, JP., Ikejiri, Y. and Zhang R., *Reoptimization of Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Loosely Routed Label Switched Path (LSP),* RFC 4736, November 2006.

[VASS02] Vasseur, JP., and Previdi, S., *Definition of an IS-IS Link Attribute sub-TLV,* Internet Draft draft-ietf-isis-link-attr-02.txt, October 2006.

[VEGE01] Vegesna, S., *IP Quality of Service*. Cisco Press, January 2001.