AGAVE

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

IST-027609

D3.2: Specification of Mechanisms, Algorithms and Protocols for Engineering the Parallel Internets

Document Identifier: AGAVE/WP3/UniS/D3.2/final	
Deliverable Type: Report **Deliverable Nature:** Public	**Contractual Date:** M27 **Actual Date:** 15 April 2008

Editor:	Ning Wang (UniS)
Authors:	*TID:* A. J. Elizondo Armengol, O. González de Dios, G. García de Blas, P. Montes Moreno *FTR&D:* M. Boucadair, B. Decraene, B. Lemoine, J.L. Le Roux *Algo:* P. Georgatsos, E. Mykoniati *UCL.uk:* D. Griffin, J. Spencer, S. S. Lor *UniS:* N. Wang, K-H. Ko, M. Amin, M. Howarth, G. Pavlou *UCL.be:* L. Iannone, D. Saucez, O. Bonaventure
Abstract:	This deliverable provides the final specifications of the algorithms and mechanisms proposed in the AGAVE project for implementing Network Planes (NPs) and Parallel Internets (PIs). Intra-domain mechanisms for enabling edge-to-edge service differentiation include multi-topology routing, INP-layer overlay routing and Multi-path Routing with Dynamic Variance (MRDV). IP tunnelling and QoS-enhanced BGP (q-BGP) are designed and enhanced respectively for implementing end-to-end PIs. In addition, ASBR fast rerouting with RSVP-TE extensions, BGP planned maintenance and BGP/IGP based traffic engineering with resilience awareness are proposed for achieving high service assurance as well traffic optimisation purposes in case of network failures. Finally service level and network level monitoring mechanisms are introduced at the end of this document.
Keywords:	Quality of Service (QoS), Traffic engineering, Network Planes, Parallel Internets, VoIP, MRDV, Multi-topology routing, Overlay routing, IP tunnelling, q-BGP, BGP planned maintenance, 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 the final result of AC3.1, AC3.2 and AC3.3 activities developed under 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 testbeds;
- 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 provides the final specifications of the algorithms and mechanisms for implementing Network Planes (NPs) within individual IP Network Providers' (INPs) domains, and also for binding NPs across multiple domains to form Parallel Internets (PIs) for end-to-end service differentiation purposes.

As mentioned in [D3.1], routing is the major dimension to be explored in the AGAVE project for realising NPs and PIs. First of all, a general overview on the proposed algorithms and mechanisms is presented in section 2 accompanied with classifications according to the service and operational requirements. Distinct novelties and lightweightness features associated with these techniques are also summaries in this section. As far as NP realisation is concerned, we introduce three lightweight intra-domain routing mechanisms, namely Multi-topology routing, INP-layer overlay routing and Multi-Paths Routing with Dynamic Variance (MRDV). The aim is to enable edge-to-edge service differentiation within individual autonomous domains with scalable and incremental mechanisms. 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 the specification of IP tunnelling and enhanced QoS-Enhanced BGP (q-BGP) are provided. In the AGAVE project, we also consider resilience requirements for both service assurance and operational performance in case of network failures. Towards this end, ASBR protection with RSVP-TE extensions, BGP planned maintenance and BGP/IGP based traffic engineering with resilience awareness are designed and implemented with full specifications presented in this document. Finally AGAVE monitoring issues are also considered, including both level monitoring mechanisms for VoIP applications and INP level Network Plane monitoring.

The validation and evaluation on the proposed algorithms/mechanisms will be done in WP4 and the relevant results will be documented in D4.2.

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 designed for implementing Network Planes, such as Multi-Topology Routing (MTR), INP-layer intra-domain 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]) have been 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 have been 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 have been selected and customised. Suitable open-source software and proprietary software owned by partners have been reviewed and selected where appropriate. The components required to realise the Parallel Internets have been designed and implemented. Enhancements to brought-in software and integration into the simulators have been also undertaken.

1.2 Structure of this document

This document is structured as follows:

- Section 2, *Design overview* gives a top-level description and classification on the design of Network Planes and Parallel Internets proposed in the AGAVE project. Lightweightness and incremental deployment issues associated with each scheme are also discussed in this section.
- Section 3, *Network Plane engineering* provides the final specifications of 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.
- Section 4, *Network Plane binding* provides the final specifications of 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, specifically on BGP planned maintenance and ASBR protection with RSVP-TE extensions.
- Section 5, *AGAVE Monitoring considerations* describes both monitoring issues at the SP level for VoIP service assurance, and those at the network layer responsible by INPs.
- Section 6, *Summary* provides a brief summary of this document.

1.3 Major updates

This section provides a summary of the major changes as compared to [D3.1].

- Enhanced description and classification of individual mechanisms designed in the AGAVE project are included in section 2. Brief descriptions on the novelties and lightweight features associated with these mechanisms are also provided.
- Multi-topology routing: Significant enhancement has been made on the offline network provisioning through link weight optimisation in order to provide maximum intra-domain path diversity. A brand new algorithm is specified in this document on adaptive traffic splitting across individual routing topologies with the aim for dynamic load balancing. The structure of the traffic engineering information base (TIB) is also enhanced based on the efficiency requirement of splitting ratio computation.
- Overlay routing: In this document we clearly indicate that the main objective is to enable fast reroute with traffic engineering awareness. A new algorithm for overlay tunnel endpoint selection is specified in this document. In addition, the intra- and inter-domain overlay routing considerations are put separately into the NP engineering and NP binding sections.
- MRDV: The description of the extension of MRDV (Multipath Routing with Dynamic Variance) to support multiple traffic classes in this deliverable differs from the previous description in [D3.1] mainly in the focus of the specification. The specification in this document focuses on detailing the mechanisms that have been finally implemented after valuating the performance of different alternatives by simulation means. For instance, simulations showed that the avoidance of secondary loops provided just slightly better results than the avoidance of primary loops. Therefore, it was decided not to include the avoidance of secondary loops in the final specification.
- IP tunnelling: In the framework of the AGAVE Project, the IP Tunnelling approach targets at making available excess resources, present but not exploited in the current Internet, thus making possible enhancing performance, supporting QoS and Parallel Internets, as described in [D3.1] As complement of the [D3.1], this document present the design and implementation of the IP Tunnelling approach, which has been split in two main components: the Tunnel Service (TS) and the Tunnel Service Controller (TSC). The TS is the component that performs encapsulation and decapsulation of IP packets. It is placed on border routers of domains and can, by selecting tunnel endpoints, send packets using different paths. The choice of the tunnel endpoints is based on feedback of the TSC, which can be deployed anywhere inside the domain.
- q-BGP enhancement: q-BGP specifications were enhanced by defining a new set of derived QoS-attribute types to be conveyed in q-BGP UPDATE messages and the means of calculating both primitive and derived attributes through dynamically measured values. Priority-based route selection policies were enhanced by the use of an equivalency margin to increase load balancing between routes selected based on the highest priority attribute.
- BGP planned maintenance: A full specification of the proposed technique for enabling fast reroute during BGP planned maintenance period is provided in this document.
- Resilience-aware BGP/IGP traffic engineering: This is formally titled intra-/inter-domain interactions in terms of resilience in [D3.1]. The problem formulation remains the same and in this document a complete algorithm specification is included.
- The AGAVE monitoring considerations, including both service layer monitoring for VoIP applications and INP layer network monitoring are specified in a dedicated section (section 5).

2 DESIGN OVERVIEW

2.1 Classification

As mentioned in [D1.1] and [D3.1], the AGAVE Project proposes a multi-dimensional paradigm for implementing and engineering Network Planes (NPs) and Parallel Internets (PIs), with routing differentiation being the main focus. Figure 1 shows the routing techniques that are considered in the AGAVE project for the implementation of Network Planes. There are three distinct requirements that need to be concerned when NPs/PIs are realised and engineered. First of all, the realisation of NPs and PIs need to fulfil the defined *service requirements* such as the contracted QoS metrics and their assurances through CPAs. These metrics may include delay, packet loss ratio, throughput as well as their availability. Secondly, *operational requirements* need to be considered from the INP's point of view, with the main objective to achieve manageability and lightweightness (e.g. to minimise the management/control complexity added to the network), as well as optimised network resource usage. Finally, *resilience requirements* concern the impact from network failures on the availability of provisioned services and operational efficiency.

The NP/PI realisation/implementation techniques considered in the AGAVE Project are positioned as follows. Multipath Routing with Dynamic Variance (MRDV), DiffServ/MPLS, Multi-topology routing (MTR), Overlay routing, q-BGP and IP tunnelling consider both service requirements and operational requirements. Among these techniques, MRDV, DiffServ/MPLS and MTR are the mechanisms used for realising NPs within a single autonomous domain. q-BGP and IP tunnelling are used for realising PIs across multiple domains. Overlay routing is a technique to be used for both cases. As far as the resilience requirements are concerned, BGP Planned Maintenance (PM) and ASBR Fast ReRoute (FRR) tackles the issue of minimising the loss-of-connectivity duration caused by the scheduled and unexpected breakdown of inter-domain links respectively. Finally Resilience-aware BGP TE is proposed for improving operational efficiency by taking into account both intra- and inter-domain link failures; more specifically, to achieve overall balanced load distribution given any single link failure scenario. Top-level descriptions on these techniques are provided in section 2.2.

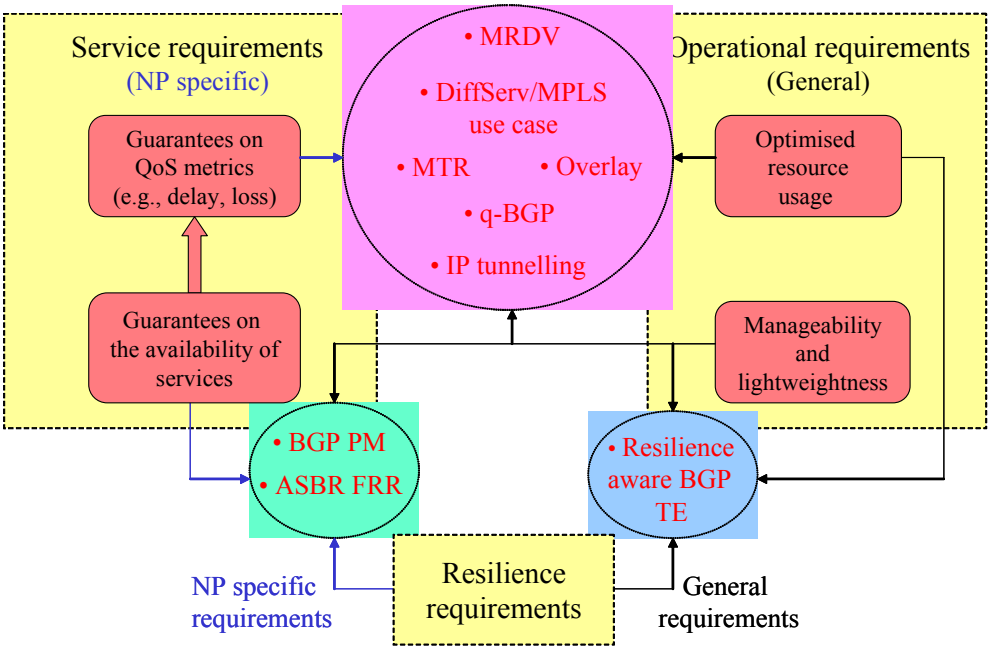


Figure 1 Positioning AGAVE techniques for NP/PI realisation

2.2 Description of individual techniques for NP/PI engineering

Multi-topology routing (MTR) is a mechanism used for implementing Network Planes within a single autonomous domain. The service objective is to provide intra-domain edge-to-edge QoS differentiation (such as delay) across NPs and enhancing QoS assurance against unexpected network/traffic dynamics. In addition, the operational objective is to perform adaptive traffic engineering (TE, e.g. load balancing) in order to maximise the efficiency of the IP network. Existing routing protocols such as Multi-topology OSPF [RFC4915] and Multi-topology IS-IS [RFC5120] can be directed used as the underlying IGP routing platform for the above purposes. The novelty is that no massive label switched paths (LSPs) are needed for providing QoS and TE but the relevant TE performance can be still close to optimality. In order to deploy the proposed MTR-based NP engineering paradigm, a centralised server called TE manager is needed within each domain.

Overlay routing is a mechanism used for implementing Network Planes within a specific domain. The service objective is to enable fast reroute (FRR) in order to minimise the duration of loss-of-connectivity duration in case of network failures. The operational objective is to perform traffic engineering through intelligent selection of overlay tunnelling endpoints in order to avoid overwhelming backup IGP paths after failure happens. Specific data-plane mechanisms that can be used for supporting this paradigm include most of the tunnelling techniques such as IP-in-IP [RFC1853] and GRE [RFC2784] etc. The novelty is to achieve both fast reroute for minimising QoS disruption to end users and optimised network resource optimisation for post-failure network load balancing.

The extension of MRDV to support multiple traffic classes (MRDV with CoS support) is a lightweight way to provide differentiated routing NP engineering on intra-domain scenarios. Each intra-domain router runs just one instance of a link-state routing protocol, which is used to build an NP-independent routing table. From measurements on link load per traffic class, which are obtained for each interface, each router rebuilds periodically NP-specific routing tables from the NP-independent routing table, providing this way network planes differentiated by routing. The mapping from the general NP-independent routing table to NP-specific ones is done by using the MRDV algorithm, which is based on the concept that suboptimal paths to route traffic towards a destination are used when optimal paths are close to be congested. The decision to use suboptimal paths is taken from the calculation of a variance parameter, which depends on the link load in the router interface for the optimal path: the higher the load, the higher the variance parameter, and the higher the variance, the higher the chances to use suboptimal paths. A different variance parameter is calculated for each traffic class, considering for the variance calculation the load generated by that traffic class and that offered by all the higher priority traffic classes. This way the variance is NP-dependent, and it is achieved NP differentiation based on routing depending on the load. Obviously, MRDV with CoS support is addressing NP differentiated routing. Besides, given that MRDV is ideal to postpone network congestion by using suboptimal paths when the optimal ones are highly loaded, MRDV with CoS support becomes a perfect way to provide NPs based on resilience. Finally, MRDV could be used, jointly with admission control mechanisms, traffic policing and shaping mechanisms at the entrance of the domain and network provisioning and dimensioning rules, to provide QoS differentiated NPs.

The tunneling support proposed in the AGAVE Project is composed of two main parts. The Tunnel Service (TS), being the protocol that actually performs tunneling by encapsulating and decapsulating IP packets. The Tunnel Service Controller (TSC), being the active service that allows classifying and managing QoS parameters of the different tunnels representing different NPs. The TS is based on the LISP proposal and is able to provide full control on the inbound traffic and does not need changes at network layer, thus being rapidly deployable. The Tunnel Service Controller (TSC) is based on the IDIPS protocol, and is able to sort the multiple paths available between a source and a destination. The classification can be done on various criteria, based on active measurements (e.g., ping) or passive measurements (e.g., BGP feeds, Netflow). TSC runs at application layer and together with its counter part, the TS, it provides a lightweight mechanism for better-than-best-effort service for inter-domain NPs, capable to inter-operate with MRDV and MTR.

q-BGP is one of the methods to maintain and distribute QoS-based routing information in Parallel Internets in AGAVE. q-BGP builds incrementally on standard BGP4 by defining two new optional attributes: a QoS Service Capability attribute, which signals a QoS aware q-BGP session and to which Parallel Internet it belongs; and a QoS_NLRI attribute which contains optional fields in UPDATE messages which describe the QoS attributes of the path expressed in the message. The work in AGAVE has been to specify how q-BGP can be used to interconnect network planes within INPs to form inter-domain Parallel Internets with different QoS characteristics. The novelty is in determining how QoS attributes should be calculated and determined within INPs, how they are advertised to adjacent INPs and then used in route selection policies so as to control local routing decisions that influence the end-to-end performance of traffic allocated to that Network Plane. An important aspect of the work considered how dynamically monitored/calculated QoS attributes can be used to reflect actual network conditions without causing repeated avalanches of q-BGP messages throughout the network and an inability to converge on a stable routing configuration.

Resilience-aware BGP traffic engineering is *not* a standalone mechanism to be used for implementing Parallel Internets. Instead, it is a resource optimisation technique for Network Plane binding with BGP/IGP as the underlying routing protocols. This paradigm can be applied on each specific "stratum" of the Parallel Internets independently. The operational objective is to achieve balanced load distribution across both intra- and inter-domain network links even in case of network failures. It mainly considers offline BGP egress point selection by taking into account the hot-potato-routing effect with the aim to minimise the maximum link utilisation given any single link failure scenario. This technique can be used to improve the overall network performance when engineering Parallel Internets with plain IP based routing protocols such as BGP and IGP as well as their extensions.

3 NETWORK PLANE ENGINEERING

3.1 Multi-topology routing

3.1.1 Overview

Currently, intra-domain multi-topology IP routing protocols include Multi-Topology OSPF (MT-OSPF) [RFC4915] and Multi-Topology IS-IS [RFC5120]. 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. We will discuss how this can be implemented in section 3.1.3 by using edge-to-edge delay differentiation as an example.

3.1.2 MTR within a single Network Plane

In this section we describe how MTR is used within a single NP for internal traffic engineering (TE) purposes (N-to-one MTR mapping). The main objective is to adaptively perform traffic control against unexpected traffic dynamics such as upsurges within the plain. This scheme can be easily extended to the scenario of Hybrid MTR mapping, e.g. across multiple NPs, each with different delay requirements.

Figure 2 provides an overall description on the proposed scheme. As it can be observed, Offline Link Weight Optimisation (*OLWO*) and Adaptive Traffic Control (*ATC*) are the two major components included in the NP Provisioning and Maintenance block. First of all, the NP Design and Creation block decides the appropriate number of MTR topologies for this Network Plane, which is fed into the *OLWO* block as a general guideline for NP Provisioning and Maintenance. *OLWO*, which optionally takes as input the forecast traffic matrix from NP Mapping, determines long-time routing configuration for each MTR topology so as to meet the QoS requirements of this NP and also the objectives of enabling path diversity across individual routing topologies (RTs).

The *OLWO* component produces as output multiple sets of MT-IGP link weights, each of which is used for a specific MTR topology. It should be noted that, although these sets of link weights enable path diversity between each source-destination (S-D) pair across individual MTR topologies, all of them should satisfy the common edge-to-edge QoS requirement specified by the single Network Plane. A salient novelty is that the optimization of the MT-IGP link weights does not rely on the availability of a traffic matrix *a priori*, which plagues existing offline IGP-based TE solutions due to inaccuracy of traffic matrix estimations. Instead, our offline link weight optimization is only based on the characteristics of the network itself, such as the physical topology.

Whilst *OLWO* focuses on static routing configuration in a long timescale, the *ATC* component provides the complementary functionality to enable dynamic control over the behaviour of traffic that

cannot be anticipated in advance. The objective is both assuring the provisioned QoS required by the Network Plane and also maintaining the desired TE performance. The inputs for the *ATC* component include the static MT-IGP link weights produced by *OLWO* and the monitored traffic dynamics at short time-scale. At each short-time interval, *ATC* computes new traffic splitting ratio across individual RTs for re-assigning traffic in an optimal way to the diverse IGP paths between each S-D pair. This functionality is handled by a centralized TE manager who has complete knowledge about the network topology and periodically gathers the up-to-date monitored traffic conditions of the operating network. These new splitting ratios are then instructed from the TE manager to individual source PoP nodes who respond by remarking the MT-IDs of their locally originated traffic accordingly.

Inside each MTR-aware router, multiple RIBs are maintained, each serving a specific MTR topology. The configuration of these RIBs is based on the pre-calculated MT-IGP link weights that are normally kept static. On the other hand, packet remarking at ingress routers, driven by the *ATC* component, enables dynamic traffic shifting among equivalent MTR topologies belonging to the same Network Plane. In summary, the forwarding decision on each incoming packet is influenced by both static RIBs as well as the pre-marked (by *OLWO*) or remarked (by *ATC*) traffic.

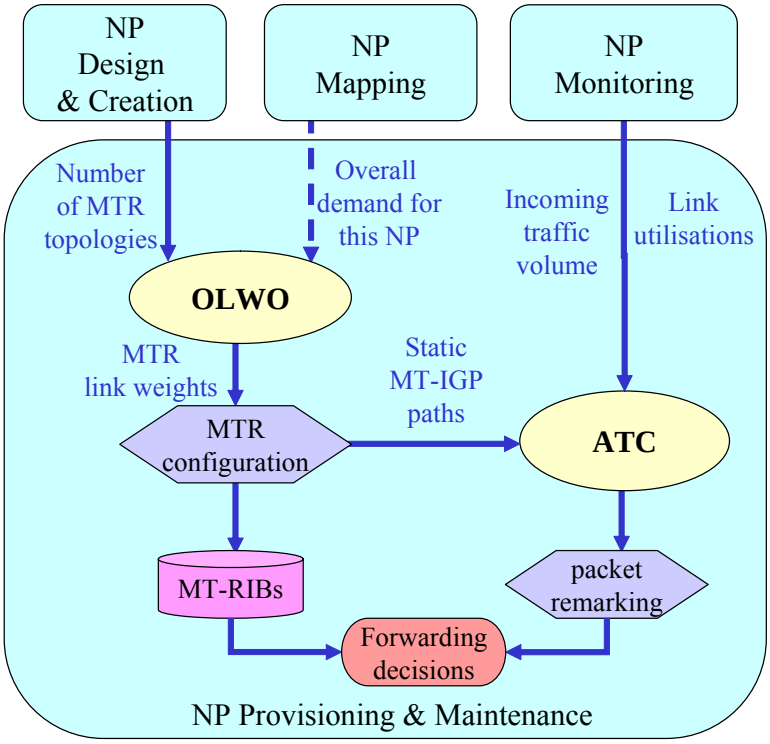


Figure 2 MTR within a single NP

3.1.2.1 Offline link weight optimisation

The network model for MT-IGP link weight optimization is described as follows. The network topology is represented as a directed graph $G=<V, E>$, where V and E denote the set of PoP nodes and inter-PoP links respectively. Each link $l \in E$ is associated with bandwidth capacity C_l. In an MT-IGP based paradigm with routing topology set R, each link is also assigned with $|R|$ distinct link weights (denoted by $w_l(r), r \in R$) where $|R|$ is the number of MT-IGP topologies to be configured. In MT-IGP based routing, an IGP path between each pair of nodes (u, v) in routing topology r, denoted by $P_{u,v}(r)$, is the shortest path according to the link weight configuration $W(r)$ for that routing topology.

Our definition of path diversity across multiple routing topologies is as follows. For each source-destination pair (*u, v*) we denote *Degree of Involvement* (*DoI*) for each link *l* as the number of routing topologies that include *l* in their shortest IGP paths between the node pair, formally:

$$DoI_l^{u,v} = \sum_{r \in R} x_l^{u,v}(r)$$

where $x_l^{u,v}(r)$ indicates whether link *l* constitutes the shortest IGP path between *u* and *v* in routing topology *r*:

$$x_l^{u,v}(r) = \begin{cases} 1 & if\ l \in P_{u,v}(r) \\ 0 & otherwise \end{cases}$$

Our ultimate objective is to minimize the chance that a single link is shared by *all* routing topologies between each source-destination pair. The objective is to avoid introducing critical links with potential congestion where the associated source-destination pairs cannot avoid using it no matter which routing topology is used. Towards this end, we define the *Full Degree of Involvement* (*FDoI*), which indicates whether a critical link *l* is included in the IGP paths between source-destination pair (*u, v*) in *all* routing topologies:

$$FDoI_l^{u,v} = \begin{cases} 1 & if\ DoI_l^{u,v} = |R| \\ 0 & Otherwise \end{cases}$$

In summary, MT-IGP link weight optimization problem is formally described as follows. To calculate $|R|$ sets of positive link weights $W(r) = \{w_l(r)\}: w_l(r) > 0, r \in R$ in order to minimize:

$$\sum_{u,v \in V} \sum_{l \in E} FDoI_l^{u,v}$$

According to this problem formulation, it can be easily inferred that, if $FDoI_l^{u,v} = 0$ for all $l \in P_{u,v}(r), r \in R$, then the source node *u* is always able to find at least one routing topology in which the IGP path to the destination node *v* can bypass the bottleneck link *l*, no matter which one becomes congested. Nevertheless, this does not necessarily mean that the IGP paths in this case are completely disjoined across multiple routing topologies. We designed and implemented a Genetic Algorithm (GA) based scheme to compute the MT-IGP link weights for the problem formulated above. The cost function (fitness) is designed as:

$$\frac{\lambda}{\sum_{u,v \in V} \sum_{l \in E} FDoI_l^{u,v}}$$

where λ is a constant value. In our algorithm we also put higher emphasis in avoiding *FDoI* for this type of low capacity links in comparison to high-capacity ones, i.e. to try to provide alternative IGP paths in other routing topologies that avoid using this link between the adjacent PoPs. In our GA based approach each chromosome *C* is represented by a link weight vector for $|R|$ routing topologies: $C = \{W(r) | r \in R\}$. The total number of chromosomes in each generation is set to 100. According to the basic principle of Genetic Algorithms, chromosomes with better fitness value have higher probability of being inherited in the next generation. To achieve this, we first rank all the chromosomes in descending order according to their fitness, i.e., the chromosomes with high fitness are placed on the top of the ranking list. Thereafter, we partition this list into two disjoined sets, with the top 50 chromosomes belonging to the upper class (*UC*) and the bottom 50 chromosomes to the lower class (*LC*). During the crossover procedure, we select one parent chromosome from *UC* and the other parent from *LC* in generation i for creating the child C^{i+1} in generation $i+1$. Specifically, we use a crossover probability threshold $K_C \in [0,0.5)$ to decide the genes of which parent to be inherited

into the child chromosome in the next generation. We also introduce a mutation probability threshold K_M to randomly replace some old genes with new ones. In our implementation K_C and K_M are set to 0.3 and 0.01 respectively. The optimized MT-IGP link weights are pre-configured in the network as the input for adaptive traffic engineering which will be detailed in the next section.

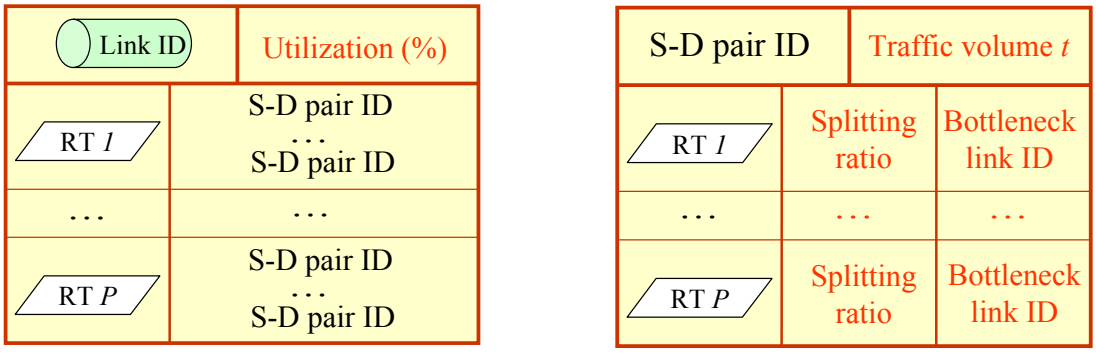
3.1.2.2 Adaptive traffic control

In this section, we present an efficient algorithm for adaptive adjustment of traffic splitting ratio at individual PoP source nodes. In a periodic fashion at a relatively short-time interval (e.g., hourly), the central TE manager needs to perform the following three operations:

- Measure the incoming traffic volume and the network load for the current interval.
- Compute new traffic splitting ratios for all PoP nodes based on the measured traffic demand and the network load for dynamic load balancing.
- Instruct individual PoP nodes to enforce the new traffic splitting ratio over their locally originated traffic.

To fulfill the second task, a traffic engineering information base (TIB) is needed by the TE manager to maintain necessary network states based on which new traffic splitting ratios are computed. Figure 3 presents the structure of our proposed TIB, which consists of two inter-related repositories, namely the Link List (LL) and the S-D Pair List (SDPL). LL maintains a list of entries for individual network links. Each LL entry records the latest monitored utilization of a link and the involvement of this link in the IGP paths between associated S-D pairs in individual RTs. More specifically, for each RT, if the IGP path between an S-D pair includes this link, then the ID of this S-D pair is recorded in the LL entry. It is worth mentioning that this involvement information remains static after the MT-IGP link weights have been configured (static information is presented in black in Figure 3 while dynamic information that needs to be updated periodically at short time scale is in red). On the other hand, SDPL consists of a list of entries, each for a specific S-D pair with the most recently measured traffic volume from S to D. Each SDPL entry also maintains a list of subentries for different RTs, with each recording the splitting ratio of the traffic from S to D, as well as the ID of the bottleneck link along the IGP path for that S-D pair in the corresponding topology.

During each *ATC* interval, the TIB is updated under two events. First, upon receiving the link utilization report from the network monitoring component, the TE manager updates the link utilization entry in the LL and the ID of bottleneck link for each S-D pair under each RT in SDPL. On the other hand, when the adaptive traffic control phase is completed and the new traffic splitting ratios are computed, the splitting ratio field in SDPL is updated accordingly for each S-D pair under each RT.



(a) Entry structure of LL

(b) Entry structure of SDPL

Figure 3 Traffic Engineering Information Base structure

We start by defining the following parameters:

$t(u,v)$ – traffic between PoP node u and v.

$\phi_{u,v}(r)$ – traffic splitting ratio of $t(u,v)$ at u on routing topology r, $0.0 \leq \phi_{u,v}(r) \leq 1.0$.

The algorithm consists of the following steps. We define an iteration counter k which is set to zero initially.

Step-1: Identify the most utilized link l_{max} in the network, which can be simply achieved by visiting the updated LL in the TIB.

Step-2: For the set of S-D pairs whose traffic flows are routed through l_{max} in *at least one but not all* the routing topologies, consider each at a time and compute its new traffic splitting ratio among the routing topologies until the first feasible one is identified. A feasible traffic flow means that, with the new splitting ratios, the utilization of l_{max} can be reduced without introducing new hot spots with utilization higher than the original value. To support this operation, all feasible S-D pairs that meet the above requirement are identified from the entry of l_{max} in the LL.

Step-3: If such a feasible traffic flow is found, accept the corresponding new splitting ratio adjustment. Increment the counter y by one and go to Step-1 if the maximum K iterations have not been reached (i.e. $y \leq K$). If no feasible traffic flow exists or $y = K$, the algorithm stops and the latest resulting values for traffic splitting ratio is configured in the corresponding entry in the SPDL in order to be executed by individual source PoP nodes.

The parameter K controls the algorithm to repeat at most K iterations in order to avoid long running time. In Step-2, the task is to examine the feasibility of reducing the load of the most utilized link by decreasing the splitting ratios of a traffic flow assigned to the routing topologies that use this link, and shift a proportion of the relevant traffic to alternative paths with lower utilization in other topologies. More specifically, the adjustment works as follows. First of all, a deviation of traffic splitting ratio, denoted by δ where $0.0 < \delta \leq 1.0$, is taken out for trial. For the aggregate traffic flow $t(u,v)$ under consideration, let R^+ be the set of routing topologies in which the IGP paths from u to v traverse l_{max}. The main idea is to decrease the sum of traffic splitting ratios on all the routing topologies in R^+ by δ and at the same time to increase the sum of the ratios on other topologies that do not use l_{max} by δ (We denote this set of topologies by R^- where $R^- = R \backslash R^+$). Specifically, for all the topologies in R^+, which share a common link with the same (maximum) utilization, their traffic splitting ratios are evenly decreased. Hence, the new traffic splitting ratio for each routing topology in R^+ becomes:

$$\phi_{u,v}(r)' = \phi_{u,v}(r) - \delta / |R^+| \qquad \forall r \in R^+$$

On the other hand, let μ_r be the bottleneck link utilization of the IGP path in routing topology $r \in R^-$. The traffic splitting ratio of each routing topology in R^- increases in an inverse proportion to its current bottleneck link utilization, i.e.

$$\phi_{u,v}(r)' = \phi_{u,v}(r) + \left(\frac{1-\mu_r}{\sum_{r \in R^-} 1-\mu_r} \times \delta\right) \quad \forall r \in R^-$$

The lower (higher) the bottleneck link utilization, the higher (lower) the traffic splitting ratio will be increased.

An important issue to be considered is the value setting for δ. If not appropriately set, it may lead to either slow convergence or overshoot of the traffic splitting ratio, both of which are undesirable. On one hand, too large value of δ may miss the chance to obtain desirable splitting ratios due to the large gap between each trial. On the other hand, too small (i.e. too conservative) value of δ may cause the algorithm to perform many iterations before the most appropriate value of δ is found, thus causing slow convergence to the equilibrium. Taking these considerations into account, we apply an algorithm to increase δ exponentially starting from a sufficiently small value. If this adjustment is able to continuously reduce the utilization of l_{max} without introducing negative new splitting ratios on R^+, the value of δ will be increased exponentially for the next trial until no further improvement on the utilization can be made or the value of δ reaches 1.0 (i.e. the maximum traffic splitting ratio that can be applied). The exponential increment of δ works as follows.

$$\delta = \frac{1}{2^{\Omega-\omega}}$$

where Ω is a constant that can be set by the network operator, and ω is the iteration counter. The pseudo code for the algorithm is shown in Figure 4.

The time complexity of the proposed algorithm is as follows. Step-1 of the algorithm can be simply done by searching all the links in the network and thus it takes $O(|E|)$. The worst-case scenario of Step-2 is to try all the traffic flows (maximum $|V|\times(|V|-1)$) until the last one is found to meet the adjustment requirement, or even none at all. In addition, for each traffic flow, there are two operations involved: (1) try at most Ω iterations to find the most appropriate value of δ and (2) adjust the traffic splitting ratio for each routing topology. Thus, Step-2 could take $O(\Omega|V|^2|R|)$. Comparing between step 1 and 2, the latter dominates the complexity as $|V|^2 >> |E|$ given the fact that the mean node degree of today's PoP level network topologies is small (3.5 on average [SPRI04]). Finally, since the algorithm runs at most K iterations, the overall computational complexity of the proposed algorithm is $O(K\Omega|V|^2\ |R|)$. We have tested the running time of our adaptive TE algorithm on both GEANT [GEANT] and Abilene [ABILE] networks. On average, it takes less than a second to compute the optimized traffic splitting ratios for each traffic matrix. This is acceptable for the adaptive TE in short time-scale such as hourly or even in minutes.

Notation: $U(l)$ is the utilization of link l

Require: A set of MT-IGP topologies R, constants K and Ω

1. $glb\_improve = TRUE$, $k = 0$
2. **while** ($glb\_improve$ & $k < K$) **do**
3. $l_{max} \leftarrow$ the most utilized link in the network
4. Let T' be the set of traffic flows routed over l_{max} in at least one but not all of the routing topologies
5. $t(u,v) \leftarrow$ the first traffic flow in T'
6. $feasible\_fnd = FALSE$
7. **while** (!$feasible\_fnd$ & not all flows in T' are examined) **do**
8. $R^+ \leftarrow$ the set of routing topologies that uses l_{max} for $t(u,v)$
9. $R^- \leftarrow R \setminus R^+$, $\omega = 0$, $\mu_{max} = U(l_{max})$
10. $best\_dlt = 0$, $loc\_improve = TRUE$
11. **while** ($\omega \le \Omega$ & $cont$) **do**
12. $\delta = \frac{1}{2^{\Omega-\omega}}$
13. $\phi_{u,v}(r)' = \phi_{u,v}(r) - \delta / |R^+| \quad \forall r \in R^+$
14. $\mu_r \leftarrow$ the bottleneck link utilization of the path for $t(u,v)$ in topology $r \in R^-$
15. $\phi_{u,v}(r)' = \phi_{u,v}(r) + (\frac{1-\mu_r}{\sum_{r\in R^-} 1-\mu_r} \times \delta) \quad \forall r \in R^-$
16. $l'_{max} \leftarrow$ the most utilized link among those traversed by $t(u,v)$ in all the routing topologies if $\phi_{u,v}(r)'$ is to be implemented
17. **if** ($U(l'_{max}) < \mu_{max}$ & $\phi_{u,v}(r)' \ge 0 \ \ \forall r \in R^+$) **then**
18. $\mu_{max} = U(l'_{max})$, $best\_dlt = \delta$, $\omega = \omega + 1$
19. **else**
20. $cont = FALSE$
21. **end if**
22. **end while**
23. **if** $\mu_{max} < U(l_{max})$ **then**
24. accept the adjusted splitting ratios based on $best\_dlt$
25. $feasible\_fnd = TRUE$
26. $k = k + 1$
27. **else**
28. $t(u,v) \leftarrow$ next traffic flow in T'
29. **end if**
30. **end while**

31. $l^*_{max} \leftarrow$ the current most utilized link in the network
32. **if** $U(l^*_{max}) \geq U(l_{max})$ **then**
33. $glb\_improved = FALSE$
34. **end if**
35. **end while**

Figure 4 Pseudo code - Adaptive traffic splitting ratio adjustment algorithm

3.1.2.3 Network monitoring

Network monitoring, which is responsible for collecting up-to-date network conditions, plays an important role for supporting *ATC* operations. *AMPLE* adopts a hop-by-hop based monitoring mechanism that is similar to the proposal of [ASGA04]. The basic idea is that, a dedicated monitoring agent deployed at every PoP node is responsible for monitoring: (1) the volume of the traffic originated by the local customers towards other PoPs (intra-PoP traffic is ignored), and (2) the utilization of the directly attached inter-PoP links. As shown in Figure 5, this monitoring agent gathers data on the locally originated traffic volume from all the access routers (ARs) attached with customers inside the PoP. Meanwhile the agent also collects the utilization of the directly attached inter-PoP links from individual backbone routers (BRs). In a periodical fashion (e.g. hourly), the central TE manager polls each individual monitoring agent within each PoP and collects their locally monitored traffic volume and link utilizations. These statistics are used by the central TE manager for updating its maintained traffic engineering information base (TIB, to be specified in the next section) and computing traffic splitting ratios for the next interval. Such a hop-by-hop based paradigm works efficiently in a TE system with a central manager. This is in contrast to the edge-to-edge based paradigms in a pure distributed fashion, where local decisions on traffic adjustment at individual source nodes may conflict with each other due to their local visibility of the network condition. As a result, traffic oscillation and network instability may occur.

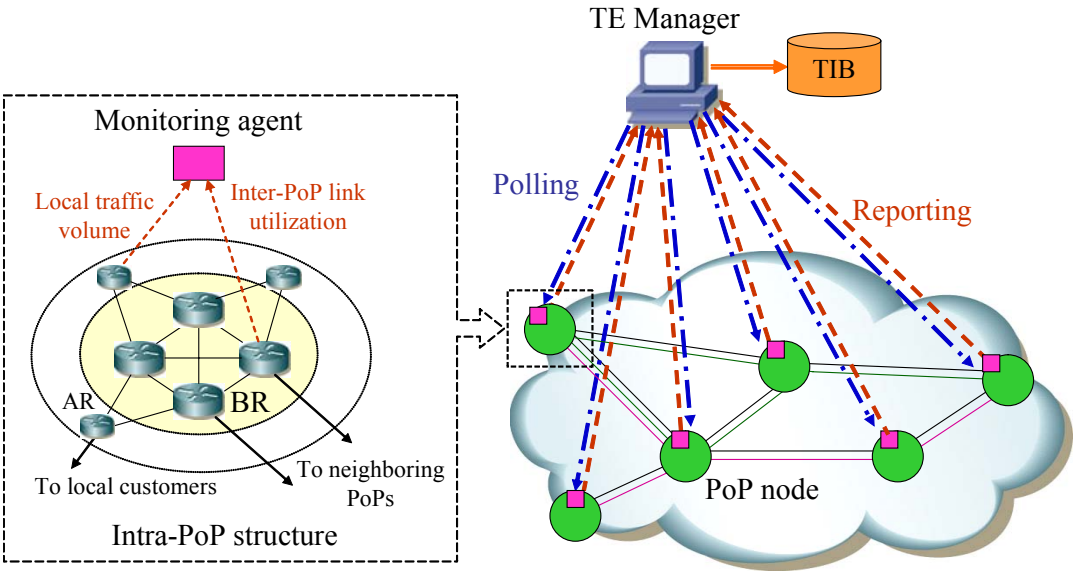


Figure 5 Network Monitoring and ATC

3.1.2.4 Woking as a whole system

After presenting the detailed information on individual components, now we briefly describe how they work altogether as a whole TE system. First of all, optimized MT-IGP link weights are configured on top of the underlying multi-topology routing platform and remain static till the next offline TE operation begins. During this period, *ATC* plays the major role for adaptively re-balancing the load according to the traffic dynamics in short-time interval. In response to the periodical polling requests

from the TE manager, the monitoring agents attached to individual PoP nodes, report back the incoming traffic volume (from access routers) and inter-PoP link utilizations (from backbone routers). The TE manager accordingly updates the traffic volume between each S-D pairs in the SDPL and link utilization information stored in the LL of the TIB. According to the obtained link utilization information, the bottleneck link ID along the IGP paths between individual S-D pairs in each RT is also updated in the SDPL. Based on the updated information the TE manager computes the new traffic splitting ratio for each S-D pair across individual routing topologies. These new splitting ratios are configured in the SDPL and the TE manager then instructs all the source PoP nodes within the network using these new values for traffic splitting during the next interval. Meanwhile these values stored in the SDPL will also be used as the starting point for the future computation of the splitting ratios in the next interval. Once each source PoP node has received the new values for traffic splitting from the central TE manager, it enforces them by remarking the MT-ID of the locally originated traffic in the new proportions across individual routing topologies.

3.1.3 MTR across multiple Network Planes

In this section we briefly introduce how MTR can be used to implement multiple Network Planes, each with specific QoS requirements. For simplicity we use edge-to-edge delay as the QoS metric associated with each NP. According to our strategy, service differentiation in terms of delay across multiple NPs is taken into account at the *OLWO* component where MT-IGP link weights are optimised for each routing topology. More specifically, the setting of link weights takes into account the edge-to-edge propagation delay of individual topologies. The delay for each network link l is denoted as d_l and the overall edge-to-edge delay constraint for each NP k is denoted as Δ_k. The edge-to-edge delay-constrained MT-IGP link weight optimization problem is formally described as follows. For each Network Plane k, to calculate $|R|$ sets of positive link weights $W(r) = \{w_l(r)\}: w_l(r) > 0,\ r \in R$ in order to:

$$\text{minimize} \qquad \sum_{u,v \in V} \sum_{l \in E} FDoI_l^{u,v}$$

subject to:

$$D_{u,v}(r) = \sum_{l \in P_{u,v}(r)} d_l \le \Delta \quad \forall u,v \in V \quad and \quad \forall r \in R$$

In order to satisfy this additional constraint, the Genetic Algorithm based optimisation of MT-IGP link weights introduced in 3.1.2.1 needs to be extended accordingly. Towards this end, the original fitness function defined in 3.1.2.1 is changed into:

$$\frac{C}{\sum_{u,v \in V} \sum_{l \in E} FDoI_l^{u,v} + \lambda \times \overline{D}}$$

where

$$\overline{D} = \begin{cases} Max(D_{u,v}(r)) - \Delta & if\ Max(D_{u,v}(r) > \Delta, \forall u,v \in V, r \in R \\ 0 & otherwise \end{cases}$$

The MT-IGP link weight optimisation using GA is performed independently for each NP with different value of Δ. Finally it is worth mentioning that the *ATC* component does not need to take into account the edge-to-edge delay constraint across individual NPs when short timescale traffic splitting adjustment is performed.

3.2 INP level overlay routing (Intra-domain considerations)

3.2.1 Overview

How to provide QoS assurance to the contracted CPA with customers is one of the key issues to be considered when engineering NPs. QoS degradation can be attributed to many reasons. One common cause of such deterioration is network failure, which becomes part of daily operations in most of IP networks. When a link or router within a network fails, the incident routers running IGP routing protocols like OSPF disseminate new link state advertisements (LSA) throughout the network to notify the failure. On receiving the updated LSA, each router re-computes its IGP routing by removing the failed component(s) from the original network topology, based on which the updated routing table is populated. This process is known as IGP re-convergence.

Unfortunately, it has been shown that network-wide IGP re-convergence may take long time to complete [ALAE00], and there is inevitably a period of disruption to the delivery of customer traffic until the entire network re-converges on the new topology. During this period, individual routers may have inconsistent views on the overall network topology and therefore transient forwarding loops can be formed. The common experiences have suggested that up to 50-millisecond duration of loss of connectivity (LoC) normally cannot be noticed by end users using real-time multimedia applications. However, IGP re-convergence in operational networks normally cannot be completed within that short-time duration. To remedy the problems caused by IGP re-convergence, an effective solution has been proposed to recover network failures in a very short time to avoid noticeable service disruptions. The solution is that, once a router detects the failure of its adjacent network component (e.g. a link or a neighbouring router), it immediately reroutes the affected traffic to a pre-computed repair path through which the traffic is forwarded to the destination, while suppressing the dissemination of the LSA on the failure. This operation is known as IP Fast Re-Route (FRR).

Whilst IP FRR is a control/data plane technique for achieving fast recovery from routing failures, it does not consider traffic re-optimization, for instance how to re-balance the overall traffic loading after the affected traffic is re-routed onto the repair paths. Without such consideration on traffic control across individual repair paths, although failures can be bypassed quickly, there could be an overwhelming amount of traffic re-routed through some repair paths, which leads to congestion on some parts of the network and eventually causes packet delay or loss. As a result, the efforts made by IP FRR techniques still lead to nowhere as QoS assurance still cannot be supported. To provide reliable QoS assurance under failures, not only fast recovery using FRR techniques but also provisioning of repair paths that optimizes post-failure network performance should be considered in conjunction.

Today, IP traffic engineering (TE) has been investigated widely in the research community, including traffic optimization under both the normal state and the post-failure scenario. Nevertheless, it is noted that all relevant TE schemes that take into account network failures only consider the ordinary IGP re-convergence scenario. Given the added complexity for achieving IP FRR, these existing approaches cannot be directly transplanted. In this sense a holistic solution to achieve IP FRR together with avoidance of post-failure traffic congestion is still yet to be investigated. To fill this gap, we propose a novel scheme to achieve comprehensive QoS assurance by considering post-failure load balancing for IP FRR. More specially, we propose a tunnel-based mechanism as the underlying IP FRR platform in the control/data plane.

The proposed mechanism makes use of intermediate routers, often known as tunnel endpoints to re-route traffic towards the final destination without traversing the failures. To perform FRR, a router that is adjacent to the failure, which is also called repairing router, tunnels the affected traffic to a tunnel endpoint from where the traffic is decapsulated and forwarded natively to the final destination. A notable observation is that, for a given network topology with specific IGP link weight configuration, multiple intermediate routers within the network may exist as candidates for feasible tunnel endpoints. In this case an opportunity exists for the network operator to perform optimized selection of tunnel endpoints for achieving post-failure load balancing if the overall traffic matrix can be estimated a priori. We propose an efficient optimization algorithm for the tunnel endpoint selection in order to

achieve a comprehensive paradigm for supporting high QoS assurance. The ultimate objective is to minimize the Maximum Link Utilization (MLU) that takes into account every single link failure scenario. More specifically, based on the overall network topology and the estimated traffic matrix, a tunnel endpoint is selected for each affected destination with regard to each link to be protected. The goal is to re-balance the overall traffic loading after the traffic is rerouted over the repair paths. All the selected tunnel endpoints need to be pre-configured by the network operator at each individual repairing router such that they can be immediately activated once the failure of the protected network component is detected.

3.2.2 Tunnel-based IP fast reroute

3.2.2.1 Motivation

Although several IP FRR mechanisms have been proposed in literature, few have considered how to optimize post-failure network performance on top of these schemes. As already mentioned, congestion may occur after the affected traffic is re-routed onto the repair paths, which nullifies the effectiveness of using IP FRR. We therefore focus on optimizing post-failure network performance for IP FRR.

A generic tunnel-based IP FRR mechanism is proposed for implementation of NPs that require high QoS assurance against network failures. This mechanism shares some similarity with the one proposed in [BRYA07] since both use tunnel encapsulation for implementing the repair path. However, there are several key differences. First of all, our mechanism allows the use of dedicated tunnel endpoints for the repair paths to different destinations, while the existing mechanism uses only a single tunnel endpoint for all the affected destinations. Such a per-destination based scheme, which has also been used by Loop-free Alternates (LFA) [ATLA08] and also [NELA07], provides higher flexibility in provisioning repair paths. Furthermore, in our mechanism, a tunnel endpoint always forwards the traffic natively to the final destination without relying on the additional direct forwarding mechanism, which cannot be naturally supported by conventional IP routers.

We note that although our work focuses on tunnel-based IP FRR mechanism, the idea of optimizing post-failure network performance can also be adapted to other IP FRR mechanisms, e.g. judicious selection of direct neighbours for LFA. However, this technique may not be directly applicable to the Not-via approach, as tunnel endpoint is always fixed to the one that is on the far side of the failed network component.

3.2.2.2 Operation and illustrative example

Our tunnel-based IP FRR mechanism allows repairing routers to be pre-configured with tunnel endpoints that are able to detour traffic from the protected link before reaching its final destination. The overall repair path consists of two shortest path segments: one from the repairing router to the tunnel endpoint (the tunnel) and the other from the tunnel endpoint to the final destination. Figure 6 illustrates this repair path.

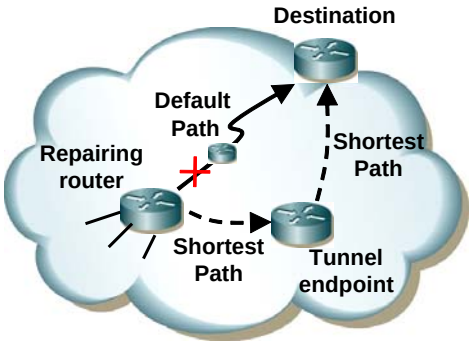


Figure 6 Repair path using the tunnel-based IP FRR mechanism

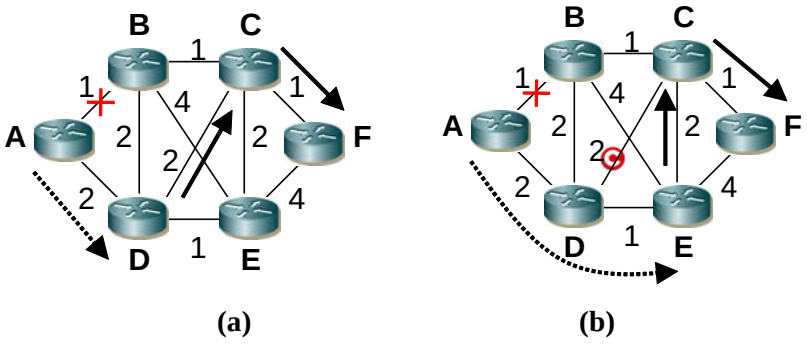


Figure 7 Illustrative example of the tunnel-based IP FRR mechanism

An example of our proposed scheme is illustrated in Figure 7(a). Given the set of IGP link weights shown in the figure, the shortest path from router *A* to *F* is *A-B-C-F*. At router A, *D* can be selected as the tunnel endpoint for the repair path that protects link A-B with regard to the traffic towards *F*. In case link *A-B* fails, the repairing router *A* immediately re-routes the traffic away from *B* to the tunnel endpoint *D* via the IP tunnel (i.e. *A-D*). Next, *D* de-encapsulates the packets and forwards the traffic natively to the final destination *F* based on the conventional IP shortest path routing (i.e. the path *D-C-F*).

However, if link *D-C* becomes congested due to the diversion of the affected traffic from the repairing router *A*, router *D* may not be a good choice of tunnel endpoint in the first place. To avoid potential post-failure congestion, router *E* may be used as the tunnel endpoint instead of *D* as shown in Figure 7(b). In this case, the traffic is re-routed onto the repair path *A-D-E-C-F* without traversing link D-C that is prone to congestion. This example shows that our tunnel-based IP FRR scheme provides flexibility in optimizing post-failure network performance by judicious selection of tunnel endpoint. To achieve optimized post-failure traffic distribution with IP FRR, the network operator needs to obtain the following information a priori in order to perform optimized tunnel endpoint selection in an offline manner: the overall network topology including the IGP link weight setting, the forecasted traffic matrix and the distinct failure scenarios to be protected. This is very similar to the input for the robust IGP traffic engineering proposed by [FORT03].

3.2.2.3 Implementation

There are several existing mechanisms for the implementation of IP tunnel in the data plane. A possible mechanism is IP-in-IP encapsulation [RFC1853]. In this case, the IP address of final destination is encapsulated in the payload of an outer IP header that contains the IP address of tunnel endpoint in its destination field. When a packet reaches a tunnel endpoint, the outer IP header is stripped off, and the original IP packet is injected into the IP stack of the tunnel endpoint. The other ways of IP tunnel implementation include GRE [RFC2784] and L2TPv3 [RFC3931]. Once again we emphasize that our proposed tunnelling approach does not rely on other additional mechanisms such as directed forwarding and un-natural deflection of traffic to an alterative next hop towards the tunnel endpoint as proposed in [BRYA07].

3.2.3 Tunnel endpoint selection

3.2.3.1 Problem formulation

Given a specific network topology with configured IGP link weights, for each link to be protected by failure, the repairing router may have multiple choices for selecting tunnel endpoint, and each could result in different post-failure network utilization. To minimize the possibility of creating post-failure network congestion, it is important to judiciously pre-determine the best tunnel endpoint such that the load distribution in the network after failure is balanced. We name this IP FRR tunnel endpoint

selection problem. We focus on single link failures but the proposed scheme can be easily adapted to router failures as well.

We now formally define the tunnel endpoint selection problem. Let the network topology be represented as a graph $G=(V,E)$ with a set of routers V and a set of unidirectional edges E with $e(x,y)$ representing the link connected from router x to y. Based on the configured IGP link weights, the shortest path from router x to y is denoted by $x \rightarrow y$. Let $f_{x,y} \subseteq V \times V$ be the traffic that is sent from router x to destination y. Note that $f_{x,y}$ includes not only the traffic that is locally originated from x but also from the other routers in the network which must traverse x before reaching y. The task of the tunnel endpoint selection problem is as follows:

For each adjacent link to be protected at each repairing router x, select a tunnel endpoint, denoted by $t_{x,y}$, for each affected destination y so that $f_{x,y}$ will be rerouted over $x \rightarrow t_{x,y} \rightarrow y$ when the protected link fails. An affected destination means that the shortest path from the repairing router to it involves the protected link. The ultimate goal is to avoid post-failure network congestion on the repair path due to careless selection of $t_{x,y}$.

We define Maximum Link Utilization (MLU) to be the utilization of the highest loaded link within the network. Under the failure of link $e(u,v) \in E$, let $\mu_{u,v}$ be the post-failure MLU after router u has rerouted the traffic for all the affected destinations via the selected tunnel endpoints. Since the tunnel endpoint selection is performed for each protected link independently due to single link failure protection, the optimization objective of the tunnel endpoint selection problem is to minimize the post-failure MLU for each of these scenarios, which is defined as

$$Minimize \ \ \mu_{x,y} \qquad \forall e(x,y) \in E$$

3.2.3.2 Heuristic algorithm

We propose an efficient algorithm for solving the tunnel endpoint selection problem. The proposed algorithm consists of two phrases.

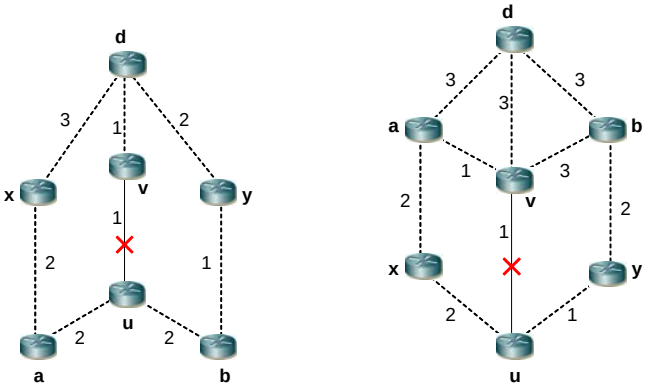
First Phrase: Feasible Tunnel Endpoint Filtering

Although any router in the network could be considered as tunnel endpoint candidate, some may cause forwarding loops and therefore are infeasible. The first step of our algorithm is to identify all the feasible tunnel endpoints for each protected link by its repairing router with regard to each affected destination. Let u and v be the head (i.e. repairing) and tail router of the link $e(u,v)$ to be protected respectively, d be the destination router, $w(u, v)$ be the IGP weight of link connecting from router u to v, and finally $dist(x, y)$ be the total IGP cost of $x \rightarrow y$. If a router is a feasible tunnel endpoint, two necessary conditions must be met:

***Constraint 1 (Not hidden behind repairing node)*:** For any router o in the network to be a feasible tunnel endpoint for u to reach destination d, u must *not* be on $o \rightarrow d$. That is:

$$dist(o, u) + dist(u, d) > dist(o, d)$$

***Example*:** As shown in Figure 8(a), router a is considered as an infeasible tunnel endpoint candidate for the protected link u-v with regard to the destination d. This is because once packets are de-capsulated at a, they will be attracted back to the repairing router u on their way to d. Router b is a feasible candidate since $b \rightarrow d$ does not involve the protected link.



(a) Constraint 1 **(b) Constraint 2**

Figure 8 Constraints for tunnel endpoint filtering

***Constraint 2 (Not hidden behind tail of protected link)*:** For any router o in the network to be a feasible tunnel endpoint candidate for u to reach d, v must not be on $u \rightarrow o$. That is:

$$w(u, v) + dist(v, o) > dist(u, o)$$

***Example*:** As shown in Figure 8(b), router a is considered as an infeasible tunnel endpoint candidate for the protected link u-v with regard to the destination d. This is because the tunnel from the repairing router u to a still traverses the protected link. Router b is a feasible candidate since $u \rightarrow b$ does not involve the protected link.

Second Phrase: Tunnel Endpoint Selection

Given the set of feasible tunnel endpoints identified in the first phase, the second phase of the algorithm is to select the best tunnel endpoint such that the post-failure MLU under the considered link protection scenario is minimized.

The basic idea of the second phrase is to first identify all the affected destinations for each of the adjacent links to be protected. Then, for each of these destinations, select the best feasible tunnel endpoint in a greedy fashion with the objective to minimize the corresponding MLU assuming the failure of the protected link. The detailed steps of the algorithm are as follows.

Input 1: A set of feasible tunnel endpoints to each affected destination for each protected link
Input 2: Network topology and traffic matrix
Step 1: Set Ω to be the current network (normal) status.

Step 2: For router x, consider a directly attached link to be protected.

Step 3: Identify all destinations $y \in V$ where the shortest paths $x \rightarrow y$ traverse the protected link. Then, remove their traffic $f_{x,y}$ from $x \rightarrow y$.

Step 4: Sort all the destinations in descending order according to their associated traffic volume $f_{x,y}$.

Step 5: For each destination y in that order,

if there exist feasible tunnel endpoints for y, then

– try to route $f_{x,y}$ to the destination via each of the feasible tunnel endpoints independently and records the corresponding post-failure MLU.

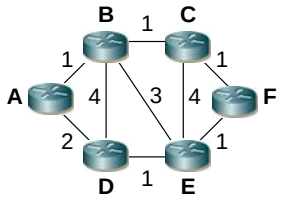
– select the one that results in the least MLU as the tunnel endpoint $t_{x,y}$.

– update the network by routing $f_{x,y}$ over $x \rightarrow t_{x,y} \rightarrow y$.

Step 6: Restore the current network status to Ω.

Step 7: Go to step 3 to consider the next adjacent link to be protected until all the adjacent links of router *x* have been processed.

3.2.3.3 Illustrative example



Dest	Next Hop
B	*B*
C	*B*
D	*D*
E	*D*
F	*B*

Figure 9 Network topology and routing table of router *A*

Protected link	Dest	Feasible tunnel endpoint
A-B	*B*	*E*
	C	*D or E*
	F	*D or E*
A-D	*D*	*C or F*
	E	*B or C or F*

Table 1 Example of feasible tunnel endpoint filtering

Protected link	Dest	Selected tunnel endpoint
A-B	*B*	*E*
	C	*D*
	F	*E*
A-D	*D*	*C*
	E	*F*

Table 2 Example of tunnel endpoint selection result

To get better understanding of our algorithm, we illustrate its operations using an example in Figure 9. We consider router *A* in the network to be the repairing router of its two directly attached links *A-B* and *A-D*. *A*'s routing table is shown next to the figure.

The algorithm starts with identifying all the feasible tunnel endpoints to each affected destination according to the two filtering criteria. For example, for destination *B*, *A-B* is the next hop adjacent link along the shortest path which needs to be protected. In this case, *C* and *F* are not feasible due to the violation of condition (2) as the traffic from *A* to these routers traverse the protected link. *D* is not feasible either due to the violation of condition (1) as the repair path from *D* to *B* (i.e. *D-A-B*) traverse the protected link. Nevertheless, *E* is feasible for the traffic to reach *B* from *A* as the repair path (i.e. *A-D-E-B* or *A-D-E-F-C-B*) does not traverse the protected link. Figure 9 shows all feasible tunnel endpoints for each destination at router *A* to protect each of its adjacent links. This procedure repeats at each router in the network for every destination.

The next step of the algorithm is tunnel endpoint selection to achieve post-failure load balancing. Given the set of feasible tunnel endpoints, the algorithm proceeds as follows. First of all, consider an adjacent link of the repairing router to be protected, e.g. link *A-B* by router *A*. In this case, traffic for *B*, *C* and *F* is affected as their shortest paths traverse the link. Given $f_{A,B}$, $f_{A,C}$ and $f_{A,F}$, the algorithm removes these traffic from the network and then performs a sorting according to their traffic volume.

Assuming that the sorting order is $f_{A,B}$, $f_{A,C}$, $f_{A,F}$. For the first destination in that order (i.e. B), the algorithm selects between D and E as the tunnel endpoint. By trying each of these tunnel endpoints one at a time over the corresponding repair paths (i.e. $A \rightarrow D \rightarrow B$ and $A \rightarrow E \rightarrow B$), the one that results in the least MLU is selected. If, for example, D is selected, $f_{a,b}$ will be routed in the network over $A \rightarrow D \rightarrow B$. Given this updated network topology, the next destination C is tried using the above procedure until the last destination F has been considered. Figure 9 shows an example of tunnel endpoint selection result. The algorithm repeats for each other router in the network independently. The overall complexity of our algorithm is $O(EV^2)$.

3.3 DiffRout NP engineering based on MRDV

This section describes how the extension of MRDV (Multipath Routing with Dynamic Variance) to support multiple traffic classes, described in [D3.1], is used to provide DiffRout (Differentiated Routing) NP engineering.

The different network planes are built by mapping a general NP-independent routing table, obtained from the link state protocol information, to NP-specific routing tables. This mapping is done by using the MRDV algorithm, which is based on the concept that suboptimal paths to route traffic towards a destination are used when optimal paths are close to be congested.

Firstly, the behaviour of MRDV with a single network plane is explained in detail in order to understand the internals of the MRDV algorithm. Next, the extension of MRDV to multiple network planes is described.

3.3.1 MRDV with a single network plane

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 towards 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:

$$M \leq M_{\min} \cdot V \quad \textbf{(1)}$$

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.

Depending on 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 lower 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:

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

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. A value of 30 seconds has been chosen for the implementation since it is high enough to avoid an unstable behaviour caused by the hysteresis cycle and low enough to detect changes in link load that can lead to link congestion.

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.

Two types of loops can be distinguished attending to the number of necessary hops to complete the loop:

- **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 10(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 10(b) shows a loop between A and B where there is an optimal path from 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 10(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.

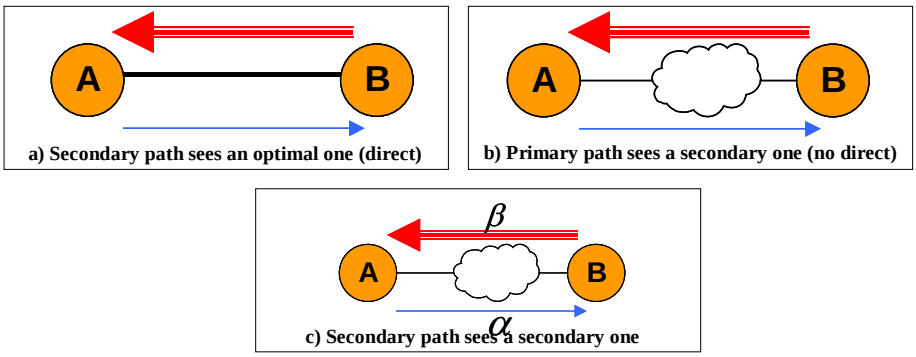


Figure 10 Types of loops

Taking into account this classification, two different mechanisms can be used when a node is going to install a new secondary path:

- **Avoidance of primary loops**. It only requires a simple process to be computed at each router. Whenever 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.
- **Avoidance of secondary loops**. A communication between routers A and B is necessary in order to infer the forwarding and return proportions, α and β (Figure 10(c)). For this purpose, a new protocol that allows both routers to implement the information exchange, the LAPM (Loop Avoidance Protocol Message), was defined. The information exchange allows inferring α and β, and once these are inferred, if β is greater than α, the secondary path is installed.

Simulations showed that the avoidance of secondary loops provided just slightly better results than the avoidance of primary loops. Therefore, it was decided not to include the avoidance of secondary loops in the final specification.

3.3.2 MRDV with multiple network planes

This section describes the MRDV with CoS support, an extension of MRDV to support multiple network planes differentiated by routing.

> *"Telefonica I+D has decided not to make public the MRDV with CoS support at this time as it is subject to IPR protection activities and a possible commercial exploitation. Once IPR has been protected, the material will be published"*

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.

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 11).

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].

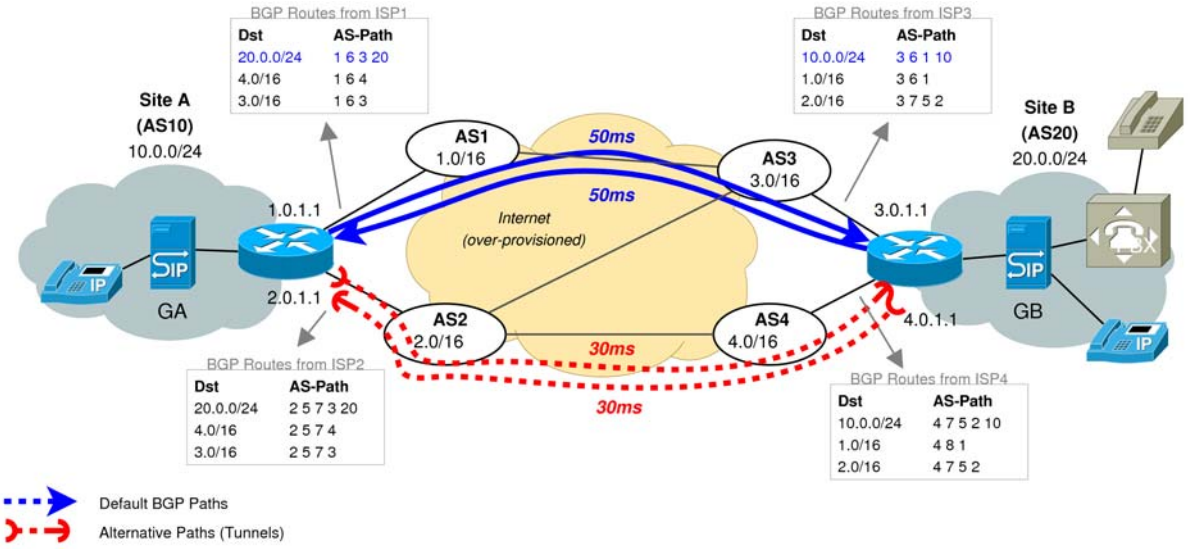


Figure 11 Using tunnels to improve the latency between two SIP gateways.

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 11. 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 12 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 12, 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.

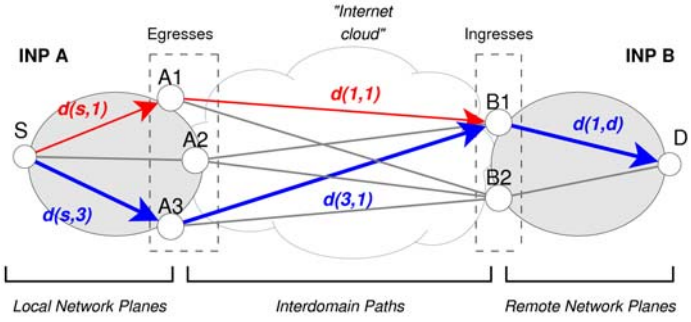


Figure 12 Multiple paths between two AS.

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 13). 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 [RFC4915, RFC5120] 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).

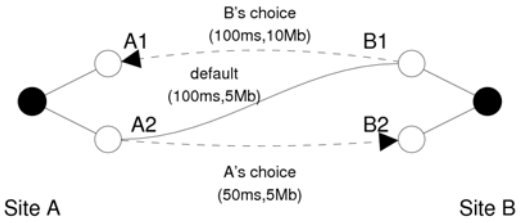


Figure 13 Conflicting objectives.

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 14 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.

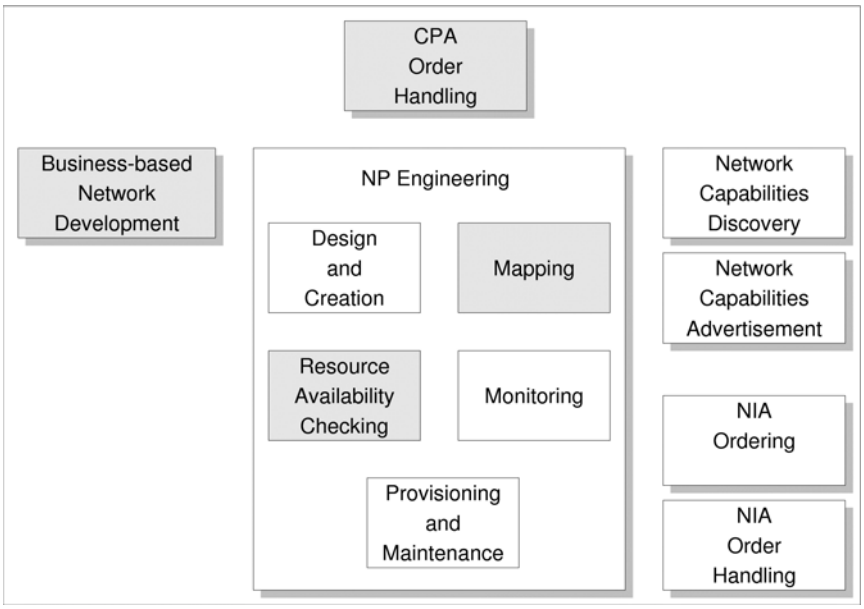


Figure 14 Functional components involved in the IP Tunneling solution.

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 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 16. 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 16 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.

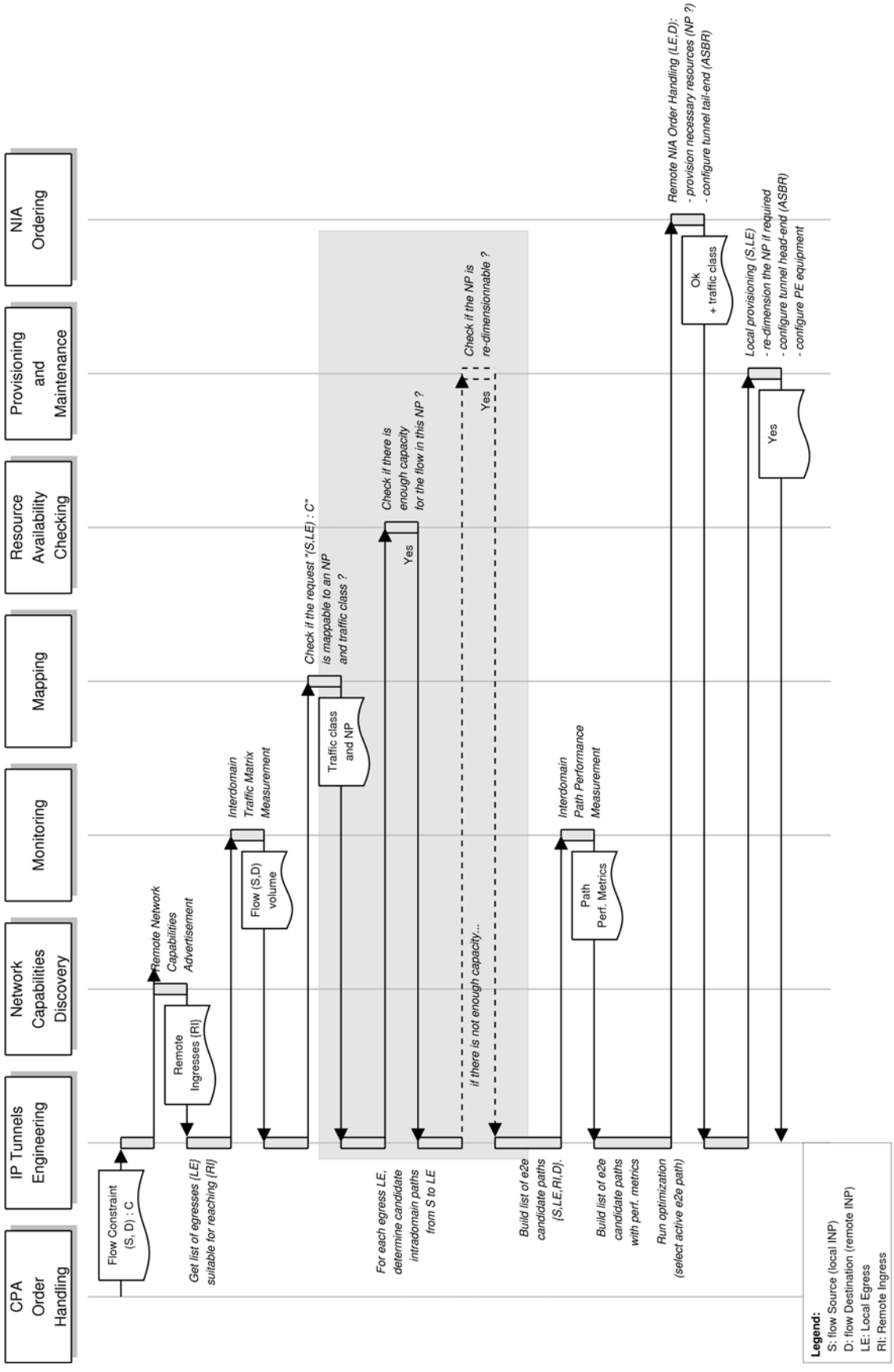


Figure 15 Flowchart of the handling of a CPA order for outgoing traffic.

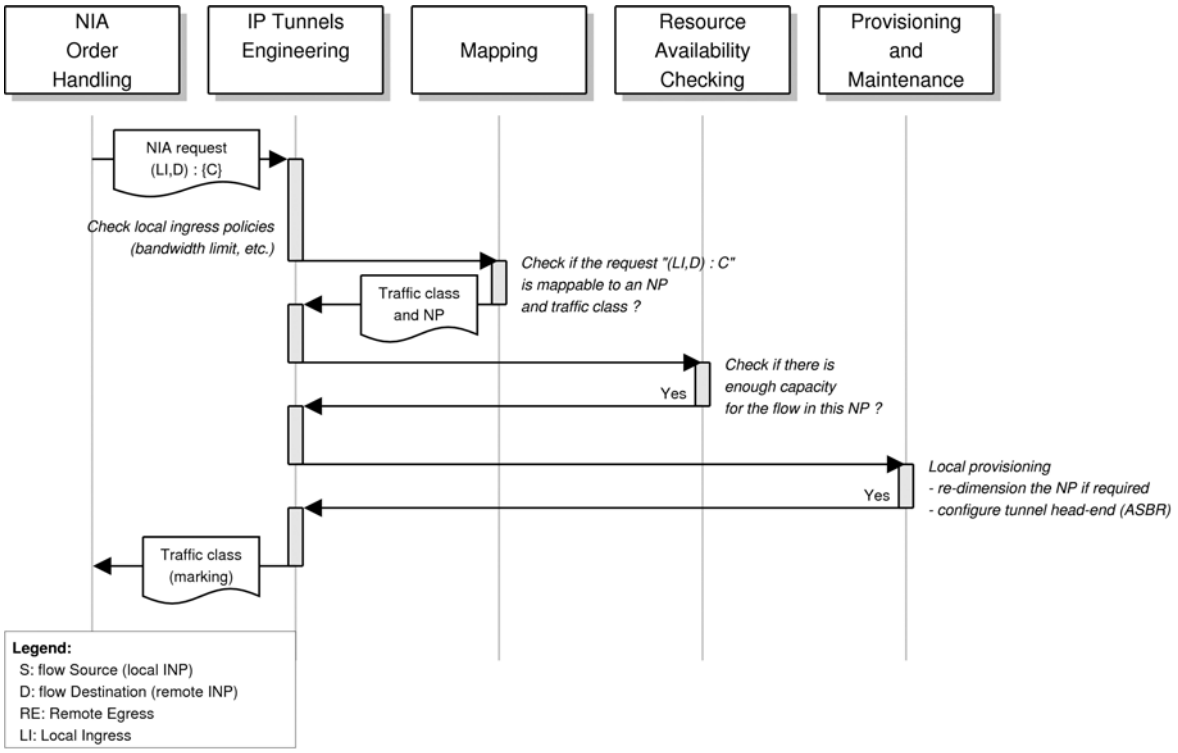


Figure 16 Flowchart of the handling of an NIA order for incoming traffic.

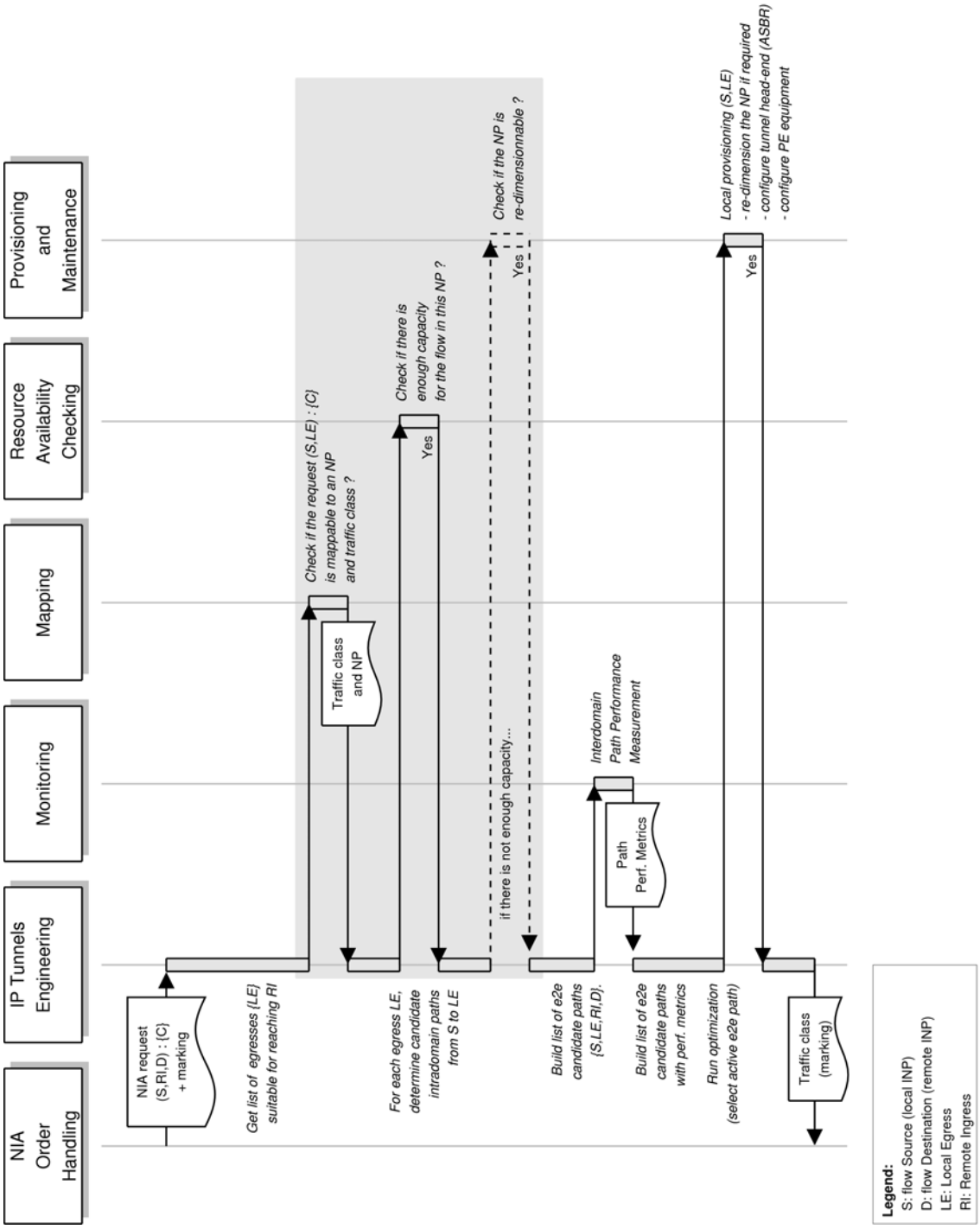


Figure 17 Flowchart of the handling of a NIA order for outgoing traffic.

4.1.4.4 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 4.1.4.4.1 and the inter-domain paths performance measurement in Section 4.1.4.4.2.

The second role of the Monitoring component is the measurement of the inter-domain traffic matrix. We describe this part in Section 4.1.4.4.3.

4.1.4.4.1 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 18 *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.

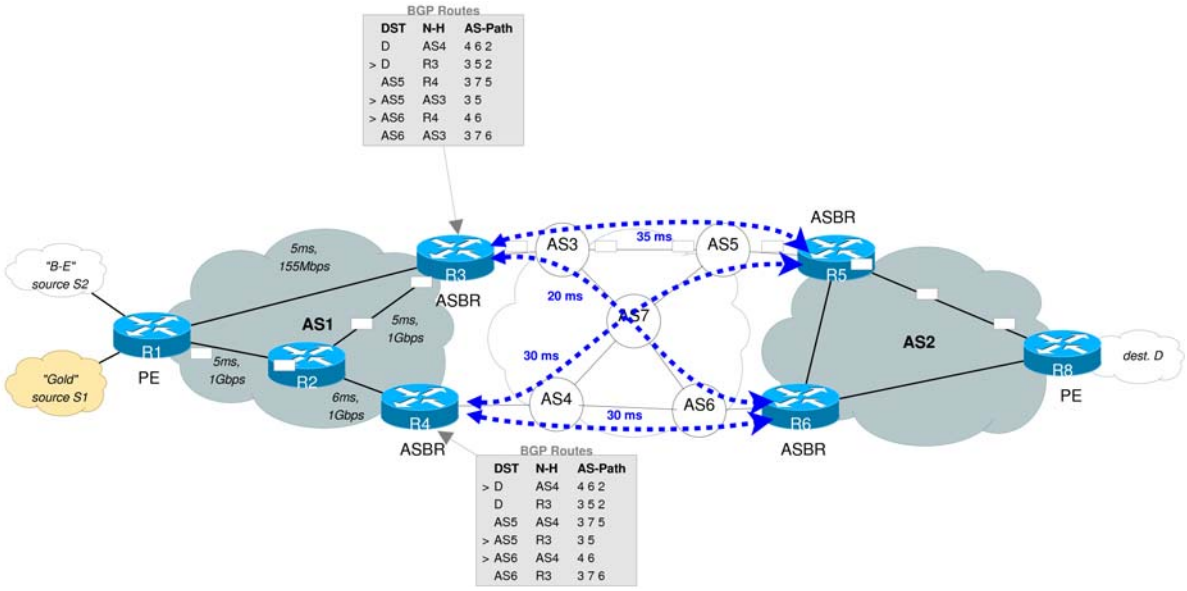


Figure 18 Inter-domain paths performance measurement.

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.

4.1.4.4.2 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 18, 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.

4.1.4.4.3 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.

4.1.4.5 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.

4.1.4.5.1 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 18, 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 18, 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 19 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. For the example shown in Figure

19, and assuming that the local INP has deployed M-ISIS with 2 virtual topologies, the list of candidate paths would be as shown in. Table 3.

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 3 List of candidate end-to-end paths.

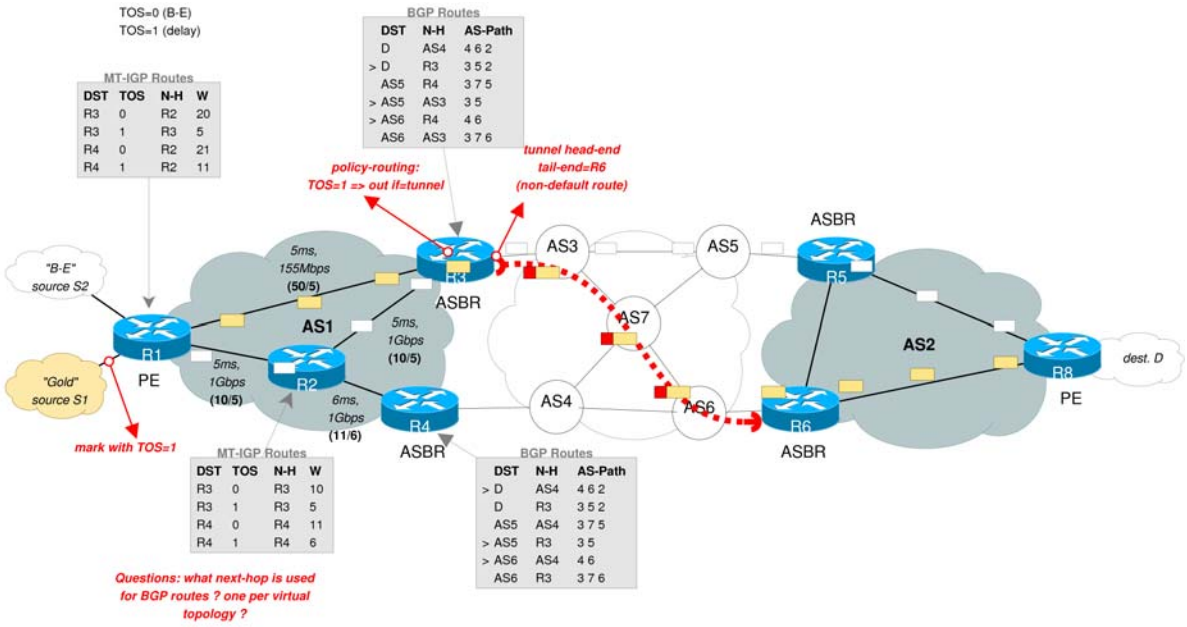


Figure 19 Example configuration if Network Planes are implemented using MTR.

4.1.4.5.2 Paths Selection

The Paths Selection algorithm is responsible for selecting the best paths among the candidate end-to-end paths obtained in Section 4.1.4.5.1. 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 Tunnelling Service Design and Control

In this section we describe the functional architecture of the IP Tunneling solution and its implementation. In particular, we first describe in Section 4.1.5.1 the IP Tunneling Service, which take care of encapsulating the traffic for inter-domain routing. Then, in Section 4.1.5.2 we describe how the paths selection is performed by using the Tunneling Service Controller.

4.1.5.1 Tunneling Service

In order to remain aligned with the ongoing work in the IETF and IRTF standardisation bodies, the TS consists on the IPv4 implementation of the LISP (Locator/ID Separation Protocol) approach [LISP06]. The LISP tunnelling Service is implemented in the kernel of the FreeBSD operating system. Figure 21 gives a snapshot of the logical architecture of the LISP TS. LISP protocol is based on concept of Locator and IDs and hence on a mapping function between these two type of entities. Border routers become the Routing LOCators (RLOCs), i.e. the tunnel ingress and egress, for all the local IP addresses, which are considered as identifiers. We can distinguish two type of mappings: *1)* local mappings and *2)* remote, temporarily stored, mappings. Local mappings consist in the association between local IPs (called EIDs -- End-host IDentifiers) and the local border routers (RLOCs). This binding is stored in the *Local Mapping Database* (*cf.* Figure 21) and changes only due to configuration or topology modification, thus not very often. Remote mappings concern the mapping between remote EIDs and theirs RLOCs. This information is stored in the *Mapping Cache*. This data structure (consisting in a radix tree) contains all the mappings necessary to correctly encapsulate packets of ongoing flows. This means that this data structure is populated in an on-demand fashion. The first packet of a flow for which no mapping exist in the cache will trigger a map lookup, whose result will be put in the cache. If the EID, for which a mapping has been requested, has more than one RLOC, the TSC is queried in order to give a priority to the RLOCs and to use the one that allows to increase performance. When an entry of the Mapping cache is not used anymore, i.e., no more flows need to be forwarded to a certain destination AS, the mapping is purged after a timeout.

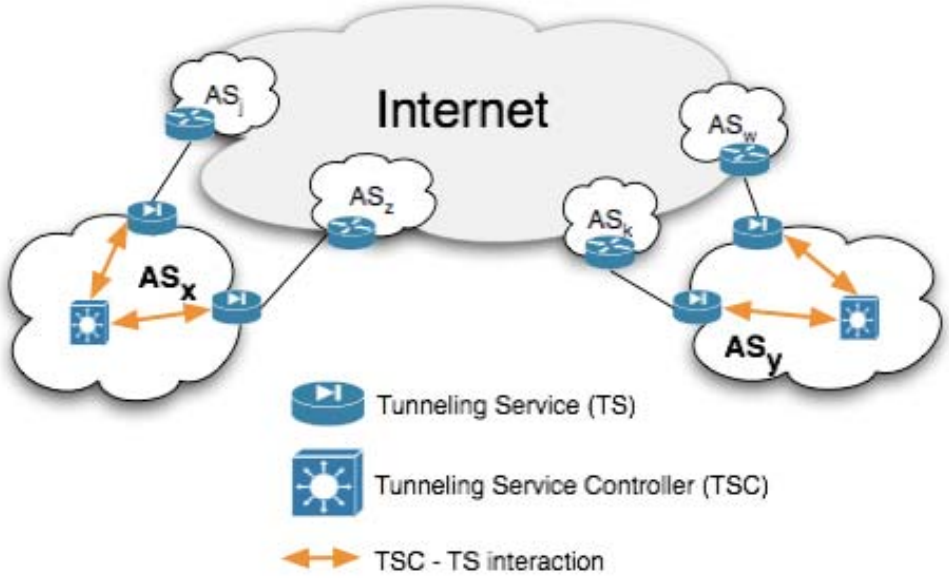


Figure 20 TS and TSC placement in the global Internet Architecture

For outgoing flows, the *Local Mapping Selector* decides whether packets need to be encapsulated or they should be forwarded in the normal way (without encapsulation). The decision is based on the content of the Local Mapping Database. If packets need to be encapsulated, they are delivered to the ITR (Ingress Tunnel Router) module. The ITR perform a simple encapsulation based on the content of both Local Mapping Database, in order to select the source RLOC, and on the content of the Mapping Cache, in order to select the best destination RLOC. Then the packet is injected in the Internet. Note that inside the Internet the packet is forwarded as a normal IP packet from source RLOC to destination RLOC.

For incoming flows, the *Egress Selector* simply selects packets that are destined to the router itself and checks whether it is a LISP encapsulated packet. If both conditions are true, this means that it is an incoming tunnelled packet and it needs to be treated by the ETR (Egress Tunnel Router) module. The ETR also checks for the mapping of the remote EID in the Mapping Cache. This operation is done for two reasons. First, if no mapping exist for the remote EID a lookup is triggered in order to retrieve the complete list of RLOCs. Note that by comparing the inner header with the outer header it is possible to rebuild a simple mapping binding the remote EID to the RLOC that tunnelled the packet. However, in order to exploit path diversity and improve performance there is the need to retrieve all of the RLOCs and query the TSC to obtain information on the best locator to use. Second, each tunnelled packet embeds in the LISP header RLOCs' reachability information, which are store in the Mapping Cache. Thus, upon reception of a tunnelled packet an update on locators' reachability information may be performed. Actually there is a third reason related to security issues. Indeed, in order to avoid spoofing, LISP relies on a simple nonce mechanism, thus returning nonce need to be checked.

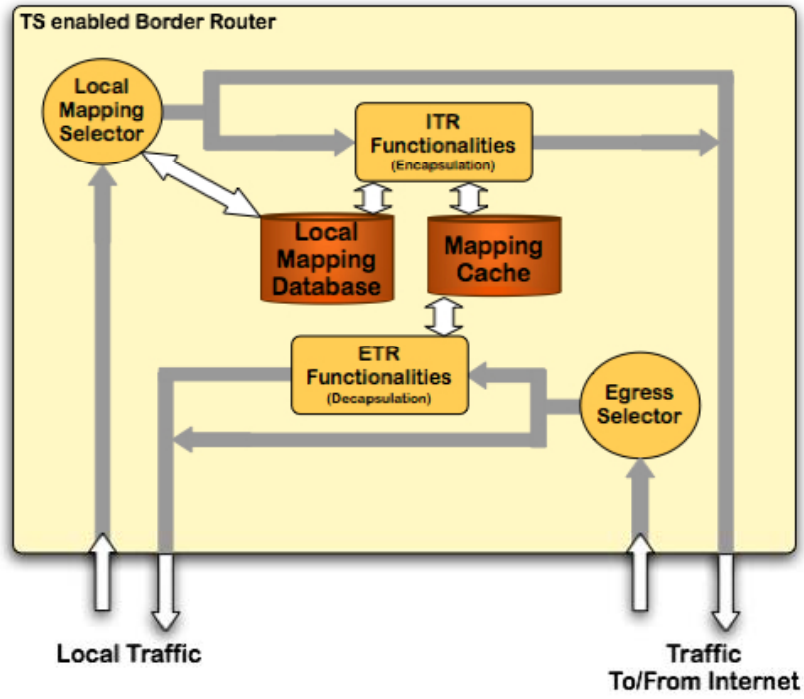


Figure 21 Logical Architecture of the Tunnelling Service

It is important to remark that RLOCs and EIDs are IP addresses. This allows to easily deploy the TS. Indeed, no changes are needed in the local AS nor in the core Internet. Tunnelling will be totally transparent for both sides. At the same time this ease the design of the TSC, since the characterisation of the performance of the paths to reach different locators simply means to characterise an IP path. Furthermore, considering the emulation of a multi-AS topology, as described in [D4.1], no changes need to be introduced on the emulation utility.

4.1.5.2 Tunnelling Service Controller Design

ISP-Driven Informed Path Selection (IDIPS) is the TSC (Tunnel Service Controller) proposed in the AGAVE project. It is implemented as a Demon running in the user-space of a FreeBSD system. The IDIPS concept comes from the following observations. First, the source and destination RLOCs partially define the path followed by messages traversing LISP tunnels. Second, the construction of the EID-to-RLOC database must be aware of this problem and a solution must be proposed to construct tunnels with good QoS properties. Unfortunately, for scalability reasons, LISP routers cannot know the quality of every path. On the other hand, for performance issues, the paths cannot be analyzed on demand. A better solution would to propose an independent service, IDIPS, which can identify the best paths based on the source and destination RLOCs and QoS requirements. To determine the best end-points of a tunnel, IDIPS constructs a list with all the possible combination of RLOCs and estimates the quality for all of them based on a local knowledge base. IDIPS then orders the paths according to a given QoS.

To estimate the quality of a tunnel (i.e., a path), the IDIPS server has one knowledge base that contains information about RLOCs (i.e., IP addresses) or cluster of RLOCs (i.e., IP prefixes). The quality of a tunnel is the combination of the QoS-related information of its ends.

IDIPS service is composed of clients and servers. LISP routers are IDIPS clients and IDIPS servers are traditionally hosted by specific devices but can be hosted by DNS servers or whatever. IDIPS server architecture is based on two main concepts: the knowledge base (KB) and the cost function (CF).

The knowledge base is a database collecting information about IP prefixes (a generalization of RLOCs). When the server needs information about a prefix, it makes a look-up in the KB and immediately obtains the result. No measurement or analysis must be done at that time. The KB is implemented with a Patricia tree, which offers efficient functions for prefixes look-up. Every prefix in the KB has some attributes associated to it. The attributes are normalized numerical values that represent a specific metric. Normalized metric follows the same principle as Local Prefs in BGP. The complexity of the metric is hidden behind a numerical value. This numerical value must summary a set of possible various information such that the classification of the prefix for this cost function only consists in sorting prefixes by numerical value of the cost. Because the value of the metrics can vary all the time, a dedicated module is charged to maintain information in the knowledge base. A solution that estimates the metric on client's demand would not be effective.

Metrics values are computed by cost functions. Cost functions are divided in two pieces. First, a routine is added in the knowledge base maintenance module. This routine is called to recompute the numerical value associated with the metric in the KB when required. Second, a fast routine is added in the decision-process. This part of the cost function gets a source and a destination prefix as input and returns the cost of the couple. The cost of the couple is a special combination of the metric associated with the source and the same metric associated with the destination. The combination of the two values must be as simple as possible (e.g., a sum) to reduce the response-time. The complexity of the cost function must be in the first routine.

When the server receives a request from a client, it extracts the source prefixes proposed by the client and the destination prefixes and creates all the possible couples. It then computes the cost for each couple and constructs a list of couples ordered by cost. The couple with the lowest cost is the more attractive.

To combine multiple cost functions, a weight is associated to each CF. The global cost of a pair is the weighted sum of its costs. The pair with the lowest cost is the most attractive. When a cost function is not applicable to at least one couple, this cost function must be removed from every prefix in the selection process. For example, if the cost function CF2 is not applicable to couple B but applicable to couples A and C. If the server must return the 3 couples A, B and C. The cost function CF2 cannot be used to order the couples.

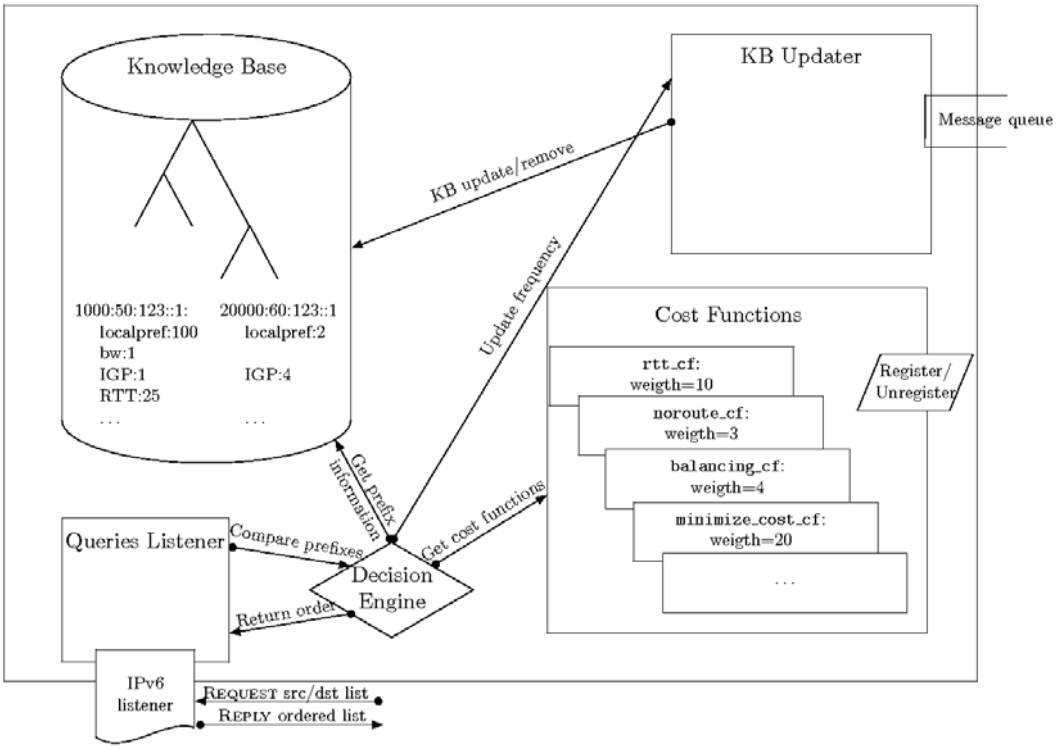


Figure 22 IDIPS Server Architecture

Figure 22 shows the global architecture of the IDIPS server. Requests from clients are received by the *Queries Listener* that receives messages from the clients and dispatches them to the *Decision Engine*. The Decision Engine is responsible for the ordering of the possible paths according to client requests. To do so, it determines the *Cost Functions* (CF) required for the ordering. The CF obtain information about the source and destination prefixes by querying the *Knowledge Base* (KB). The KB is maintained with external information by the *KB Updater*. The Queries Listener is also responsible for sending back the Idips replies to the clients.

The KB can be seen as a database storing various information about prefixes. Typically, every entry in the KB is a prefix with some attributes. The attributes are normalized numerical values representing specific metrics. Attributes follow the same principle as *Local Prefs* in BGP: the complexity of the metric is hidden behind a numerical value. However, unlike BGP Local Prefs, prefixes can have an arbitrary number of attributes associated to them, to allow the use of several unrelated metrics.

Storing prefixes instead of addresses in the KB provides more flexibility. Indeed, for most of the paths, a general information about the global performances of the network that owns the prefix is sufficient (e.g., is the network of the destination host reachable?). However, for some addresses, a specific information about it might be required (e.g., is the server reachable and what is the latency to it?). While treating with prefixes, this distinction is straightforward, the server works with general prefixes (e.g., **/48**) and, in some particular cases, the prefix is just equivalent to the address (e.g., a **/128** prefix).

The KB is divided in 3 parts. First, a Patricia tree, named *Responsibility Base* (RB), maps source prefixes to ISPs. Second, for each ISP, a *prefix information tree* (PIT) is maintained. Finally, a hashtable maintains a correspondence between ISPs and their prefixes information tree.

Prefixe information trees are Patricia trees where the nodes represent prefixes such that any child always refers to a more specific prefix of its parent. Every node points to its parent, its optional children and two hashtables. The hashtables are used to store the value of the attributes.

We define two types of attributes, each one being stored in its own hashtable. On one side, the *inheritable* attributes are those for which the value is the same for the prefix and all its sub-prefixes. On the other side, *uninheritable* attributes are those for which the value is not propagated to the sub-prefixes. When both uninheritable and inheritable attributes are defined, the first one is chosen. Attributes are divided into two categories, the *built-in* and the *custom* attributes. The built-in attributes have a specific meaning for the IDIPS server and are used internally to make special operations (e.g., disabling a particular prefix). On the contrary, custom attributes have no fixed meaning, their interpretation is left to the cost functions.

Attributes can represent any metric but they must always follow the *transitivity* principle: if A > B and B > C according to the attribute, then A > C for the same attribute. This property is required in order to compare the behavior of different prefixes.

More details about the IDPIS protocol can be found in [IDIPS00].

4.2 INP-level overlay routing (Inter-domain considerations)

In this section, we describe how to extend the INP-level overlay network into multiple Autonomous Systems (ASes) in order to support IP FRR across multiple domains. As both IGP and BGP routing are involved in this scenario the design of inter-domain overlay networks becomes much more difficult. Another obstacle to the construction of an efficient inter-domain overlay networks is that INPs normally do not disclose their network topologies and routing configurations (e.g., OSPF link weights and BGP route configurations) to each other, and also do not permit unauthorised installation of third-party facilities in their networks. As a result, it is difficult or even impossible for one single INP to deploy the overlay infrastructure across multiple domains owned by different INPs. In this

section, we first investigate the scenario that one single INP owns multiple ASes such that it is possible to deploy an inter-domain overlay network across multiple domains belonging to this INP. Thereafter we will discuss how multiple INPs can cooperate with each other to enable optimised overlay routing across each other's domains. Specifically, an approach based on path computation element (PCE [RFC4655]) will be introduced.

4.2.1 Scenario 1 – One INP owns multiple domains

In this section, we describe how a single INP performs overlay routing across multiple domains owned by itself. First of all, the INP needs to select overlay nodes within each of its local domains. For simplicity, we assume that these overly nodes resides on edge routers including PEs and ASBRs. As we mentioned before, the basic idea of applying overlay routing is to detour (if necessary) traffic delivery from conventional paths decided by IGP/BGP routing according to the QoS requirements. Towards this end, the design of inter-domain overlay should consider both IGP/BGP routing configurations. For scalability consideration, we adopt single hop overlay approach for routing on top of multiple domains.

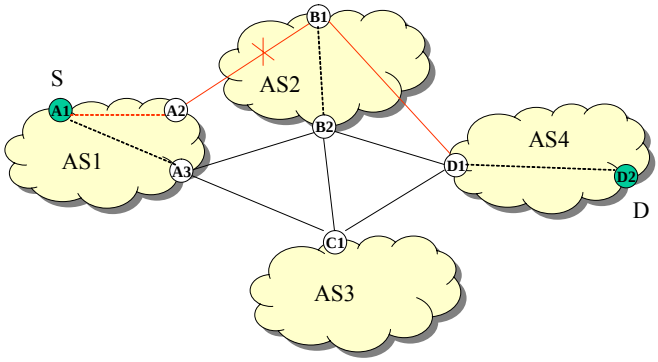
Figure 23 illustrates how a single INP should consider the selection of next hop overlay node for each PE pair so as to enable effective overlay paths selections. The task in this scenario is for each PE to find a proper ASBR as the intermediate overlay node towards the destination PE [1]. Now we consider the BGP configuration in AS1. As shown in table (a) in the figure, the BGP local preference (Local_pref) configuration indicates that the desired egress point towards AS4 is ASBR A2 (Local_pref = 100), which is also the desired egress towards AS2 (Local_pref = 100). In this case, it can be inferred that the default BGP path from the source PE A1 to the destination PE D2 is {A1, A2, B1, D1, D2}. If we assume the inter-domain link (A2, B1) is congested, one ASBR can be selected as the intermediate overlay node to bypass the hot spot for the customer's traffic from A1 to D2. In this example, potentially there are three possible candidates as the intermediate overlay nodes, namely A3, B2 and C1[2]. If we study the BGP configuration in AS1, we can find that only C1 can be selected for successful overlay node to avoid using the congested inter-domain link. The reason for this is as follows. If A3 is selected, then the traffic from A1 will first be tunnelled to this ASBR, but as A2 is the best egress points towards the final destination D2, A3 still needs to forward the data towards A2 first, which will use the congested link. If B2 is selected, traffic from A1 will first be tunnelled to B2 whose address belongs to AS2's prefix. According to AS1's BGP configuration, A2 is the best egress point towards AS2, which means that the congested inter-domain link (A2, B1) is still involved in the actual traffic delivery path from A1 to D2. Finally, if C1 is selected as the intermediate overlay node, the resulting overlay path is (A1, A3, C1, D1), and it successfully excludes the congested inter-domain link. It should be noted that the change of BGP routing configuration may affect the selection of overlay nodes. For example, if AS1's BGP routing is configured in table (b) in Figure 23, then B2 will become a feasible intermediate overlay node for bypassing the inter-domain link (A2, B1), because customer flows will be tunnelled to B2 via A3 (Local_pref = 100) rather than A2 (Local_pref = 50).

The example above illustrates the main idea of selecting ASBRs as potential overlay nodes between PEs. More specifically, the task is to identify between each pair of PE, based on the underlying BGP configuration, a set of potential overlay nodes for effectively detouring traffic from the default BGP paths. Once a proper set of overlay nodes is selected, QoS performance is monitored between the source PE and the corresponding sets of selected overlay nodes, as well as these overlay nodes and the destination PE. In order not to preclude the possibility of using default BGP paths, PE source and destination pairs may also perform end-to-end monitoring along these non-tunnelled paths. Similar to the intra-domain scenario, the monitored QoS performance is periodically propagated across the

[1] As ASBRs are normally not attached with end users, they are only served as intermediate overlay nodes for detouring traffic between PEs, but they are not considered as "source/destination" overlay nodes.

[2] We assume that the IP address to which the overlay path is tunnelled always belongs to the address prefix associated with its own AS. Put in other words, the addresses allocated to the "public" interface attached with inter-domain links are not used for tunnelling.

overlay network. However, in order to reduce the communication overhead, it is not necessarily to flood the QoS information by all ASBRs to all PEs. Instead, between each pair of PEs, only the selected ASBRs (as intermediate overlay nodes) should be responsible for disseminating the QoS information towards the source PE.



		ASBR	
		A2	*A3*
Destination AS number	*AS2*	100	50
	AS3	50	100
	AS4	100	50

(a) Local_pref configuration 1 (AS1)

		ASBR	
		A2	*A3*
Destination AS number	*AS2*	50	100
	AS3	50	100
	AS4	100	50

(b) Local_pref configuration 2 (AS1)

Figure 23 Inter-domain overlay construction within one INP

4.2.2 Scenario 2 – Domains belong to different INPs

In this section we briefly discuss a more general scenario where individual domains belong to different INPs. As we mentioned previously, the main difficulty in enabling inter-INP overlay routing is due to the limited knowledge about the global QoS performance, such that it is not possible for one single INP to construct inter-domain overlays across multiple INPs. In this case, the most appropriate solution is for individual INPs to collaborate with each other and jointly construct a fully distributed overlay system for dynamic inter-INP paths selections. In effect, the recently proposed Path Computation Element system has offered an ideal platform for this purpose. The idea of constructing overlay paths across ASBRs is actually similar to the computation of loose paths mentioned in [RFC4655]. More specifically, the PCE in individual domains work in a client/server fashion to compute inter-domain path hop by hop at the domain level. Instead of computing an explicit router level path across domains, loose path computation is normally only responsible for exploring one or more multiple ASBRs along the end-to-end path.

In Figure 24, we assume that the four ASes belong to different INPs. In order for these domains to jointly achieve overlay traffic delivery with end-to-end QoS requirements, dedicated PCEs are installed within individual ASes. The PCE inside each domain is able to gather the QoS information within its own AS by means of TE-extension of IGPs (e.g., OSPF-TE). For each PE which wants to establish an overlay path, it first needs to make a request for path computation towards its local PCE, which will then decides the best egress point of the source domain and then forwards the request towards a desired downstream PCE to compute the rest of the overlay path. In the figure, a possible scenario could be as follows. The PE router A1 needs to establish an overlay path to reach the PE router D2 in AS4. It first makes an overlay path computation request to PCE1 who is responsible for exploring the intra-domain segment of the overlay path (e.g., to select ASBR A2 as the overlay egress point), and then it forwards the path computation request to PCE2 in AS2 to compute the next loose hop (e.g. B2, due to an observed congestion between ASBRs B1 and B3). Thereafter, PCE2 may choose to contact PCE4 in the destination AS, which decides that the loose intra-domain path (D1, D2) should constitute the overlay path towards the destination PE. As a result, the overall overlay path is {A1, A2, B2, D1, D2}. From this example it can be noticed that the PCE based overlay path

computation is not necessarily decided by a single hop overlay node as it was described in the last section.

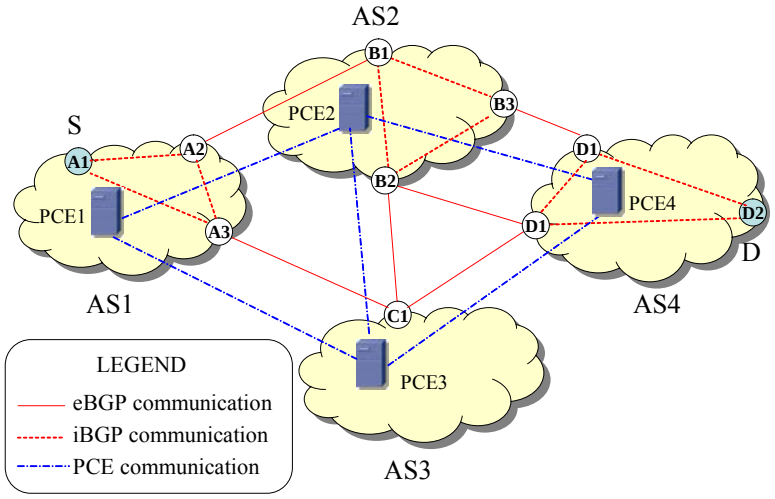


Figure 24 PCE based Inter-domain overlay

4.3 q-BGP enhancement

One of the methods to maintain and distribute routing information in Parallel Internets in AGAVE is the use of a QoS-enhanced version of BGP, called q-BGP. This section describes the adaptation of q-BGP for use in AGAVE and the 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 to which Parallel Internet it belongs.
- A QoS_NLRI attribute which expresses which Parallel Internet this message belongs to and the optional fields which describe the QoS attributes of the path expressed in the message.

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

The following enhancements were investigated as part of the q-BGP work in AGAVE:

- 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.
- Parallel Internet optimisation: How a single plane is optimised globally based on local domain decisions and how multiple planes may interact assuming hard and softer partitioning.

4.3.1 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.

4.3.1.1 Primitive types

- **Average one-way delay**: The average delay a packet can expect when following this path.
- **Minimum one-way delay**: The minimum delay a packet can expect when following this path.
- **Available bandwidth**: 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**: The expected rate of packet loss that can be expected when following this path.

4.3.1.2 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.

4.3.1.2.1 Jitter, or Inter-packet delay variation

Jitter is an important attribute 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.

4.3.1.2.2 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:

4.3.1.2.3 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.

4.3.1.2.4 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.

4.3.2 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:

4.3.2.1 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].

4.3.2.2 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.

4.3.2.3 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 is a significant part of the work into q-BGP as it is required to avoid "QA rush" and to optimise Parallel Internet usage.

4.3.3 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.

4.3.3.1 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].

4.3.3.2 Alternative route selection processes

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

4.3.3.2.1 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:

$$\frac{\sum_{n} \alpha_n \frac{QAn_{incoming}}{QAn_{typical}}}{n}$$

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.

4.3.3.2.2 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.

4.3.3.3 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. These techniques are critical when dynamically monitored QAs are being used in advertisements.

4.3.3.4 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.

4.3.4 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.

4.3.4.1 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 and Parallel Internets.

4.3.5 q-BGP and the co-existence of planes

Previously all investigations into q-BGP were made with a single plane with the assumption 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.

4.4 BGP planned maintenance

4.4.1 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 ([AHMA06]). 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 is 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.

4.4.2 BGP graceful shutdown for planned maintenance

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.

4.4.3 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):

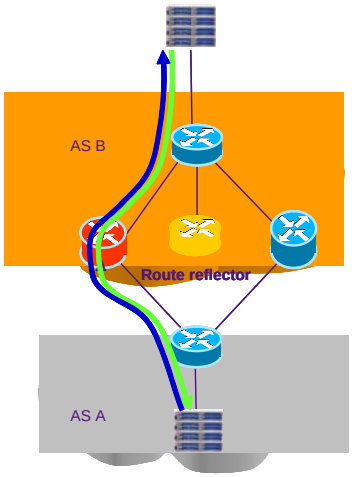


Figure 25 Topology creating LoC during BGP PM

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):

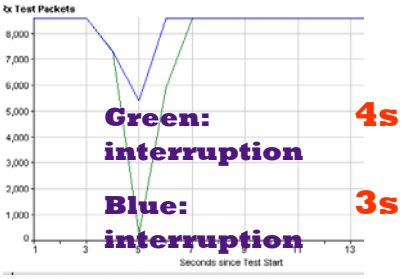
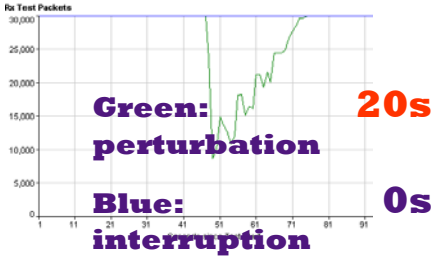
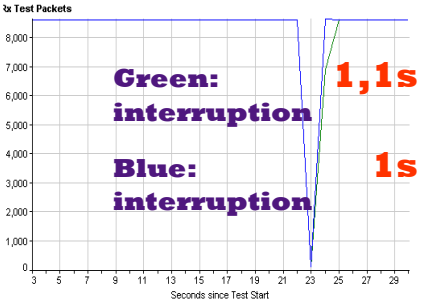
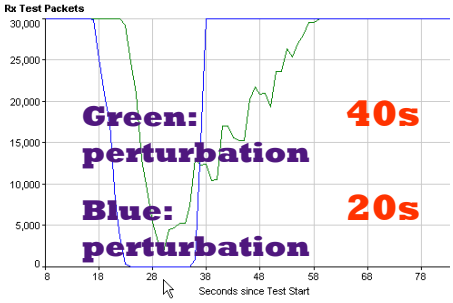
	Case 1: 100 BGP routes need to be rerouted.	Case 2: 100 000 BGP routes need to be rerouted.
Reload of C12C (shut the whole BGP router)		
Clear IP BGP C12C (shut one eBGP session)		

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 does not explicitly reload its line card. With the recent implementations of the NSF (Non Stop Forwarding) and GR (Graceful 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 does not have an alternate route. When the nominal path is shutdown, the ASBR starts dropping packets and advertise the failures to its neighbours. The peers try to find an alternate route but this may requires some additional BGP message exchange.

This behaviour 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 signalling and routing issue. Some routing or signalling protocols are already addressing it, for example MPLS-TE in

[VASS01], GMPLS in [ALI06], IS-IS TE in [VASS02], link state IGP (OSPF or IS-IS) in [FRAN05] and [FRAN06]…

4.4.4 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 bring 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, labelled, 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.

4.4.4.1 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:

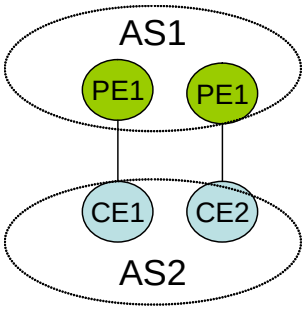


Figure 27 eBGP topology 2PE-2CE

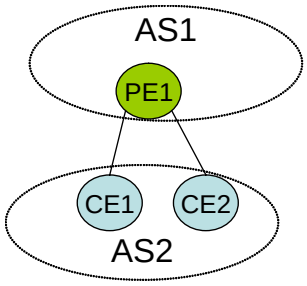


Figure 28 eBGP topology PE-2CE

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

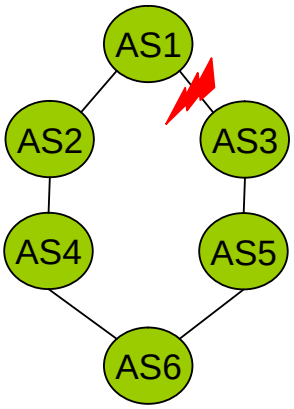


Figure 29 eBGP Internet wide topology

4.4.4.2 iBGP topologies

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

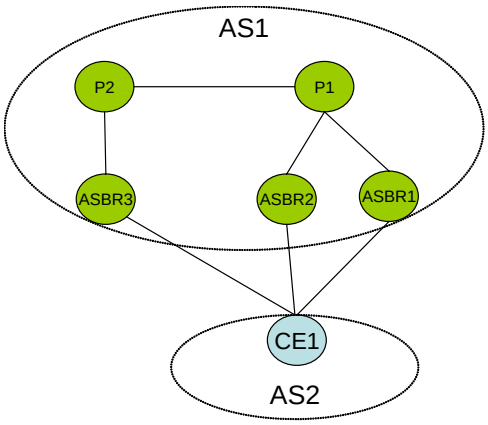


Figure 30 iBGP full mesh topology

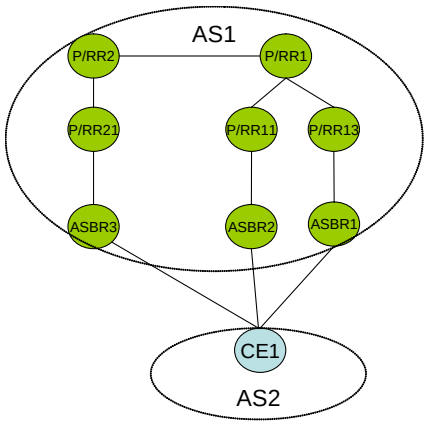


Figure 32 iBGP hierarchical RR topology

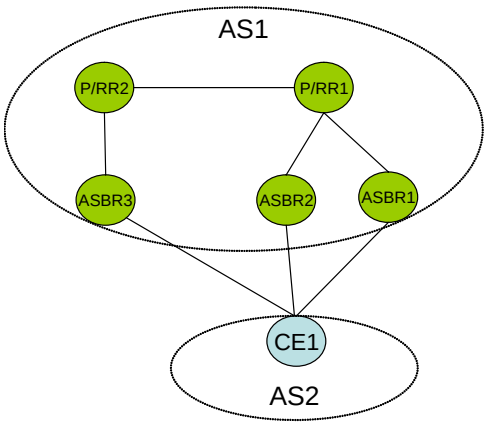


Figure 31 iBGP RR topology

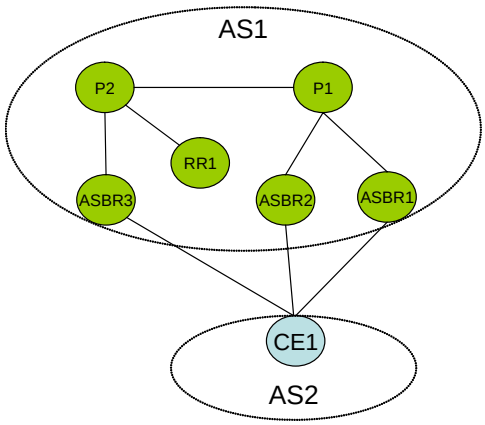


Figure 33 iBGP centralized RR topology

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 (e.g. full mesh implies a full mesh of iBGP sessions between

all routers of the AS) and the name of the router (e.g. RR are Route Reflectors which centralizes iBGP sessions).

4.4.5 Solution

Given the goal to have a lightweight solution which could be deployed rapidly by ISPs, we choose to restrict the solution space in order to avoid protocol extension. We choose, as much a possible, a solution relying on operation procedures which can be performed by ISPs. However, the full coverage of the requirements do requires some protocol and implementation enhancement in order to have no loss of connectivity in all possible BGP topologies.

The procedures described can be applied to avoid packet loss for outbound and inbound traffic flows initially forwarded along the peering link to be shut down. These procedures allow routers to keep using old paths until alternate ones are learned, ensuring that routers always have a valid route available during the convergence process.

4.4.5.1 Terminology

We will use the following terminology:

- g-shut initiator: a router on which the session shutdown is performed for the maintenance.
- g-shut neighbour: a router that peers with the g-shut initiator via (one of) the maintained session(s).
- Initiator AS: The Autonomous System of the g-shut initiator.
- Neighbour AS: The Autonomous System of the g-shut neighbour.
- Affected prefix : a prefix initially reached via an eBGP peering link undergoing the maintenance, or learned via an iBGP peering undergoing the maintenance.
- Affected router: a router having an affected prefix.
- Nominal / old / pre-convergence path: a BGP path via the peering link(s) undergoing the maintenance. This path will no longer exist after the shutdown.
- Backup / new / post-convergence path: A path toward an affected prefix that will be selected as the best path by an affected router for that prefix, when the link is shut down and the BGP convergence is completed.
- Transient alternate path: A path towards an affected prefix that may be transiently selected as best by an affected router during the convergence process but that is not a post-convergence path.
- Loss of Connectivity (LoC): The occurrence of a state of a router such that an affected prefix is unreachable.

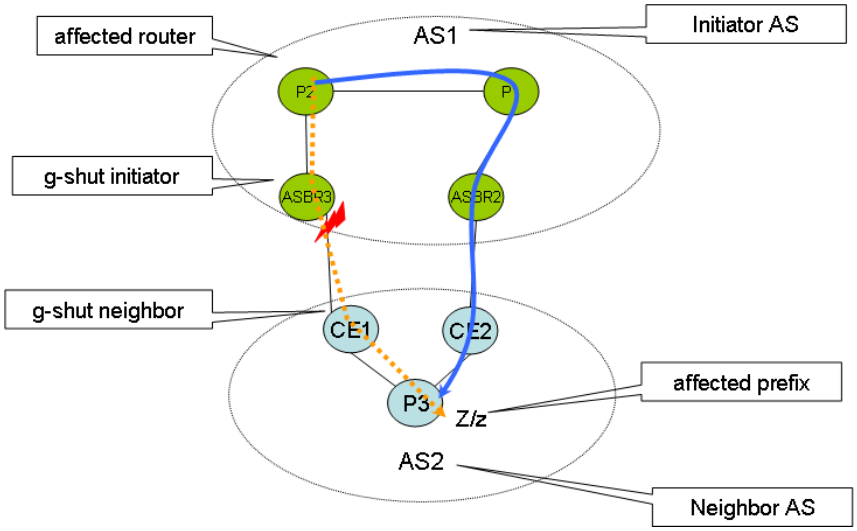


Figure 34 BGP g-shut terminology

4.4.5.2 Packet loss upon manual eBGP session shutdown

Packets can be lost during a manual shutdown of an eBGP session for two reasons.

First, routers involved in the convergence process can transiently lack of paths towards an affected prefix, and drop traffic destined to this prefix. This is because alternate paths can be hidden by nodes of an AS. This happens when the paths are not selected as the best ones by the ASBR that receive them on an eBGP session, or by Route Reflectors that do not propagate them further in the iBGP topology because they do not select them as the best ones.

Second, within the AS, routers' FIB can be transiently inconsistent during the BGP convergence and packets towards affected prefixes can loop and be dropped. Note that these loops only happen when BR-to-BR encapsulation is not used within the AS.

4.4.5.3 Solutions to avoid packet losses

This section describes means for an ISP to reduce the transient loss of packets upon a manual shutdown of a BGP session. The first solution is to improve the availability of the alternate paths on all routers in all times and conditions. This is also referred as increasing route diversity. The second solution is to keep the current route diversity but to search for the backup path when needed by the planned maintenance operation.

4.4.5.3.1 Improving availability of alternate paths

Using advertise-best-external on ASBRs improves the availability of alternate paths in route reflectors upon a convergence. Hence it reduces the LoC duration for the outbound traffic of the ISP upon an eBGP Session shutdown by reducing the iBGP path hunting.

All solutions that improve the availability of paths beyond the Route Reflectors barrier also help in reducing the LoC. These have been discussed for years but are not yet available, probably due to their implementation complexity.

Also, the LoC for the inbound traffic induced by a lack of alternate path propagation within the iBGP topology of a neighbouring AS is not under the control of the operator performing the maintenance.

4.4.5.3.2 Graceful shutdown procedures for eBGP sessions

In this section, we describe a procedure to apply to reduce the LoC with readily available BGP technologies, and without assuming particular iBGP design for the AS performing the maintenance and in the neighbouring ASes.

4.4.5.3.2.1 Outbound traffic

The goal is to render the affected primary paths less desirable by the BGP decision process of affected routers, still allowing the old paths to be used during the convergence while alternate paths are propagated to the affected routers.

A decrease of the Local-Pref value of the affected paths can be issued in order to render the affected paths less preferable, at the highest possible level of the BGP Decision Process.

This operation can be performed by reconfiguring the out-filters associated with the iBGP sessions established by the g-shut initiator.

The modification of the filters must supplant any other rule affecting the local-pref value of the old paths.

The modification of the out-filters will not let the g-shut initiator switch to another path, as the input of the BGP decision process of that router does not change.

As a consequence, the g-shut initiator will not send a withdraw message over its iBGP sessions. It will however modify the Local-Pref of the affected paths so that upstream routers will switch to alternate ones.

When the actual shutdown of the session is performed, the g-shut initiator will itself switch to the alternate paths.

4.4.5.3.2.2 Inbound traffic

The solution described for the outbound traffic can be applied at the neighbour AS. This can be done either "manually" or by using a community value dedicated to this task.

4.4.5.3.2.2.1 Phone call

The operator performing the maintenance of the eBGP session can contact the operator at the other side of the peering link, and let him apply the procedure described above for its own outbound traffic.

4.4.5.3.2.2.2 Community tagging

A community value (referred to as GSHUT community in this document) can be agreed upon by neighbouring ASes. A path tagged with this community must be considered as soon to be affected by a maintenance operation.

4.4.5.3.2.2.2.1 Configuration

A g-shut neighbour is pre-configured to set a low local-pref value for the paths received over eBGP sessions which are tagged with the GSHUT community.

This rule must supplant any other rule affecting the local-pref value of the paths.

This local-pref reconfiguration should be performed at the out-filters of the iBGP sessions of the g-shut neighbour. That is, the g-shut neighbour does not directly withdraw or select alternate paths upon the reception of paths tagged with this community. However, it will propagate updates of these paths, by lowering their local-pref values.

4.4.5.3.2.2.2.2 Operational behaviour

Upon the manual shutdown, the output filter associated with the maintained eBGP session will be modified on the g-shut initiator so as to tag all the paths advertised over the session with the GSHUT community.

4.4.5.3.2.2.2.3 Transitivity of the community

If the GSHUT community is an extended community, it should be set non transitive.

If a normal community is used, this community should be removed from the path by the ASBR of the peer receiving it. If not, the GSHUT community may be removed from the path by the ASBR of the peer, before propagating the path to other peers.

Not propagating the community further in the Internet reduces the amount of BGP churn and avoids rerouting in distant ASes that would also recognize this community value. In other words, it helps concealing the convergence at the maintenance location.

There are cases where an inter-domain exploration is to be performed to recover the reachability, e.g., in the case of a shutdown in confederations where the alternate paths will be found in another AS of the confederation. In such scenarios, the community value should be allowed to transit through the confederation but may be removed from the paths advertised outside of the confederation.

When the Local-pref value of a path is conserved upon its propagation from one AS of the confederation to the other, there is no need to have the GSHUT community be propagated throughout that confederation.

4.4.5.3.2.2.2.4 Easing the configuration for G-SHUT

From a configuration burden viewpoint, it would be much easier to have the GSHUT community value be standardized.

First, an operator would have a single configuration rule to be applied at the maintenance time, which would not depend on the identity of its peer. This would make the maintenance operations less error prone.

Second, a single in-filter related to g-shut could be configured for all BGP sessions, at the g-shut neighbour.

4.4.5.3.3 Graceful shutdown procedures for iBGP sessions

If the iBGP topology is viable after the maintenance of the session, i.e, if all BGP speakers of the AS have a path towards all affected prefixes after the convergence, then a shutdown of an iBGP session does not lead to transient unreachability.

However, in the case of a shutdown of a router, a reconfiguration of the out-filters of the g-shut initiator should be performed to set a low local-pref value for the paths originating from other protocols which are redistributed in BGP by the g-shut initiator.

This behaviour is equivalent to the recommended behaviour for paths "redistributed" from eBGP sessions to iBGP sessions in the case of the shutdown of an ASBR.

4.4.5.4 Forwarding modes and forwarding loops

If the AS applying the solution does not rely on encapsulation to forward packets from the Ingress Border Router to the Egress Border Router, then transient forwarding loops and consequent packet losses can occur during the convergence process, even if the procedure described above is applied.

Using out-filter as a first step avoids the forwarding loops between the g-shut initiator and the upstream routers. Indeed, when this first step is applied, the g-shut initiator keeps using its own external path and lets the upstream routers converge to the alternate ones. During this phase, no forwarding loops can occur between the g-shut initiator and its upstream routers. When the first step is

finished, all the upstream routers have switched to alternate paths and the transition performed by the g-shut initiator will be loop-free. Transient forwarding loops between other routers will not be avoided with this procedure.

4.4.5.5 Dealing with Internet policies

A side gain of the maintenance solution is that it can be used to reduce the churn implied by a shutdown of an eBGP session.

For this, it is recommended to apply the filters modifying the local-pref value of the paths to values strictly lower than, but as close as possible to, the Local-pref values of the post-convergence paths.

For example, if a peering link is shut down between a provider and one of its customers, and another peering link with this customer remains active, then the value of the local-pref of the old paths should be decreased to the smallest possible value of the 'customer' local_pref range, minus 1. Thus, routers will not transiently switch to paths received from shared-cost peers or providers, which could lead to the sending of withdraw messages over eBGP sessions with shared-cost peers and providers.

Proceeding like this reduces both BGP churn and traffic shifting as routers will less likely switch to transient paths.

In our example, transient unreachabilities in the neighbouring AS that are due to the sending of "abrupt" withdraw messages to shared-cost peers and providers are also prevented.

4.4.5.6 Effect of the g-shut procedure on the convergence

This section describes the effect of applying the solution.

4.4.5.6.1 Maintenance of an eBGP session

This section describes the effect of applying the solution for the shutdown of an eBGP session.

4.4.5.6.1.1 Propagation on the other eBGP sessions of the g-shut initiator

Nothing is propagated on the other eBGP sessions when the out-filters reconfiguration step is applied. The reconfiguration is indeed only defined for its iBGP sessions.

The reconfiguration of the iBGP out-filters will trigger the reception of alternate paths at the g-shut initiator. As the eBGP in-filters have not been modified at that step, the old paths are still preferred by the g-shut initiator.

4.4.5.6.1.2 Propagation on the other iBGP sessions of the g-shut initiator

During the out-filter reconfiguration, path updates are propagated with a reduced local-pref value for the impacted paths. As a consequence, Route Reflectors and distant ASBRs select and propagate alternate paths through the iBGP topology as they no longer select the old paths as best.

When the shut-down is performed, the g-shut initiator propagates the alternate paths that it received on eBGP sessions to its iBGP sessions. Also, it withdraws on its iBGP sessions the paths for which the best alternative was received over its iBGP sessions.

4.4.5.6.1.3 Propagation of updates in an iBGP full-mesh

No transient LoC can occur if a reconfiguration of the iBGP out-filters on the g-shut initiator is performed.

4.4.5.6.1.4 Propagation of updates from iBGP to iBGP in a RR hierarchy

Upon the reception of the update of an old path with a lower local-pref value, Route Reflectors will either propagate the update, or select an alternate path and propagate it within their RR iBGP full-

mesh, to their own Route Reflectors in the case of a multiple level Route Reflector hierarchy, and to their clients.

During the convergence process controlled with the described procedure, some corner case timings can trigger transient unreachabilities.

A typical example for such transient unreachability for a given prefix is the following:

1. A Route Reflector RR1 only knew about the primary path upon the shutdown.
2. A member of its RR full-mesh RR2, propagates an update of the old path with a lower local-pref.
3. Another member RR3 processes the update, selects an alternate path, and propagates an update in the mesh.
4. RR2 receives the alternate path, selects it as best, and hence withdraws the updated old path on the iBGP session of the mesh.
5. If for any reason, RR1 receives and processes the withdrawn path generated in step 4 before processing the update generated in step 3, RR1 transiently suffers from unreachability for the affected prefix.

In such corner cases, the solution improves the iBGP convergence behaviour/LoC but does not ensure 0 packet loss, as we cannot define a simple solution relying only on a reconfiguration of the filters of the g-shut initiator.

The root cause is that even in the iBGP topology, a BGP update can be translated into a withdrawn. And we have seen above that in some corner cases, this withdrawal can use a different iBGP signalling path than the update and eventually could propagate faster.

The solution would be to not translate updates into withdrawals within the AS. An existing solution on ASBR is to use the "BGP external best" trick which allows an ASBR to advertise through iBGP its best *external* route even if it is not the best route that it is using (i.e. the ASBR uses a path learnt by iBGP). This solution is "out of scope" of the BGP specification but is implemented by some router vendors. For the purpose of BGP g-shut, we extend this trick for the Route Reflector by defining the "BGP cluster best" behaviour. With such behaviour, the route reflector advertises the best route learnt over its iBGP client session, even if it is not the best route that it is using (i.e. the ASBR uses a path learnt by iBGP).

4.4.5.6.2 Maintenance of an iBGP session

If the shutdown does not temper with the correctness of the iBGP topology, the described procedure is sufficient to avoid LoC.

4.4.5.6.3 Applicability of the g-shut procedures

The applicability of the g-shut procedure described in section 4.4.5.3.2 "Graceful shutdown procedures for eBGP sessions" to the cases presented in section 4.4.4 "Requirements for the BGP solution" can be shown by combining the effects described in this section.

4.4.5.6.4 In-filter reconfiguration

An In-filter reconfiguration on the eBGP session undergoing the maintenance could be performed instead of out-filter reconfigurations on the iBGP sessions of the g-shut initiator.

Upon the application of the maintenance procedure, if the g-shut initiator has an alternate path in its Adj-Rib-In, it will switch to it directly.

If this new path was advertised by an eBGP neighbour of the g-shut initiator, the g-shut initiator will send a BGP Path Update message over its iBGP and eBGP sessions.

If this new path was received over an iBGP session, the g-shut initiator will select that path and directly propagate a withdraw message over the iBGP sessions for which it is not acting as a Route Reflector. There can be iBGP topologies where the iBGP peers of the g-shut initiator do not know about an alternate path, and hence may drop traffic.

Also, applying an In-filter reconfiguration on the eBGP session undergoing the maintenance may lead to transient LoC in full-mesh iBGP topologies if:

a) An ASBR of the initiator AS, ASBR1 did not initially select its own external path as best, and

b) An ASBR of the initiator AS, ASBR2 propagates an Update message along its iBGP sessions upon the reception of ASBR1's update following the in-filter reconfiguration on the g-shut initiator, and

c) ASBR1 receives the update message, runs its Decision Process and hence propagates a withdraw of its external path after having selected ASBR2's path as best, and

d) An impacted router of the AS processes the withdrawal of ASBR1 before processing the update from ASBR2.

Applying a reconfiguration of the out-filters prevents such transient unreachabilities.

Indeed, when the g-shut initiator propagates an update of the old path first, the sending of the withdrawal by ASBR2 does not trigger unreachability in other nodes as the old path is still available. Indeed, even though it receives alternate paths, the g-shut initiator keeps using its old path as best as the in-filter of the maintained eBGP session has not been modified yet.

Applying the out-filter reconfiguration also prevents packet loops between the g-shut initiator and its direct neighbours when encapsulation is not used between the ASBRs of the AS.

4.4.5.6.5 Multi Exit Discriminator tweaking

The MED attribute of the paths to be avoided can be increased so as to force the routers in the neighbouring AS to select other paths.

The solution only works if the alternate paths are as good as the initial ones with respect to the Local-Pref value and the AS Path Length value. In the other cases, increasing the MED value will not have an impact on the decision process of the routers in the neighbouring AS.

4.4.5.6.6 IGP distance poisoning

The distance to the BGP next-hop corresponding to the maintained session can be increased in the IGP so that the old paths will be less preferred during the application of the IGP distance tie-break rule. However, this solution only works for the paths whose alternates are as good as the old paths with respect to their Local-Pref value, their AS Path length, and their MED value.

Also, this poisoning cannot be applied when next-hop self is used as there is no next-hop specific to the maintained session to poison in the IGP.

4.4.5.7 Security considerations

By providing the g-shut service to a neighbouring AS, an ISP provides means to this neighbour to lower the local-pref value of the paths received from this neighbour on their peering links.

The neighbour could abuse the technique and do inbound traffic engineering by declaring some prefixes as undergoing a maintenance so as to switch traffic to another peering link.

If this behaviour is not tolerated by the ISP, it should monitor the use of the g-shut community by this neighbour.

4.5 ASBR protection with RSVP-TE egress fast reroute

4.5.1 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 avoids 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 4.5.2. Section 4.5.3 defines the RSVP-TE *Egress FRR* mechanism. Finally section 4.5.4 defines extended BGP next-hop resolution procedures so as to support ASBR protection with RSVP-TE *Egress FRR*.

4.5.2 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 learnt 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 Base, 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 35 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.

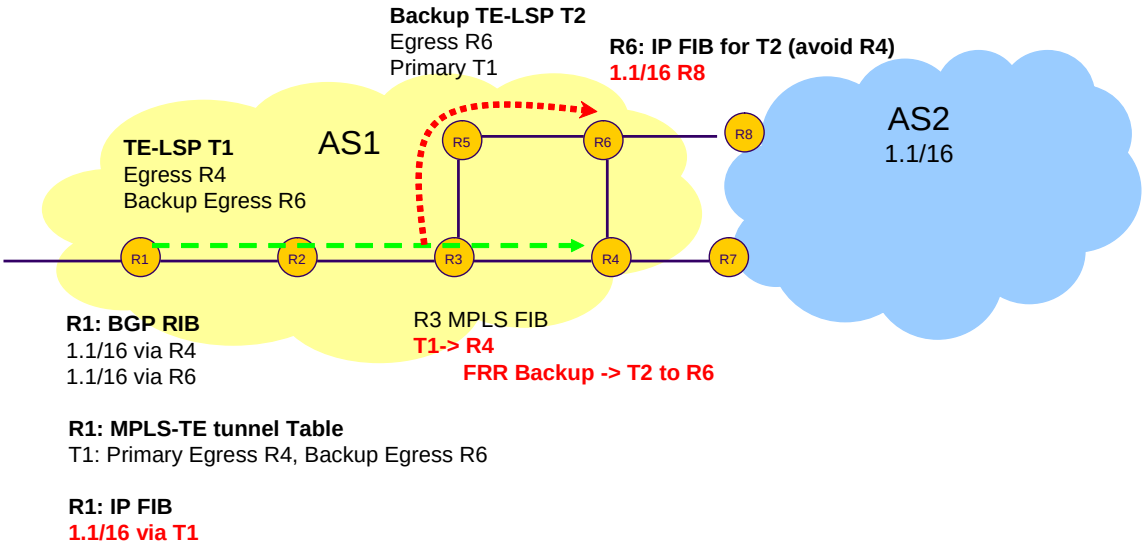


Figure 35 Egress ASBR protection with RSVP-TE Egress FRR

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 36 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.

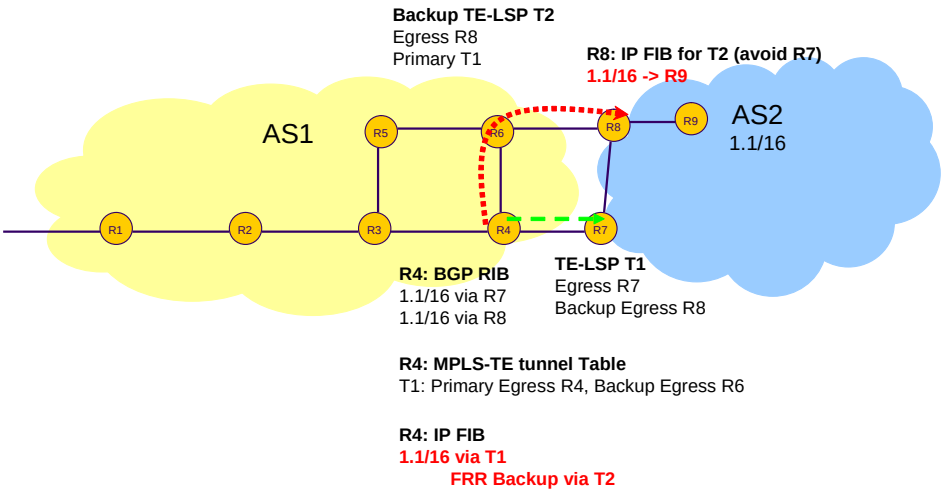


Figure 36 Ingress ASBR protection with RSVP-TE Egress FRR

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 37 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.

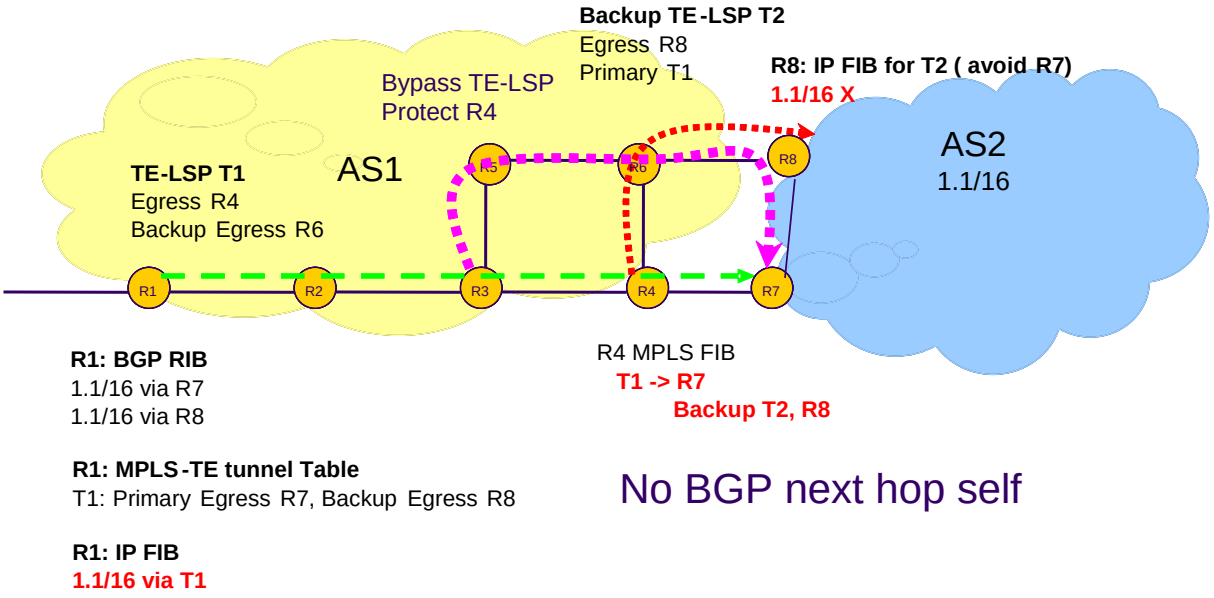


Figure 37 Egress ASBR, Ingress ASBR and inter-AS link protection with RSVP-TE Egress FRR

4.5.3 RSVP-TE Egress Fast Reroute

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

4.5.3.1 Egress FRR Terminology

The *Egress FRR* system described in Figure 38, 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.

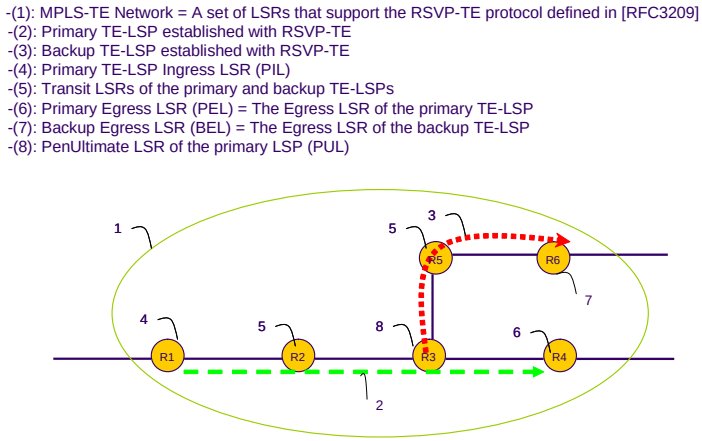


Figure 38 Egress FRR System

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

4.5.3.2 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 4.5.3.2.1 describes the required information, and section 4.5.3.2.2 proposes, for the sake of illustration, a way to encode this information in RSVP-TE.

4.5.3.2.1 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 an *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".

4.5.3.2.2 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).

4.5.3.3 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.

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

4.5.3.3.1.1 Procedure before failure

During the establishment of the primary TE LSP, the Ingress LSR includes in the RSVP-TE *Path* message, in addition to the 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 4.5.4).

An LSP protected with *Egress FRR* can be used for static routing, IGP routing (autoroute announce) or BGP routing (see also section 4.5.4).

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.

4.5.3.3.1.2 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.

4.5.3.3.1.3 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.

4.5.3.3.1.4 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 will 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 deleting the old LSP, if it is still alive.

4.5.3.3.2 Procedures on the Penultimate LSR (PUL)

4.5.3.3.2.1 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 (i.e. Primary Egress LSR is up) the backup output is not active.

4.5.3.3.2.2 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.

4.5.3.3.2.3 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.

4.5.3.3.2.4 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.

4.5.3.3.3 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 is received on this LSP this means that the primary Egress LSR has failed and that it must not forward 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 is 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.

4.5.3.4 Make before break procedure

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

RSVP-TE LSP re-optimization 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 redirects traffic on this new LSP and the old LSP is finally turned down. This allows tunnel re-optimization with minimum impact on the traffic. RSVP-TE make before break procedures are detailed in [RFC3209].

Practically, re-optimization 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 re-optimization). The frequency actually depends on the frequency of the above events. Note that the re-optimization can be event driven or timer driven. The timer driven approach is recommended for stability reasons.

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

Upon primary LSP re-optimization, 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.

4.5.3.5 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 a case, the reserved bandwidth is not the sum but the maximum of the two LSP bandwidths.

4.5.3.6 Example

Figure 39, Figure 40, Figure 41, and Figure 42 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 39).

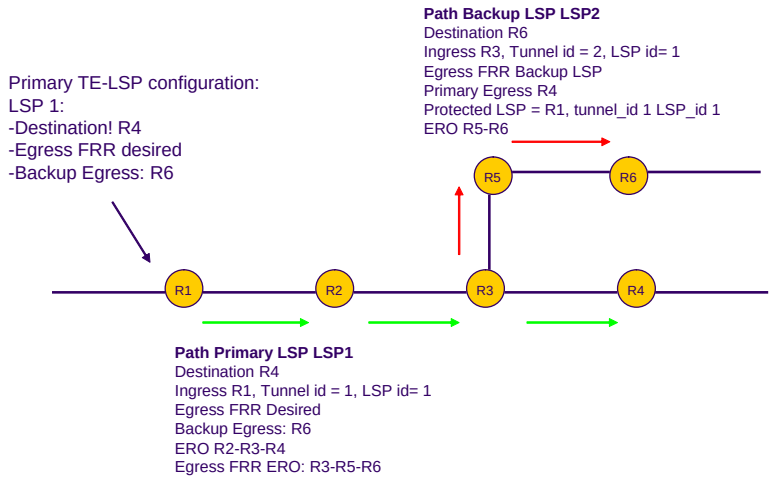


Figure 39 Signalling of primary and backup LSPs: Path message

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 40 and Figure 41, "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.

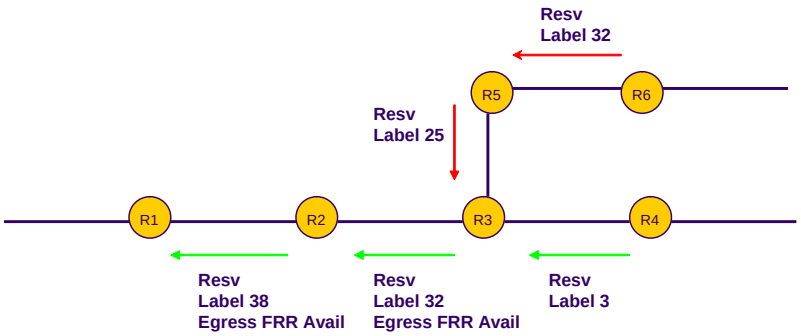


Figure 40 Signalling of primary and backup LSPs: Resv message

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 41 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.

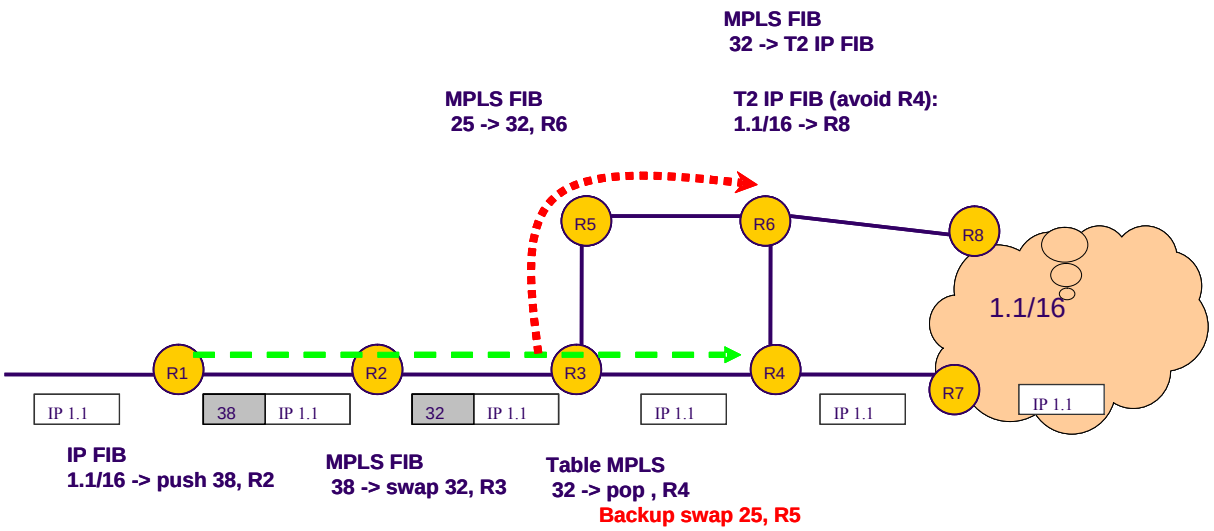


Figure 41 Packet forwarding before failure

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 42 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.

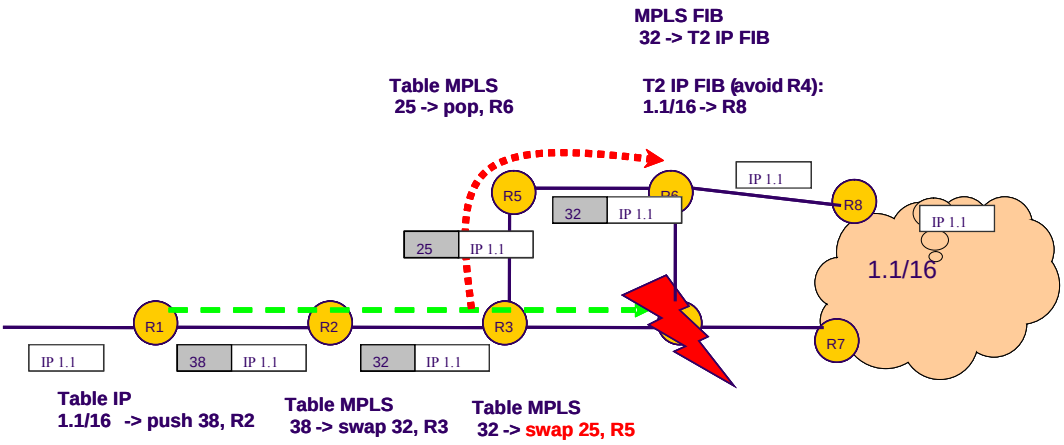


Figure 42 Packet forwarding during failure

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

4.5.4 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 are 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).

4.5.5 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 require 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 LSR 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.

4.6 Robust Egress point selection

4.6.1 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).

4.6.2 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 4.7 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).

4.6.3 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^l_{inter}	Capacity of the egress point l
x^l_{sk}	A binary variable indicating whether prefix k is assigned to the egress point l in state s
u^l_s	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^{FS}_{Ave}	Average of maximum egress point utilization across all failure states

Table 4 Notation used for the robust egress point selection problem

In this section, we present our robust egress point selection optimization problem formulation. Table 4 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:

$$\forall s \in S : Minimize\, U_{\max}(s) = Minimize \underset{\forall l \in L/\{s\}}{Max} (u^l_s) = Minimize \underset{\forall l \in L/\{s\}}{Max} \left(\frac{\sum_{k\in K}\sum_{i\in I} x^l_{sk} t(k,i)}{c^l_{\text{inter}}}\right) \quad (1)$$

Under any FS s, the term $x^l_{sk}t(k,i)$ 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:

$$Minimize\ F=(1-w)U_{max}(\varnothing)+wU^{FS}_{Ave}\ ,\ 0 \le w \le 1 \quad (2)$$

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

subject to the following constraints:

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

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

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

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

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.

4.6.4 The Primary and Secondary Egress Point Selection Example

For better understanding of our robust egress point optimization problem, we provide an example in Figure 43(a)-(c). Figure 43(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 43(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. $PEP_{t(i1,k1)} \rightarrow l1$, $PEP_{t(i1,k2)} \rightarrow l2$ and $PEP_{t(i2,k2)} \rightarrow l2$. In addition, Figure 43(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. $SEP_{t(i1,k2)} \rightarrow l3$ and $SEP_{t(i2,k2)} \rightarrow l3$. Note that the traffic demand headed towards the unaffected destination prefix (i.e. *k1*) has remained intact.

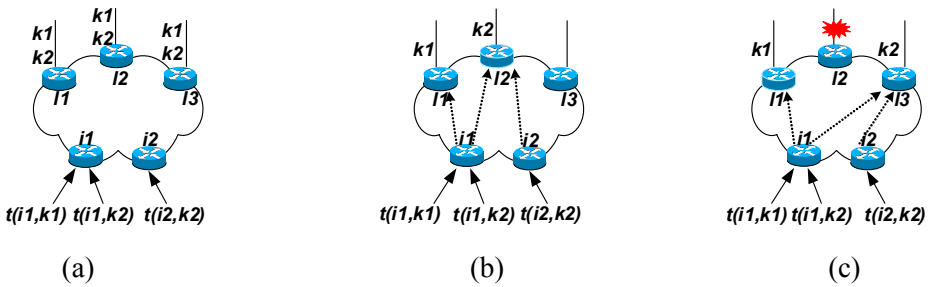


Figure 43. (a) Outbound TE inputs, (b) PEP Selection and (c) SEP Selection for *k2*

4.6.5 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.

4.6.5.1 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.

4.6.5.2 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 44, 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 44 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 44 line 2). List all the prefixes in $PEP_{current}$ assigned to the Most Utilized EP under the current FS ($l_s^{MostUtilized}$)[3].

[3] $l_s^{MostUtilized}$ 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 44 lines 3 to 7).

Step 2. Shift the prefix's PEP from $l_s^{MostUtilized}$ to the Least Utilized EP ($l_s^{LeastUtilized}$)[4] (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 44 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 44 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 44 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 44 lines 13-14).

4.6.5.3 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|*.

4.6.5.4 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.

4.6.5.5 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

[4] $l_s^{LeastUtilized}$ 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.

4.6.6 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[5] 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.

4.6.6.1.1 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 45. 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 5. Figure 45(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 45(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).

4.6.6.2 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

[5] 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 45(c) shows the result of the GLOBALR based on the PEP selection solution shown in Figure 45(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. $u_{l1}^{l2} = \frac{60+10+10+10}{100} = 0.9, u_{l1}^{l3} = \frac{80+40+10+10}{150} = 0.933$).

4.6.6.3 *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 45(d) shows the result of the GREEDYR based on the PEP selection solution shown in Figure 45(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 5 Input traffic flows

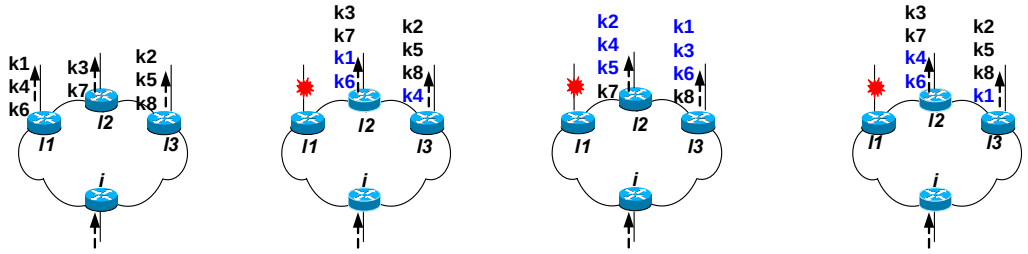


Figure 45. Different algorithms for destination prefix assignment

4.7 Resilience-aware BGP/IGP traffic engineering

4.7.1 Introduction

In general, Traffic Engineering (TE) is a technique that can be adopted by INPs to optimise the performance of their operational IP networks. Engineering the traffic within an AS boundary based on IGP, called *intra-AS TE*, is effectively the tuning of the link weights [FORT03, NUCC03, NUCC07,

SRID05], whereas selecting the best egress points for traffic to be sent to the next-hop ASes, called *inter-AS outbound TE*, is effectively the adjustment of BGP route attributes [FEAM03, BRES3]. Recent studies in [MARK04, BONA07] have shown that both intra- and inter-AS link failures are part of the daily routines in large IP backbone networks, and most of these failures are common and transient.

When a link fails, traffic is diverted to alternative paths, thus increasing the load on these new serving paths and possibly leading to congestion. To avoid this, one might take a reactive approach of re-computing the IGP link weights and/or BGP route attributes after the failure. However, this may not be practical for two reasons. First, due to the transient nature of failures, there would be insufficient time for INPs to re-compute the best post-failure TE configuration and implement it before the failed link is restored. Second, the new configuration will have to be advertised to every router in the network, and every router will have to re-compute the shortest path to every other router and to re-select its best egress point. This can lead to considerable instability, aggravating the situation already created by the link failure.

Although the reactive approach may not be appropriate or even feasible, transient link failures can be handled by computing the set of TE configurations in a proactive manner that is robust to all potential link failures. The goal of such a robust TE approach is to obtain a reasonably good network performance both under the normal state (i.e. absence of failures) and also under any potential link failure. Various kinds of robust TE approaches based on IGP link weight optimization and BGP egress selection have been proposed. These proposals, however, make their TE approaches robust either only to intra-AS or only to inter-AS transient link failures. They have neglected the interactions between robust intra- and inter-AS TE, specifically the impact of intra-AS link failures on robust inter-AS outbound TE and the impact of inter-AS link failures on robust intra-AS TE. As a result, the overall network performance may not be truly robust to link failures if these interactions are not considered. We therefore in the following sections investigate the impact of both intra- and inter-AS transient link failures on robust TE and propose a joint robust TE approach.

In the next section, we give an overview of our objective and design. We further explain the TE and link failure interactions with an illustrative example in Section 4.7.3. Section 4.7.4 presents the problem formulation of the joint robust TE approach. Then we detail our proposed two-phase heuristic in Section 4.7.5. Note that we present evaluation methodology and results in D4.2.

4.7.2 Overview of the objective and design

We propose a joint robust TE approach based on IGP link weight assignment for intra-AS and inter-AS outbound TE that is robust to all potential single intra- or inter-AS link failures. The goal is to find a set of IGP link weights that minimizes the intra- and inter-AS Maximum Link Utilization (MLU) under both the normal state and the worst case across all single link failure states while also taking Hot Potato Routing (HPR) into account.

As shown in Figure 46, to achieve our objective, the NP provisioning and maintenance functional block encompasses an offline joint robust TE optimiser component. The task of this component is to minimize the intra- and inter-AS link failure impact on TE performance by computing an optimum set of IGP link weights that takes HPR into account. Therefore, the offline joint TE optimiser component requires the following inputs: (a) the connectivity of intra- and inter-AS nodes and their link capacity, (b) the overall traffic matrix based on the subscribed CPA and NIA demands (from the NP mapping and NIA order handling functional blocks), (c) remote destination prefixes and their reachability information from BGP routing tables. The outputs of this unit is a set of IGP link weights that by taking HPR into account determines the intra-AS path both under Normal State (NS i.e. no failure) and Failure States (FS i.e. either intra- or inter-AS link failure) and also the egress point selection under both NS and FSs.

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

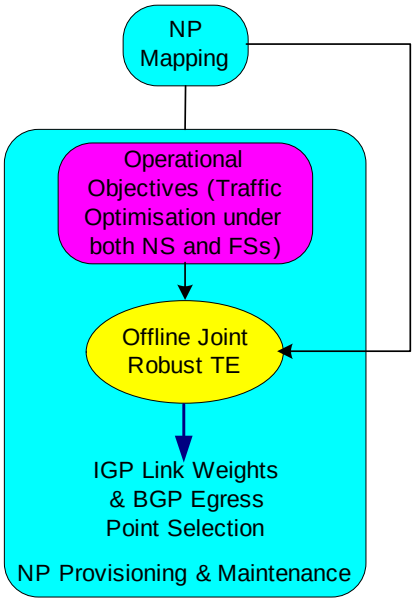


Figure 46 Joint Robust TE in AGAVE architecture

4.7.3 Example of interactions

In one scenario, if an inter-AS link (or egress point) fails, the inter-AS traffic is diverted from the failed egress point to other alternative egress points. This may cause a huge load increase not only at these new serving egress points but also at any link along the IGP paths between some ingress and the new egress points. In the other scenario where multiple egress routers have BGP routes that are equally good (i.e. they have the same local preference, AS path length, origin type, and multiple-exit-discriminator) for a routing prefix, each router in the AS directs the traffic to its closest egress point in terms of IGP distance. This is also known as Hot-Potato Routing (HPR). If an intra-AS link fails, the IGP distance between some ingress and egress points may change, causing thus some ingress points to divert the traffic to different egress points due to the HPR effect. These HPR changes are responsible for many of the large traffic shifts [TEIX05] in operational networks. Therefore, failure of an intra-AS link may shift a large proportion of traffic to other egress points and lead to a sudden load increase there. This may also result in excessive traffic to be sent to downstream ASes, violating the traffic exchange limits specified in their peering agreements.

In Figure 47(a-e) we illustrate how the aforementioned interactions, if not taken into account, can affect the robustness of the overall TE performance in terms of link failure. The performance metric we use is the intra- and inter-AS MLU under Normal State (NS) and some Failure States (FSs) where each FS corresponds to a single link failure. Link utilization is calculated as the total traffic load on the link divided by its bandwidth capacity. The intra-AS (or inter-AS) MLU under state *s* is the highest utilization among all the operational intra-AS (or inter-AS) links under that state.

The network in Figure 47 consists of three egress points (*j1*, *j2* and *j3*) with equal egress link capacity of 100 Mbps, two ingress points *i1* and *i2*, inter-AS traffic flows *t1=t_inter(i1,k1)*=40Mbps, *t2=t_inter(i1,k2)*=40Mbps, *t3=t_inter(i2,k3)*=20Mbps and remote destination prefixes *k1*, *k2* and *k3*, where *t_inter(i,k)* denotes the inter-AS traffic flow that enters the network from ingress point *i* and destined at prefix *k*. In this example, we assume that *k1* can be reached through all the egress points while *k2* can only be reached through *j2* and *k3* can be reached through *j1* and *j3* only. The network has several intra-AS links between ingress and egress points. The value on each link represents the IGP link weight. The capacity of bold links is 200Mbps while the capacity of the rest of the links is 100Mbps.

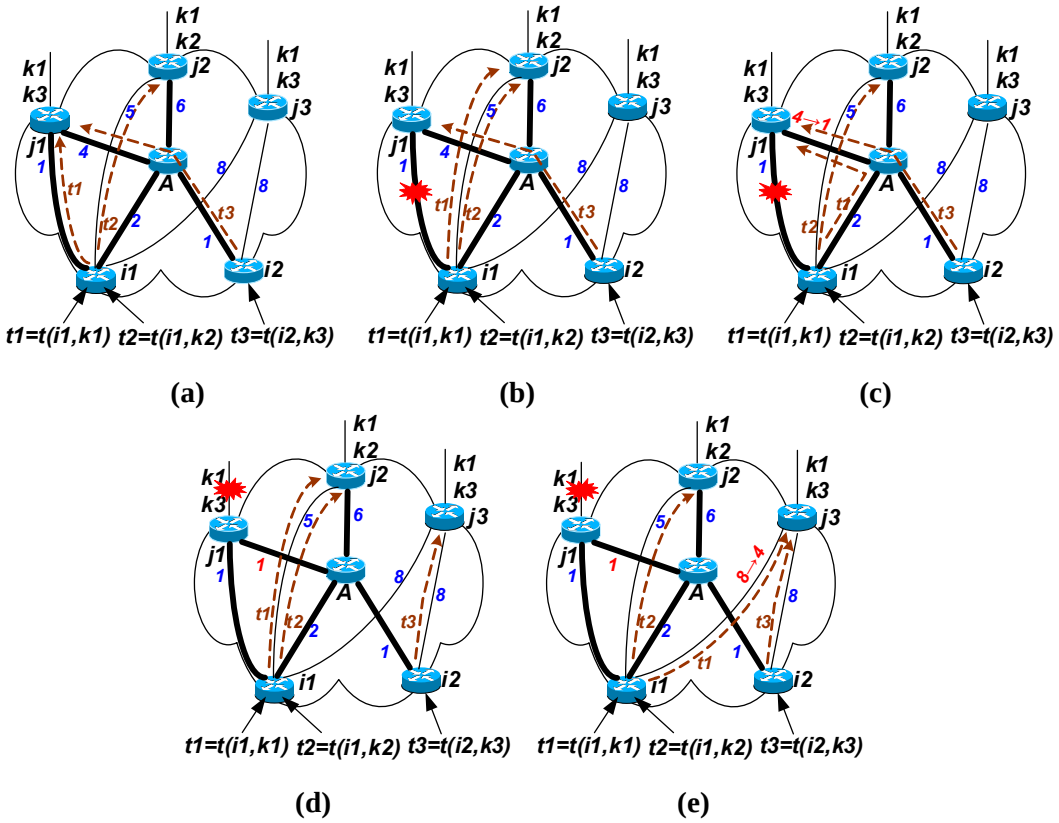


Figure 47 Traffic demand assignment under (a) NS, (b) *i1-j1* FS, (c) *i1-j1* FS with a changed IGP link weight, (d) *j1* FS, (e) *j1* FS with a changed link weight.

Note that throughout Section 4.5, we only consider the egress points that have "equally good" BGP routes towards each destination prefix. Therefore the egress point selection for the inter-AS traffic is determined by the IGP distance between individual ingress/egress pairs according to HPR. This scenario is inline with the fact that current ISPs often use HPR to control their inter-AS egress traffic.

Figure 47(a) shows the assignment of traffic flow *t1*, *t2* and *t3* to egress points *j1*, *j2* and *j1* respectively under NS. In this assignment, the inter- and intra-AS MLU would be on inter-AS link *j1* and intra-AS link *i1-j2* respectively and would be equal to ((40+20)/100,40/100)=(0.6,0.4).

Figure 47(b) shows the traffic flow assignment when intra-AS link *i1-j1* fails (i.e. *s={i1-j1}*). This failure disrupts the inter-AS traffic flow *t1* and shifts its egress point from *j1* to *j2* due to the HPR. The inter- and intra-AS MLU would then become ((40+40)/100,(40+40)/100)=(0.8,0.8) on inter-AS link *j2* and intra-AS link *i1-j2* respectively. Hence, the failure leads to an increase in the utilization of both intra- and inter-AS links.

However, this increased link utilization can be avoided if the IGP link weight of *A-j1* was set to 1. As shown in Figure 47(c), when the intra-AS link *i1-j1* fails, the egress point of *t1* would not change and the inter and intra-AS MLU would be reduced to ((40+20)/100,40/100)=(0.6,0.4). Hence, an appropriate IGP link weight setting can avoid increase in the link utilization and change of egress points for the inter-AS traffic.

Figure 47(d) shows the traffic assignment when inter-AS link *j1* fails (i.e. *s={j1}*). This failure shifts *t1* and *t3* from *j1* to *j2* and *j3* respectively. The shifting of traffic increases both the inter- and intra-AS MLU, which would become ((40+40)/100, (40+40)/100)=(0.8,0.8). Note that, in this case, change of the egress point due to HPR and disruption of *t1* and *t3* are inevitable, since the egress point *j1* has no reachability to *k1* anymore. By comparing Figures 1c and 1d, we observe that, even though the overall network utilization under a failure of intra-AS link has been improved by an IGP link weight change, it remains poor when an inter-AS link fails.

Nevertheless, such poor overall network utilization would not happen if the IGP link weight of *i1-j3* was set to 4. As shown in Figure 47(e), when the inter-AS link *j1* fails, the inter- and intra-AS MLU would become ((40+20)/100,40/100)=(0.6,0.4), which is identical to the results achieved under NS.

From this example, we can see that intra- and inter-AS link utilization can be improved with a set of appropriately configured link weights that takes into account both intra- and inter-AS transient link failures as well as the routing changes effects of HPR; that is the issue we investigate in the following sections.

4.7.4 Joint robust TE problem formulation

4.7.4.1 Inputs

As mentioned in Section 4.7.2, the following inputs are required for our problem.

1) Traffic Matrix (TM): this represents a matrix of traffic demand from each network point to each other over some time interval. In general, three types of traffic matrix can be identified in ISP networks. First of all, each element of the inter-AS traffic matrix, *t_inter(i,k)*, represents the total volume of inter-AS traffic from ingress point *i* towards destination prefix *k* that is reached through a downstream AS. Secondly, some traffic is destined locally within the network and we call this local traffic. Therefore, each element of this local traffic matrix, *t_loc(i,j)*, represents a volume of traffic from ingress point *i* destined to egress access point *j*. Finally, each element of the intra-AS traffic matrix, *t_intra(i,j)*, represents the total volume of intra-AS traffic from ingress point *i* destined to egress point *j*. Therefore, intra-AS traffic covers all the traffic that traverses the network including both the inter-AS traffic and local traffic. Thus, each element of the intra-AS traffic is the sum of local intra-AS and inter-AS traffic volume between each pair of ingress and egress nodes.

2) ***Network Topology*****:** this contains information about the connectivity of intra-, inter-AS nodes and link capacity.

3) ***Reachability of Destination Prefixes*****:** this consists of the advertisements of destination prefixes received by each egress point. This reachability information can identify which destination prefix can be reached through which egress points and it may be obtained from the BGP routing information base (Adj-RIB-In) of each egress router.

4.7.4.2 Problem formulation

Given the inputs, the objective of the joint robust TE is to minimize the intra- and inter-AS Maximum Link Utilization (MLU) under NS and also to minimize the worst-case intra- and inter-AS MLU across all intra- and inter-AS FSs. Each intra-AS (or inter-AS) FS corresponds to the network with a specific intra-AS (or inter-AS) link failure. By all states we include NS as well as all intra and inter-AS FSs. We denote intra-, inter-AS FSs and all states by S^{Intra} , S^{Inter} and S^{All} respectively and demonstrate them as follows.

$$S^{Intra} = \{\forall l \in L\} \tag{4.1}$$

$$S^{Inter} = \{\forall j \in J\} \tag{4.2}$$

$$S^{All} = \{\varnothing \cup (\forall l \in L) \cup (\forall j \in J)\} \tag{4.3}$$

As mentioned earlier, the intra-AS (or inter-AS) MLU under state *s* is defined as the highest utilization among all the operational intra-AS (or inter-AS) links under that state. Also, the worst-case intra-AS (or inter-AS) MLU across all states is the highest utilization among the MLU of all intra-AS (or inter-AS) states.

To achieve our objective, the optimization problem is to compute a set of IGP link weights that by taking the HPR into account determines the routes between each pair of ingress and egress points as well as the egress points for inter-AS traffic. We define $W=(w_1,w_2,...,w_l,...,w_n)$ as a vector of IGP link

weights where w_l is the weight of link l. We also define $x^l_{(i,j)}(s,W)$ as a binary variable and its value is equal to one if intra-AS traffic flow *t_intra(i,j)* traverses intra-AS link l under state s with IGP link weight setting W and zero otherwise. The worst-case intra-AS MLU across all states can be formulated as follows:

$$\underset{W}{Minimize}\, U^{intra}_{worst\_AllStates} = \underset{W}{Minimize} \underset{\forall s \in S^{All}}{Max}\, U^{intra}_{max}(s) \tag{4.4}$$

where

$$\forall s \in S^{All} : U^{intra}_{max}(s) = \underset{\forall l \neq s}{Max}(u^l_{intra}(s,W)) = \underset{\forall l \neq s}{Max}\left(\frac{\sum_{\forall i \in I}\sum_{\forall j \in J} x^l_{i,j}(s,W).t\_intra(i,j)}{c^l_{intra}}\right) \tag{4.5}$$

c^l_{intra} denotes the capacity of intra-AS link l and $u^l_{intra}(s,W)$ represents the utilization of l under state s with IGP link weight setting W. Note that the intra-AS MLU under NS ($U^{intra}_{max\_NS}$) can be calculated by (4.5) if state s represents only NS (i.e. $s=\varnothing$). If the failure states are limited to only intra-AS link failure (i.e. $s \in S^{Intra}$) then the expression in (4.4) represents the worst-case intra-AS MLU across only all intra-AS FSs (i.e. $U^{intra}_{worst\_IntraFSs}$). Similarly, if the failure states are limited to only inter-AS link failures (i.e. $s \in S^{Inter}$) then the expression in (4.4) represents the worst-case intra-AS MLU across only all inter-AS FSs (i.e. $U^{intra}_{worst\_InterFSs}$). In other words:

$$U^{intra}_{max\_NS} = U^{intra}_{max}(\varnothing) \tag{4.6}$$

$$U^{intra}_{worst\_IntraFSs} = \underset{\forall s \in S^{Intra}}{Max}\, U^{intra}_{max}(s) \tag{4.7}$$

$$U^{intra}_{worst\_InterFSs} = \underset{\forall s \in S^{Inter}}{Max}\, U^{intra}_{max}(s) \tag{4.8}$$

Clearly the worst-case intra-AS MLU under all FSs can be obtained as follows:

$$U^{intra}_{worst\_AllFSs} = \underset{\forall s \in S^{All}-\{\varnothing\}}{Max}(U^{intra}_{worst\_IntraFSs}, U^{intra}_{worst\_InterFSs}) = \underset{\forall s \in S^{All}-\{\varnothing\}}{Max}\, U^{intra}_{max}(s) \tag{4.9}$$

Similar to the above robust intra-AS TE problem formulation, we define $y^j_{(i,k)}(s,W)$ as a binary variable and its value is equal to one if inter-AS traffic flow *t_inter(i,k)* is assigned to egress point j under state s with IGP link weight setting W and zero otherwise. Hence, the worst-case inter-AS MLU across all states can be formulated as:

$$\underset{W}{Minimize}\, U^{inter}_{worst\_AllStates} = \underset{W}{Minimize} \underset{\forall s \in S^{All}}{Max}\, U^{inter}_{max}(s) \tag{4.10}$$

where

$$\forall s \in S^{All} : U^{inter}_{max}(s) = \underset{\forall j \neq s}{Max}(u^j_{inter}(s,W)) = \underset{\forall j \neq s}{Max}\left(\frac{\sum_{\forall i \in I}\sum_{\forall k \in K} y^j_{i,k}(s,W).t\_inter(i,k)}{c^j_{inter}}\right) \tag{4.11}$$

c^j_{inter} denotes the capacity of inter-AS egress link j and $u^j_{inter}(s,W)$ represents the utilization of j under state s with IGP link weight W. Similar to (4.6) to (4.8) for the inter-AS utilization we have

$$U^{inter}_{max\_NS} = U^{inter}_{max}(\varnothing) \tag{4.12}$$

$$U^{inter}_{worst\_IntraFSs} = \underset{\forall s \in S^{Intra}}{Max}\, U^{inter}_{max}(s) \tag{4.13}$$

$$U^{inter}_{worst\_InterFSs} = \underset{\forall s \in S^{Inter}}{Max}\, U^{inter}_{max}(s) \tag{4.14}$$

$$U^{inter}_{worst\_AllFSs} = \underset{\forall s \in S^{All}-\{\varnothing\}}{Max}(U^{inter}_{worst\_IntraFSs}, U^{inter}_{worst\_InterFSs}) = \underset{\forall s \in S^{All}-\{\varnothing\}}{Max} U^{inter}_{max}(s) \quad (4.15)$$

Therefore, the problem of our joint robust TE can be formulated as follows:

$$\underset{W}{Minimize}(U^{intra}_{max\_NS}, U^{intra}_{worst\_AllFSs}, U^{inter}_{max\_NS}, U^{inter}_{worst\_AllFSs}) \quad (4.16)$$

subject to the following constraints:

$$\forall i,i' \in I, k \in K, s \in S, g \in Out(k): \sum_{j \in Out(k)/Q(i,g,k)} y^{j}_{i,k}(s,W) + \sum_{j' \in Q(i,g,k)} y^{j'}_{i',k}(s,W) \le 1 \quad (4.17)$$

$$\forall j \in J, i \in I, k \in K, s \in S \text{ if } y^{j}_{i,k}(s,W)=1 \text{ then } j \in Out(k) \quad (4.18)$$

$$\forall i \in I, k \in K, s \in S: \sum_{j \in Out(k)} y^{j}_{i,k}(s,W)=1 \quad (4.19)$$

$$\forall j \in J, i \in I, k \in K, s \in S: y^{j}_{i,k}(s,W) \in \{0,1\} \quad (4.20)$$

Constraint (4.17) is the proximity constraint [BRES03], which ensures that the HPR is obeyed. In (4.17) Q is a utility function that is used to specify, for a given ingress node i and egress link j and a given prefix k, the set of alternative egress links for k that are closer than j. Thus $Q(i,j,k)$ is defined as the set of edge links where $Q(i,j,k)=\{g | g \in Out(k) \wedge d(i,g)<d(i,j)\}$. This proximity constraint ensures that if for some i, k and $g \in Out(k)$, the egress link for $t\_inter(i,k)$ is not selected from $Q(i,g,k)$ (that is if the egress link for $t\_inter(i,k)$ is chosen from $Out(k)\backslash Q(i,g,k)$ then for all i' the egress link for $t(i',k)$ cannot be chosen from $Q(i,g,k)$). Constraint (4.18) ensures that if the traffic flow from ingress point i destined to prefix k is assigned to egress point j under state s, then this prefix must be reachable through that egress point. Constraints (4.19) and (4.20) ensure that the traffic flow from ingress point i to prefix k is assigned to only one egress point that has routing reachability to this prefix under state s (i.e. there is no traffic splitting).

According to (4.16), our joint robust TE is a complex quadruple-objective optimization problem. To simplify the problem, we first categorize these four objectives into two wider objectives at intra- and inter-AS levels. We therefore have the joint robust TE problem reduced to a bi-objective optimization problem as follows:

$$\underset{W}{Minimize}(U^{intra}_{max\_NS}, U^{intra}_{worst\_AllFSs}) \quad (4.21)$$

$$\underset{W}{Minimize}(U^{inter}_{max\_NS}, U^{inter}_{worst\_AllFSs}) \quad (4.22)$$

However, these two objectives may be in conflict: intra-AS resource utilization may only be improved at the expense of degradation in the utilization of inter-AS resources and vice versa. Consequently, we need to further simplify the problem in order to eliminate such conflict. We therefore resort to using the ϵ-constraint method [CHAN83], in which the performance of an objective is optimized while the other one is constrained by not exceeding a tolerance value. Now the important question is which one of these objectives should be a constraint? Since inter-AS links are often bottleneck links in the Internet and significant amount of Internet traffic such as peer to peer traffic is routed across these links, we decided to put more efforts on bandwidth optimisation across inter-domain links. In addition, an inter-AS link is relatively more difficult to upgrade compared to an intra-AS link due to time-consuming and complicated negotiation between two ASes. It is also important to ensure that traffic exchange limits on peering agreements with downstream ASes are not violated. For these reasons, we place a constraint on the robust inter-AS TE objectives.

By placing a constraint on the utilization of inter-AS resources, the intra-AS resource utilization has to be optimized. However, this objective itself also consists of two conflicting objectives [NUCC03,NUCC07, SRID05]: improving the worst-case intra-AS MLU under all FSs may lead to performance degradation in the intra-AS MLU under NS. To further simplify the problem, we adopt a

weighted sum approach to transform these two intra-AS objectives into one. Therefore, the optimization problem of the joint robust TE can be formulated as follows:

$$\underset{W}{Minimize}(U^{intra}_{max\_NS}, U^{intra}_{worst\_AllFSs}) = \underset{W}{Minimize}((1-\alpha)U^{intra}_{max\_NS} + \alpha U^{intra}_{worst\_AllFSs}) \quad (4.23)$$

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

$$U^{inter}_{worst\_AllStates} \le \varepsilon \quad (4.24)$$

where $0 < \varepsilon \le 1$. The constraint ensures that the inter-AS MLU across all states is less than ε. Since $U^{inter}_{worst\_AllStates}$ can be calculated as follows

$$U^{inter}_{worst\_AllStates} = \underset{\forall s \in S^{All}}{Max}(U^{inter}_{max\_NS}, U^{inter}_{worst\_IntraFSs}, U^{inter}_{worst\_InterFSs}) \quad (4.25)$$

the above constraint implies that

$$U^{inter}_{max\_NS} \le \varepsilon \quad (4.26)$$

$$U^{inter}_{worst\_IntraFSs} \le \varepsilon \quad (4.27)$$

$$U^{inter}_{worst\_InterFSs} \le \varepsilon \quad (4.28)$$

According to the above problem formulation, we aim to optimize the intra-AS MLU under NS and the worst-case MLU among all intra-AS FSs while respecting the inter-AS utilization constraint across all states. Since optimizing the intra-AS MLU for both NS and FSs has been proven to be NP-hard [Nucc03,Nucc07,Srid05] and adding the inter-AS utilization constraint makes the problem even more complicated, we resort to heuristics to solve the problem efficiently.

4.7.5 Proposed two phase heuristic

We propose a two-phase heuristic to solve our problem. The first phase consists of a local search algorithm to find an initial set of IGP link weights that satisfies the inter-AS utilization constraint (4.24). Based on this set of IGP link weights, in the second phase, we optimize the link weights towards intra-AS TE objective (4.23) while preserving the inter-AS utilization constraint.

4.7.5.1.1 Phase I

The local search algorithm in phase 1 consists of three steps:

Step 1. *Initialization*: generate an initial solution ($W^{initial}$) by setting the weight of each link inversely proportional to its capacity. Run Dijkstra's SPF algorithm for $W^{initial}$ while taking into account HPR to determine the egress points for inter-AS traffic and the IGP routes between each pair of ingress and egress points. Calculate the initial worst-case inter-AS MLU under all states ($U^{inter\_initial}_{worst\_AllStates}$) using (4.25). Initialize the current solution ($W^{current} = W^{initial}$) and update the current performance metric ($U^{inter\_current}_{worst\_AllStates} = U^{inter\_initial}_{worst\_AllStates}$). If this value is less than the value of ε, then terminate the local search algorithm by returning the current IGP link weights as an input to the algorithm in phase II; otherwise proceed to steps 2 and 3.

Step 2. *Neighbourhood search*: a move is applied to transform the current solution into a neighbour solution. Perform a move by randomly picking up a link and increase or decrease its weight by a random value. Re-run Dijkstra's SPF algorithm for this new set of IGP link weights taking into account the HPR. Calculate the worst-case inter-AS MLU under all states ($U^{inter\_new}_{worst\_AllStates}$). If the new solution yields lower utilization than the current solution (i.e. $U^{inter\_new}_{worst\_AllStates} < U^{inter\_current}_{worst\_AllStates}$), accept the move by updating the current IGP link weights and performance metric ($W^{current} = W^{new}$, $U^{inter\_current}_{worst\_AllStates} = U^{inter\_new}_{worst\_AllStates}$); otherwise repeat this step until such a solution is found.

Step 3. *Check stopping criterion*: repeat step 2 for the next iteration until the current worst-case inter-AS MLU under all states ($U_{worst\_AllStates}^{inter\_current}$) is less than the value of ε. However, if there is no significant improvement on $U_{worst\_AllStates}^{inter\_current}$ after a certain number of iterations, this means that the algorithm is unlikely to find solutions that satisfy the desired inter-AS utilization constraint, possibly due to high amount of traffic load. In this case, we have to increase the value of ε by a step value denoted by *c*. In other words, $\varepsilon_{new} = \varepsilon + n \times c$, where *n* is a positive integer value, acts as a coefficient for the step value. The increase in the value of ε by coefficient *n* continues until a solution that satisfies the constraint is found. Once the relaxed constraint is satisfied, terminate the local search algorithm by returning the current IGP link weights as an input to the intra-AS TE optimization in phase II.

4.7.5.1.2 Phase II

Our algorithm in phase II follows the Tabu Search (TS) technique [GOLV97]. The procedure of our algorithm is as follows.

1) Neighbourhood search: we perform the following steps to identify the best move in the neighbourhood:

Step 1. Identify two sets of intra-AS links – those whose utilizations are within a small percentage of the MLU (heavily utilized) and those whose utilizations are within a small percentage of the minimum link utilization (lightly utilized). Take the most utilized link in the heavily utilized set into consideration.

Step 2. Increase the weight of the chosen link from the heavily utilized category by a random value in an attempt to remove the traffic from that link and reduce its load. Select a link randomly from the lightly utilized link set and decrease its weight by a random value in attempt to attract more traffic over this link from the highly utilized links.

Step 3. Run Dijkstra's SPF algorithm for the current IGP link weights with the HPR to re-calculate the egress points for the inter-AS traffic and the IGP routes for the intra-AS traffic. Then calculate objective function (4.23) and constraint (4.24).

Step 4. Repeat step 2 and 3 until either a feasible solution that satisfies the constraint is found or the upper limit of repetition is reached.

Step 5. Select the next most utilized intra-AS link and repeat steps 2 to 5 until all the links in the heavily utilized link set have been considered.

Step 6. Among all feasible solutions, choose the one with the minimum intra-AS MLU and consider it as the current solution.

2) Tabu list: The tabu list memorizes the most recent moves, operating as a first-in-first-out queue. As suggested in [GOLV97], the size of the tabu list depends on the size and characteristics of the problem. In our problem, the tabu list consists of the links whose weights have been recently changed and the amount of increase/decrease applied to the corresponding link weight.

3) 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 search and is applied when there is no obvious performance improvement after a certain number of iterations. For a diversification, several links are picked up from each of the lightly and heavily utilized link sets. The weights of the selected links from the former set are decreased while the weights of the selected links from the latter set are increased. Note that any solution produced by the diversification is acceptable if it is feasible.

4) Stopping Criterion: the search procedure stops if either the pre-defined maximum number of iterations is reached or there is no pre-defined performance improvement for objective function (4.23) after a certain number of consecutive diversifications.

5 AGAVE MONITORING CONSIDERATIONS

5.1 AGAVE monitoring architecture

Within the 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.

5.1.1 Interfaces

Within the 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 the 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.

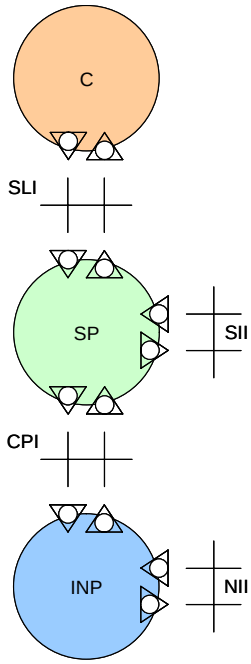


Figure 48 VoIP interfaces

5.1.2 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.

5.1.3 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.

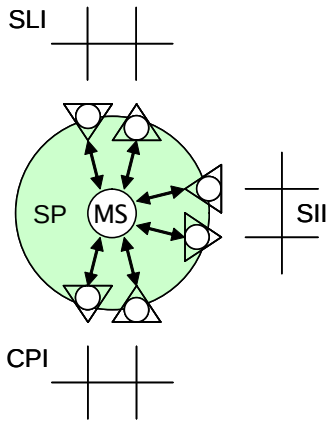


Figure 49 Service Provider Monitoring Server

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.

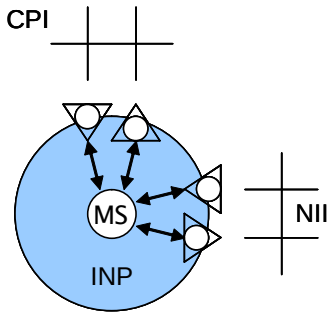


Figure 50 INP Monitoring Server

5.1.4 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.

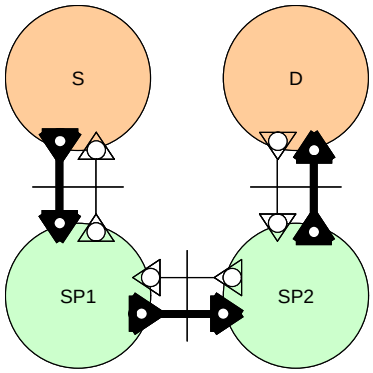


Figure 51 Call establishment monitoring

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.

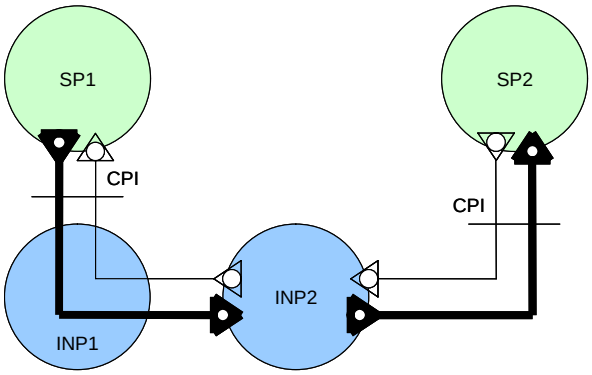


Figure 52 Remote CPI

5.1.5 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.

5.2 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 5.1.

5.2.1 Objectives

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*.

5.2.2 NP 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.

5.2.3 Other 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.

5.3 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.

5.3.1 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 rate 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.*

5.3.2VoIP 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.

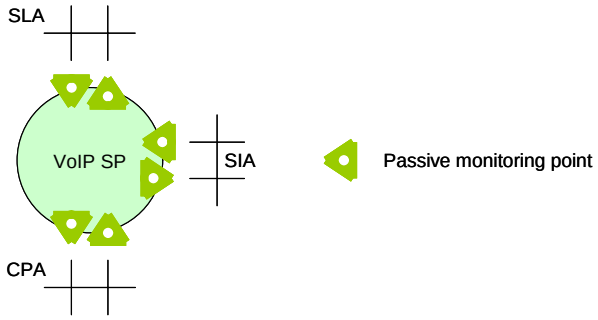


Figure 53 Passive VoIP monitoring

In order to encompass passive monitoring issues, an active monitoring could be used if VoIP service providers 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.

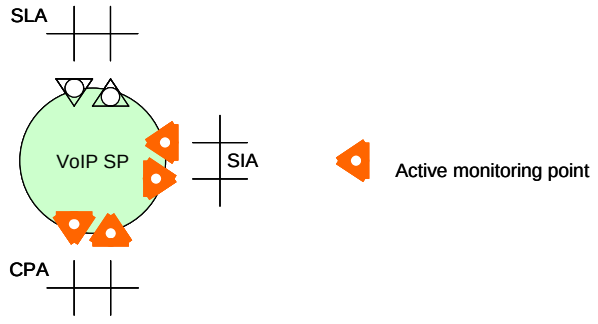


Figure 54 Active VoIP monitoring

5.3.3 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 infer 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 final specification of 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, MTR, 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 validation and evaluation of these work items will be performed in Work Package 4 (WP4) and the final results will be documented in Deliverable D4.2.

7 REFERENCES

[ABILE] The Abilene network topology and traffic traces:

http://www.cs.utexas.edu/~yzhang/research/AbileneTM/

[AGAR05] Agarwal, S., et al., *Measuring the Shared Fate of IGP Engineering and Interdomain Traffic*, Proc. *IEEE ICNP*, 2005

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

[ALAE00] Alaettinoglu, C., Jacobson, V. and Yu, H., *Towards Milli-Second IGP Convergence*, Internet Draft, draft-alaettinoglu-ISIS-convergence-00, Nov. 2000.

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

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

[ASGA04] Asgari, H., et al., *Scalable Monitoring Support for Resource Management and Service Assurance*, *IEEE Network Magazine*, November 2004.

[ATLA08] Atlas, A., et al., *Basic Specification for IP Fast Reroute: Loop-free Alternates*, IETF Internet Draft, draft-ietf-rtgwg-ipfrr-spec-base-11, Feb. 2008.

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

[BONA07] Bonaventure, O., et al., *Achieving Sub-50 Milliseconds Recovery Upon BGP Peering Link Failures*, *IEEE/ACM Transactions on Networking*, Vol.15, No.5, pp. 1123-1135, Oct 2007

[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*, Proc. *IEEE INFOCOM*, pp. 916-926, 2003

[BRYA07] Bryant, S., et al., *IP Fast Reroute using tunnels*, Internet Draft, draft-bryant-ipfrr-tunnels-03, Nov. 2007.

[CAGR05] M. A. Callejo Rodríguez, J. Andrés Colás, G. García de Blas, F. J. Ramón-Salguero and J. Enríquez-Gabeiras, *A Decentralized Traffic Management Approach for Ambient Networks Environments*, 16th IFIP/IEEE International Workshop on DSOM 2005, p.145-156. Springer.

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

[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.

[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.

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

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

[D3.1] Wang, N., et al, *Initial Specifications of Mechanisms, Algorithms and Protocols for Engineering the Parallel Internets*, AGAVE Deliverable D3.1, February 2007

[D4.1] Iannone, L. et al, *Test Specification and Experimentation Plan*, AGAVE Delvierable D4.1, July 2007

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

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

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

[FEAM03] Feamster, N., et al., *Guidelines for Interdomain Traffic Engineering*, *ACM SIGCOMM Computer Communications Review*, Vol. 33, No. 5, pp. 19-30, October 2003

[FORT00] Fortz, B., et al, *Internet Traffic Engineering by Optimizing OSPF Weights*, Proc. *IEEE INFOCOM*, pp. 519-528, 2000

[FORT03] Fortz, B., Thorup, M., *Robust Optimization of OSPF/ISIS Weights*, Proc. International Conference on Network Optimization, Oct. 2003.

[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.

[GEANT] The GEANT network topology: http://www.geant.net/upload/pdf/GEANT_Topology_12-2004.pdf

[GRIF07] D. Griffin et al., *Inter-domain Routing through Quality of Service Class Planes*, IEEE Communications, special issue on Quality of Service Routing Algorithms for Heterogeneous Networks, Vol. 45, No. 2, pp. 88-95, IEEE, February 2007.

[GOLV97] Golver, F., et al., *Tabu Search*, Kluwer Academic Publisher, Norwell MA 1997

[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.

[IDIPS00] Saucez D. et al., *IDIPS: ISP-Driven Informed Path Selection*, Internet Draft IETF Network Working Group, *draft-saucez-idips-00.txt,* February 2008.

[IDIPS-A] Bonaventure O. et al., *The case for an informed path selection service*, Internet Draft IETF Network Working Group, *draft-bonaventure-informed-path-selection-00.txt,* February 2008.

[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.

[LISP06] Farinacci D. et al., *Locator/ID Separation Protocol (LISP)*, Internet Draft IETF Network Working Group, *draft-farinacci-lisp-06.txt*, February 2008.

[MARK04] Markopulou, A. et al., *Characterization of Failures in an IP Backbone*, Proc. *IEEE INFOCOM*, Vol. 4, pp. 2307-2317, 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.*

[NELA07] Nelakuditi S., et al., *Fast Local Rerouting for Handling Transient Link Failures*, *IEEE/ACM Transactions on Networking*, Apr. 2007.

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

[NUCC03] Nucci, A., et al., *IGP Link Weight Assignment for Transit Link Failures*, Proc. *International Teletraffic Conference (ITC)*, 2003

[NUCC07] Nucci, A., et al., *IGP Link Weight Assignment for Operational Tier-1 Backbones*, *IEEE/ACM Transactions on Networking*, Vol. 15, No. 4, pp. 789-802, August 2007

[RAEG03] F. J. Ramón-Salguero, J. Andrés-Colás, J. Enríquez-Gabeiras and G. García-De Blas. *Estrategias de encaminamiento dinámico para posponer la congestión de la red*, Telecom I+D, 2003.

[REAM02] F. J. Ramón-Salguero, J. Enríquez-Gabeiras, J. Andrés-Colás and A. Molíns-Jiménez. *Multipath Routing with Dynamic Variance*, COST 279 Technical Report TD02043, 2002.

[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.

[RFC1853] Simpson, W. *IP in IP Tunneling*, IETF RFC 1853, Oct. 1995.

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

[RFC2784] Farinacci, D. et al., *Generic Routing Encapsulation (GRE)*, IETF RFC 2784, Mar. 2000.

[RFC3931] Lau, J. et al., *Layer Two Tunneling Protocol – Version 3 (L2TPv3)*, IETF RFC 3931, Mar. 2005.

[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.

[RFC4915] Psenak, P. et al, *Multi-Topology (MT) Routing in OSPF*, IETF RFC 4915, July 2007.

[RFC5120] Przygienda, T. et al, *M-ISIS: Multi Topology (MT) Routing in IS-IS*, IETF RFC 5120, Feb. 2008.

[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. [SPRI04] Spring, N. et al., *Measuring ISP Topologies with Rocketfuel*, IEEE/ACM Transactions on Networking, Vol. 12, No. 1, February 2004, pp2-16

[SPRI04] Spring, N. et al., *Measuring ISP Topologies with Rocketfuel*, IEEE/ACM Transactions on Networking, Vol. 12, No. 1, February 2004, pp2-16

[SRID05] Sridharan, A. et al., *Making IGP Routing Robust to Link Failure*, Proc. *IFIP Networking*, 2005

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

[TEIX05] Teixeira, R. et al., *Traffic Matrix Reloaded: Impact of Routing Changes*, Proc. *PAM Workshop*, March 2005

[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.