Internet Traffic Engineering

O. Bonaventure[1], P. Trimintzios[2], G. Pavlou[2], B. Quoitin[3] (Editors)
A. Azcorra[4], M. Bagnulo[4], P. Flegkas[2], A. Garcia-Martinez[4], P. Georgatsos[5], L. Georgiadis[6], C. Jacquenet[7], L. Swinnen[3], S. Tandel[3], S. Uhlig[1]

[1] CSE Dept, Université Catholique de Louvain, Belgium
[2] Centre for Communication Systems Research,
University of Surrey, Guildford, Surrey, GU2 7XH, U.K.
[3] Infonet group, University of Namur, Belgium
[4] Departamento de Ingeniera Telemtica,
Universidad Carlos III, Madrid, Spain
[5] Algonet S.A. 206 Syggrou Ave, 17 672, Athens, Greece
[6] Electrical and Computer Engineering Dept.
Aristotle University of Thessaloniki, 54 124, Thessaloniki, Greece
[7] France Telecom, Rennes, France

Abstract. Traffic engineering encompasses a set of techniques that can be used to control the flow of traffic in data networks. We discuss several of those techniques that have been developed during the last few years. Some techniques are focused on pure IP networks while others have been designed with emerging technologies for scalable Quality of Service (QoS) such as Differentiated Services and MPLS in mind. We first discuss traffic engineering techniques inside a single domain. We show that by using a non-linear programming formulation of the traffic engineering problem it is possible to meet the requirements of demanding customer traffic, while optimising the use of network resources, through the means of an automated provisioning system. We also extend the functionality of the traffic engineering system through policies. In the following, we discuss the techniques that can be used to control the flow of packets between domains. First, we briefly describe interdomain routing and the Border Gateway Protocol (BGP). Second, we summarise the characteristics of interdomain traffic based on measurements with two different Internet Service Providers. We show by simulations the limitations of several BGP-based traffic engineering techniques that are currently used on the Internet. Then, we discuss the utilisation of BGP to exchange QoS information between domains by using the QOS_NLRI attribute to allow BGP to select more optimum paths. Finally, we consider the multi-homing problem and analyse the current proposed IPv6 multi-homing solutions are analysed along with their impact on communication quality.

1 Introduction

The Internet is becoming a targeted support for a wide range of IP service offerings, ranging from dial-up access to more sophisticated offers, such as Virtual

Private Networks. The success of some of these services is now conditioned by the ability of the service providers to commit on the provisioning of a guaranteed level of quality, which will possibly be negotiated with the customer, and which will depend on the network resource availability.

Internet Traffic Engineering (TE) is one of the methods that can be used by service providers to commit to those guarantees. TE is defined as that aspect of Internet network engineering dealing with the issue of performance evaluation and performance optimisation of operational IP networks [1]. Different techniques are applicable for isolated domains, also called autonomous systems (AS) and for the global Internet which is composed of more than 13.000 distinct ASes. This chapter discusses both these applications of traffic engineering.

Inside a single domain, Traffic Engineering systems can be categorised based on the different TE styles and views [1]. Time-dependent TE uses historical information based on periodic variations in traffic to pre-program routing plans and other TE control mechanisms, while state-dependent TE adapts dynamically the routing plans based on the current state of the network. The route computation can be done either on-line or off-line. In centralised TE a central authority determines the routing plans and other TE control parameters, while in distributed TE route selection is determined by each router autonomously, based on the router's view of the state of the network.

Differentiated Services (DiffServ) [2] is the emerging technology to support Quality of Service (QoS) in IP backbone networks in a scalable fashion. Multi-Protocol Label Switching (MPLS) [3] can be used as the underlying technology to support traffic engineering. It is possible to use these technologies in conjunction in order to provide adequate quality guarantees for traffic with QoS requirements. This can be done through careful traffic forecasting based on contracted services with customers and subsequent network provisioning in terms of routing and resource provisioning. In the first part of this chapter we discuss how to meet traffic requirements while optimising the use of the intra-domain resources with a traffic engineering system. In addition we use the policy-based management paradigm [4] to dynamically guide the behaviour of the traffic engineering system that provisions the network in order to meet high-level business objectives.

Besides the need to optimise the flow of packets inside a network, there is also a strong need to optimise the flow of packets between networks. At the time of writing, MPLS is not yet used to solve traffic engineering purposes between domains. Thus the only solution today is to rely on the existing interdomain routing protocol, namely the Border Gateway Protocol [5].

In the second part of this chapter, we focus on four issues related to the support of traffic engineering across interdomain boundaries. The first issue has to do with the characteristics of the interdomain traffic that we summarise based on measurements taken from several ISPs. A second issue is how BGP can be used to control the flow of best-effort interdomain traffic. We highlight the limitations of the current BGP-based traffic engineering techniques through simulations. However, as mentioned earlier, Differentiated Services are more and more used inside isolated autonomous systems and there is a clear need to support similar services across interdomain boundaries. One of the ways to provide such services is by extending BGP to distribute QoS information. The last issue that we address is that autonomous systems are often multi-homed, i.e. connected to several providers. This multi-homing is often used to improve the performance or reduce the cost of the interdomain traffic. However, this multi-homing also stresses the interdomain routing system by increasing the size of the BGP routing tables. Better multi-homing strategies are currently being developed for IPv6.

This chapter is divided in two main parts. The first part focuses on intra-domain traffic engineering and begins with Sect. 2 that presents the related work. Then, Sect. 3 describes a Traffic Engineering System Architecture. Section 4 describes the system's operation cycle and in Sect. 5 we present an algorithm for network dimensioning, i.e. offline traffic engineering, with some simulation results. In Sect. 6 we enlist the potential policies related to network dimensioning and we present a policy enforcement example.

The second part of the chapter is devoted to the issues that arise when considering the interconnection of distinct domains. In section 7 we briefly describe interdomain routing and the BGP protocol. Then, in section 8, we summarise the characteristics of interdomain traffic. In section 9.1, we show the difficulty of selecting Internet paths based on the current BGP attributes by studying BGP routing tables from various ISPs. In section 9.2, we present a detailed simulation study of the performance of one technique often used by ISPs to control their incoming traffic. Section 10 is devoted to the discussion of QoS extensions to BGP. Finally, in section 11 the discuss several approaches to the multi-homing problem in both IPv4 and IPv6 environments.

2 Related Work in Intra-domain Traffic Engineering

The problem of traffic engineering has attracted a lot of attention in recent years. Traffic Engineering entails the aspect of network engineering that is concerned with the design, provisioning, and tuning of operational Internet networks. In order to deal with this important emerging area, the Internet Engineering Task Force (IETF) has chartered the Internet Traffic Engineering Working Group (tewg) [6] to define, develop, specify, and recommend principles, techniques and mechanisms for traffic engineering in IP-based networks. The IETF has defined the basic principles for traffic engineering [1], the requirements for MPLS traffic engineering [7], and the requirements to support the inter-operation of MPLS and Diffserv for traffic engineering [8]. It is in the plans of tewg to look into technical solutions for meeting the requirements for Diffserv-aware MPLS traffic engineering, the necessary protocol extensions, inter-operability proposals and measurement requirements. In addition there some recent proposals in the IETF to extend the information included in Link State Advertisements (LSAs) of intra-domain routing protocols, like Open Shortest Path First (OSPF). The traffic engineering extensions [9] to OSPF, will enable the flooding of the extended LSAs to all routers within a domain, which will then store the received TE information into a TE Database (TED) so that it can be facilitated by constraint path computation algorithms.

Two similar works with the work presented here are the Netscope [10] and RATES [11]. Both of them try to automate the configuration of the network in order to maximise network utilisation. The first one uses measurements to derive the traffic demands and then by employing the offline algorithm described in [12] it tries to offload overloaded links. The latter uses the semi-online algorithm described in [13] to find the critical links which if they are chosen for routing will cause the greatest interference (i.e. reduce the maximum flow) of the other egress-ingress pairs of the network. Both of these works do not take into account any QoS requirements and only try to minimise the maximum load of certain links.

The traffic engineering and provisioning work we will describe in the following sections is aimed to be used in conjunction with service management functionalities [14], and all together could be parts of an extended Bandwidth Broker (BBs) [15]. A BB is an agent that works within a domain, keeps track of the domain's resource allocation and accepts or rejects new requests for using the network [15]. The extended BB could in addition provision the network, and thus by creating the resource allocation is in position to take more flexible decisions. Another functionality to which a BB can play an intermediate role is that of end-to-end allocation of resources, built through the peering connections with the adjacent domain's BBs. There were recently some proposals for such end-to-end frameworks [16] and [17]. The work in [16] proposes a two level admission control, one for the provisioning of QoS-pipes between and within domains, and a second level for individual flows which make use of the pre-provisioned QoS-pipes. This two level approach to admission control, proposed also in [18], fits very well with our two-level traffic engineering system and thus could be used in conjunction, to form parts of a BB. This chapter will focus on the traffic engineering and provisioning functionalities of the BB, a broad discussion on the service engineering functionalities can be found in [19].

The offline traffic engineering and provisioning, which we collectively call *network dimensioning*, algorithm described later in this chapter targets to solve problems that can be categorised as (class-based) *offline* traffic engineering [1]. Such problems can be naturally modelled as multi-commodity network flow optimisation problems [20]. The related works use optimisation formulations, focusing on the use of linear cost functions, usually the sum of bandwidth requirements, and in most of the cases they try to optimise a single criterion, i.e. minimise the total network cost.

The advantage of the linear problem formulation is that it can be optimally solved by using linear programming methods, i.e. the network simplex algorithm [20]. On the other hand, a linear cost function of link load does not penalise heavily-loaded links enough, resulting in poorer traffic distribution across the network. In addition, such linear formulations can take into account more than one optimisation criteria as a linear combination of the respective objectives which is not very flexible. In our approach presented here, we formulate the problem in a non-linear fashion, combining as criteria the minimisation of both total network cost and of maximum link load.

In [21] the traffic-engineering problem is seen as a multi-priority problem, formulated as a multi-criterion optimisation problem on a predefined traffic matrix. This approach uses the notion of predefined admissible routes that are specific for each QoS class and each source-destination pair, where the objective is the maximisation of the carried bandwidth. In [22], the authors address the resource allocation and routing problem in the design of Virtual Private Networks (VPNs). The main objective is to design VPNs which will have allocated bandwidth on the links of the infrastructure network such that, when the traffic of a customer is optimally routed, a weighted aggregate measure over the service provider's infrastructure is maximised, subject to the constraint that each VPN carries a specified minimum. The weighted measure is the network revenue, which is a function of the traffic intensity. The algorithm proposed in that paper solves first the optimal routing problem for each VPN independently. Then it calculates for each VPN the linear capacity costs for all the links. These quantities are used to modify appropriately the current capacity allocations so that the network revenue of the infrastructure network for the new capacities is maximised. It is shown in [22] that this is equivalent to minimising a linear function of the capacity costs subject to constraints imposed by the link capacities.

In [23] a model is proposed for off-line centralised traffic engineering over MPLS. This uses one of the following objectives: resource-oriented or traffic-oriented traffic engineering. The resource-oriented problem targets load balancing and minimisation of resource usage. Capacity usage is defined as the total amount of capacity used and load balancing is defined as one minus the maximal link utilisation. The objective function that has to be maximised is a linear combination of capacity usage and load balancing, subject to constraints imposed by the capacity of the links. The traffic-oriented model suggests an objective function that is a linear combination of fairness and throughput, where throughput is defined as the total bandwidth guaranteed by the network and fairness as the minimum weighted capacity allocated to a traffic trunk. In [24] the authors propose an algorithm which has two phases, a pre- processing phase and an on-line one. In the pre-processing phase the algorithm uses the notion of multi-commodity flows, where commodities correspond to traffic classes. The goal is to find paths in the network to accommodate as much traffic as possible from the source to the destination node. The algorithm tries to minimise a linear cost function of the bandwidth assigned to each link for a traffic class. The second phase performs the on-line path selection for LSP requests by using the pre-computed output of the multi-commodity pre-processing phase.

The offline traffic engineering works discussed so far, are assuming that the anticipated traffic is estimated in the form of a traffic matrix on ingress to egress node basis with fixed quantities as expected bandwidth requirements. The work in described in [25], relaxes the last requirement, and proposes the stochastic traffic engineering framework, where the entries in the traffic matrix are not fixed values but are based on some Gaussian distribution. The authors found that the variability of the demand has a great impact on path selection. Though this last work improves the assumption on traffic matrix with fixed values, it is still based on the ingress to egress assumption, which is known as the pipe model. The authors of [26] first proposed the hose resource provisioning model, where there is no need for a full traffic matrix but we only need to know the total traffic an border node injects/receives into/from the network. This work introduced algorithms for designing minimum cost networks based on the hose model, based on the steiner tree approach [26], while in [27] proved that the optimal hose provisioning problem is *NP*-hard and proposed some heuristics. Finally, [28] discusses the bandwidth efficiency of the hose model compared to the pipe and gives a lower bound for the hose model realisation.

Works like [12], [29] and [30] try to achieve optimal routing behaviour by appropriately configuring the shortest path routing metrics, assuming no MPLS is supported by the network. Wang et al. in [29] proved theoretically that any routing configuration, including the optimal one, could be achieved by the appropriate setting of the shortest path routing metrics.

Finally, online algorithms are mainly based on extensions of the QoS-routing [31], [32]. These approaches are heuristics, recently known in the IETF as Constraint Shortest Path First (CSPF), which utilise information kept in traffic engineering databases populated through information obtained from the routing flooding mechanisms [9] about link capacities, unreserved capacity, colour affinities etc. Other online traffic engineering approaches [33], [34] and [35] mainly focus on load balancing on multiple equal or non-equal cost paths.

3 Two-level Intra-domain Traffic Engineering

In [36] we proposed a two-level traffic engineering system operating both at long-to-medium and medium-to-short time scales. The relevant architecture is depicted in Fig. 1.

At the long-to-medium time scale, Network Dimensioning maps the traffic requirements to the physical network resources and provides dimensioning directives in order to accommodate the predicted traffic demand. At the medium-to-short time scales, we manage the routing processes in the network, performing dynamic load balancing over multiple edge-to-edge paths, and we ensure that link capacity is appropriately distributed among the PHBs in each link by appropriately selecting the scheduling discipline and buffer management parameters. This part is realised by the Dynamic Route and Dynamic Resource Management.

Dynamic Route Management operates at the edge nodes and is responsible for managing the routing processes in the network, performing mainly dynamic load balancing. An instance of Dynamic Resource Management operates at each router and aims to ensure that link capacity is appropriately distributed among the PHBs in that link by setting the relevant buffer and scheduling parameters. By setting appropriately how the link capacity is partitioned between the several queues and if they can make use of any *unused* capacity or if they act in isolation,

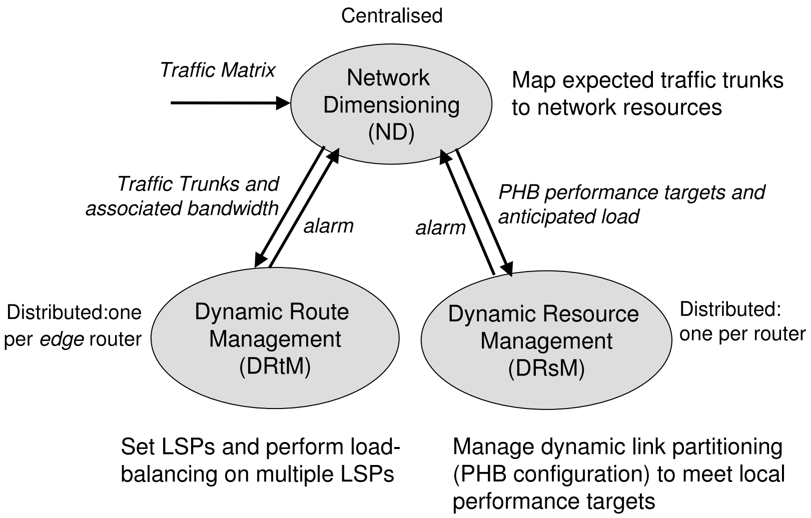
Centralised
Traffic Matrix
Network Dimensioning (ND)
Map expected traffic trunks to network resources
Traffic Trunks and associated bandwidth
alarm
alarm
PHB performance targets and anticipated load
Distributed:one per *edge* router
Dynamic Route Management (DRtM)
Dynamic Resource Management (DRsM)
Distributed: one per router
Set LSPs and perform load-balancing on multiple LSPs
Manage dynamic link partitioning (PHB configuration) to meet local performance targets

Fig. 1. Two-level Traffic Engineering

as in hierarchical scheduling disciplines, we can achieve the performance targets that were set by Network Dimensioning.

It should be noted that in this traffic engineering approach, MPLS LSPs are used purely to denote a set of explicit paths, without having explicitly assigned bandwidth within the network, while the proposals in the IETF [7], [37] and research efforts [22], [23] assume that bandwidth is assigned to LSPs. Network dimensioning comes up with a set of LSPs realising a traffic trunk and *logically* associated bandwidth, e.g. for the A-Z 1.2Mbps trunk, LSPs A-B-D-Z 0.4Mbps, A-E-Z 0.5Mbps and A-F-G-Z 0.3Mbps may be produced, but the bandwidth association is only kept at the ingress node (A in this case), with Dynamic Route Management performing load balancing of incoming traffic to those LSPs.

In summary, through this approach network provisioning is effectively achieved by taking into account both the long-term service level subscriptions in a time-dependent manner and the dynamic network conditions that are state-dependent. We argue that both levels are necessary when considering an effective traffic engineering solution, since when used independently they have shortfalls (*centralised* vs. *distributed, time-* vs. *state-dependent, static* vs. *dynamic*), which can only be overcome when they are used in conjunction.

4 Service-driven Class-based Traffic Engineering

The Traffic Engineering system presented in the Sect. 3 does not act in isolation but is driven by service level functions [14], not depicted in Fig. 1, for offering and establishing Service Level Agreements (SLAs). The service-driven TE work is based on the technical part of an SLA, which is called the Service Level Specification (SLS) [38]. The SLSes are used for handling the admission of service requests. These service level functions set the traffic-related target objectives of the Traffic Engineering functions to achieve. More specifically, the service layer provides the Traffic Matrix to the Traffic Engineering functions, which specifies anticipated QoS traffic demand between network edges.

The traffic engineering is also class-based and we use the notion of quality of service class (QoS-class), which represents traffic that belongs to a particular PHB and has delay and/or packet loss requirements within a particular range (see Table 1). The entries in the Traffic Matrix are the traffic trunks. A traffic trunk represents aggregate traffic that belongs to a particular QoS-class and has a certain topological scope (see Table 1). In fact, traffic trunks are aggregates of QoS traffic having the transfer characteristics of the associated QoS-class between network edges of the provider's domain.

Traffic demand is forecasted from the current SLS subscriptions, historical data and the Service Providers' expectations (e.g. sales targets). Based on these traffic forecasts, the network is appropriately dimensioned (i.e. off-line traffic engineered) by the TE functions, in terms of PHBs and their configuration parameters and in terms of QoS route constraints.

Table 1. Definition of QoS-class and Traffic Trunk

QoS-class	:	PHB,	Delay range,	Packet Loss range	
Traffic Trunk	:	QoS-class,	Ingress IP address,	Egress IP address,	Bandwidth

4.1 Traffic Forecast

Traffic forecasting for Traffic Matrix derivation has attracted the attention of many researchers in the last years [39], [40], [41], with considerable results. All the relevant works up to now are based on statistical processing of measurement and historical data. We believe that in the near future advanced services will be offered and customers are going to subscribe to such services through the notion of the SLS [38]. Thus we propose that Traffic Forecasting should also take into account the customer subscriptions, in addition to network measurements. In this section we will discuss a few issues on Traffic Matrix derivation based on customer service subscriptions.

The role of the Traffic Forecast is to estimate the anticipated traffic for each forecasting period. The forecasting periods are determined based on the forecasting period schedule. For each forecasting period, four successive functions are performed: translation, mapping, aggregation, and forecasting.

Translation. The Customer SLSes stored in the repository use a customer-oriented terminology. These SLSes have the following semantics. They are the either bi-directional or unidirectional and follow either the pipe or the VPN

(Virtual Private Network) model. The first function of translation is to decompose these SLSes into a number of simple unidirectional SLSes, which follow only the pipe model (one ingress to one egress). The ingress and egress are specified in geographical terms, therefore there is a need to translate those into IP source-destination addresses and from them to infer the specific ingress-egress addresses of our Autonomous System (AS). Services are given names (Olympic Gold/Silver/Bronze Service, Virtual Leased Line). This terminology must use networking semantics (throughput, delay, loss). So, another important aspect of translation is from the customer SLS to the appropriate network parameters.

Mapping. The simple unidirectional SLSes could potentially request (or be translated to) any value of delay, loss or throughput, while the network supports only a few discrete values. The mapping function is therefore to map the SLS requirements to the services actually supported by the network, i.e. the QoS classes (see Table 1). Another important function is to map the IP source-destination addresses. To illustrate the mapping function, consider the following simple example. A customer SLS may require "1Mbps of Virtual Leased Line Premium service with 50ms delay between London and Manchester, . . . ". The translation function would translate this Customer SLS into to two simple unidirectional SLSes: "1Mbps, EF, 50ms delay, from 122.12.2.143, egress 103.124.32.111, . . . " and another one with the same values but in the opposite direction. Suppose the network offers 2 services: a 10ms and 100ms delay premium virtual wire services, therefore we need to map the SLS to the 10ms service. The mapping is static as far as the mapping algorithms are fixed and deterministic. The result is a list of SLSes that are active during the next provisioning period and are defined in network terms.

Aggregation. Up to this point of the forecasting procedures we had information proportional to the number of customers and SLSes. For scalability, we base our TE solution on traffic loads per ingress/egress pair and per class of service. The simple unidirectional SLSes are aggregated into traffic trunks by "adding" their throughput requirements if and only if they have the same ingress/egress pair and they require a similar treatment, i.e. they belong to the same QoS class.

Forecasting has two main functions. On one hand, an over-subscription factor per QoS class is included. This factor is defined as the ratio of the capacity reserved by all the SLSes in a given QoS class to the capacity expected to be actually used. For expensive SLS types, the over-subscription factor is likely to be one. For cheaper services, the factor may be larger. On the other hand, at this stage we have all the ingress-egress demand (traffic matrix). It is possible to run an extrapolation algorithm utilising the information on the history of traffic matrices. Candidate algorithms are the spline or more generally any polynomial extrapolation method.

4.2 Provisioning and Forecasting Scheduling

In this section we describe the timing and scheduling properties of our system. We assume that scheduling in the proposed model is performed at two levels:

1. The Network Provisioning Scheduler, which defines provisioning periods.
2. The Traffic Forecast Scheduler, which defines forecasting periods.

Network Provisioning Scheduler. The proposed Traffic Engineering System aims to provision the network in order to provide the required QoS to contracted SLSes while at the same time optimising the usage of resources. As the traffic requirements that are derived from the SLSes change over time, the provisioning guidelines need to be updated. This is performed periodically through the Network Provisioning Scheduler. The system is required to recalculate a set of provisioning guidelines for each provisioning period. As the current period comes to an end, the Provisioning Scheduler will trigger re-provisioning of the network. Typical frequencies for the scheduler could be once a day or once a week.

Traffic Forecast Scheduler. As the provisioning periods are quite long (days, weeks), the traffic requirements may vary during a provisioning period. The granularity of the Network Provisioning Scheduler is not fine enough to optimise the configuration of the network. We therefore use a second scheduler, the Traffic Forecast Scheduler, which has the following characteristics (see also Fig. 2):

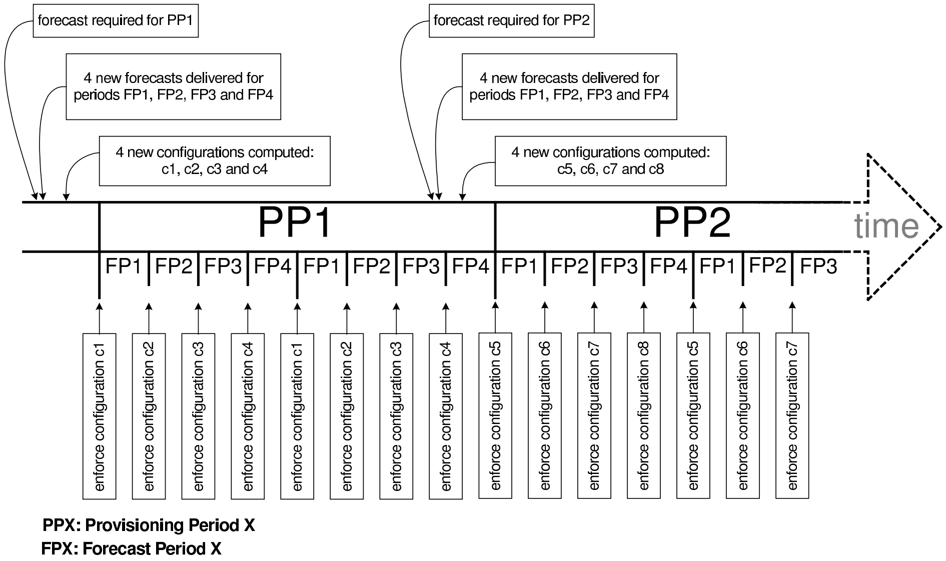

Fig. 2. Two-level scheduling

- It is not necessarily periodic (could schedule one 4 hour period followed by one 10 hour period)
- For all forecasting periods $FP(i)$ that constitute a provisioning period PP, it holds: $FP(i) = PP$

The principle behind the double scheduling is that network provisioning is invoked once every provisioning period, and calculates multiple configurations, which will be enforced at each forecasting period triggered by the Traffic forecast scheduler.

Scheduling Example. To clarify the roles of each scheduler, this section provides some examples. Consider that provisioning period is a week and that Traffic Forecast Scheduler defines 4 6-hour forecasting periods (forecasting period being periodic in this case). This means that once a week, 4 logical topologies are calculated: one for the morning, one for the afternoon, one for the evening and one for the night (6 hour periods). During each day of the week, Traffic Forecast Scheduler triggers at 06:00 to enforce the morning configuration, at 12:00 to enforce the afternoon configuration, at 18:00 to enforce the evening configuration and at 24:00 to enforce the night configuration. This is shown in Fig. 2.

There is no requirement for the Traffic Forecast Scheduler to be periodic. For example, we could have 4 forecasting periods per day, morning, afternoon, evening and night with different lengths as shown in Table 2.

Table 2. Forecasting periods schedule example

Morning	: 08:00 - 12:00
Afternoon	: 12:00 - 17:00
Evening	: 17:00 - 22:00
Night	: 22:00 - 08:00

Another more realistic and more complex example could be to have 5 forecasting periods: weekday morning, afternoon, evening, night, and all weekend. The interesting feature of this arrangement is that the Traffic Forecast Scheduler does not trigger the enforcement of the configurations in a simple cyclical fashion: the 4 weekday configurations must be implemented cyclically for 5 days, and the weekend configuration is enforced for the remaining two days of the week.

5 Network Dimensioning

Network Dimensioning (ND) is time- and state-dependent offline traffic engineering. It performs automated provisioning and is responsible for the long to medium term configuration of the network resources. By configuration we mean the definition of LSPs as well as the anticipated loading for each PHB on all interfaces, which are subsequently being translated by Dynamic Resource Management into the appropriate scheduling parameters (e.g. priority, weight, rate limits) of the underlying PHB implementation. The values provided by Network

Dimensioning are not absolute but are in the form of a range, constituting directives for the function of the PHBs, while for LSPs they are in the form of multiple paths to enable multi-path load balancing. The exact PHB configuration values and the load distribution on the multiple paths are determined by Dynamic Resource Management and Dynamic Route Management respectively, based on the state of the network, but should always adhere to the Network Dimensioning directives.

Network Dimensioning runs periodically, getting the expected traffic per PHB in order to be able to compute the provisioning directives. The objectives are both traffic and resource-oriented. The former relate to the obligation towards customers, through the SLSes. These obligations induce a number of restrictions about the treatment of traffic. The resource-oriented objectives are related to the network operation, more specifically they are results of the high-level business policy that dictates the network should be used in an optimal fashion. The basic Network Dimensioning functionality is summarised in Fig. 3.

```
Input:
Network topology, link properties (capacity, propagation delay, PHBs)

Pre-processing:
Request traffic forecast, i.e. the potential traffic trunks
Obtain statistics for the performance of each PHB at each link
Determine the maximum allowable hop count per traffic trunk according
to the PHB statistics

Optimisation phase:
Start with an initial allocation (e.g. using the shortest path for
each traffic trunk)
Iteratively improve the solution such that for each traffic trunk
find a set of paths:
-The minimum bandwidth requirements of the traffic trunk are met
-The hop-count constraints K is met (delay/ loss requirements are met)
-The overall cost function is minimised

Post-processing:
Allocate any extra capacity to the resulted paths of each PHB according
to resource allocation policies
Sum the path requirements per link per PHB, give minimum (optimisation
phase) and maximum (post-processing phase) allocation directives to
Dynamic Resource Management
Give the appropriate multiple paths calculated in the optimisation
phase to Dynamic Route Management
```

Fig. 3. Network Dimensioning functionality

Congestion is the main cause of network performance degradation that influences both these traffic engineering objectives. Two are the main reasons behind congestion: either the demand exceeds the network capacity, or the traffic is not spread optimally, causing parts of the network to be over-utilised while others are under-utilised. The first reason can only be handled by adequate network planning, i.e. adding more physical resources. The latter can be handled by adequate bandwidth and routing management, and this is what Network Dimensioning is aiming for.

Network Dimensioning has as input the network topology (including link capacities), the estimated traffic matrix (expected traffic trunks, see Sect. 4.1), and resource-oriented policy directives. It will provide as an output a list of explicitly routed paths for each source destination included in the traffic matrix, and the capacity requirements for each PHB, and therefore physical queue for every interface of all the network nodes.

5.1 Network and Traffic Model

The network is modeled as a capacitated directed graph $G = (V, E)$, where V is the set of nodes and E the set of links. Each link $l \in E$ is specified by the pair $l = (v_{l,i}, v_{l,e}$ where $v_{l,i}$, $v_{l,e}$ are the nodes $\in V$, where traffic is entering (ingress) and exiting the link (egress) respectively.

With each unidirectional link l we associate the following parameters: the link physical capacity c_l, the link propagation delay d_l^{prop}, the set of PHBs H, supported by the link. For each PHB $h \in H$ we associate a bound d_l^h (deterministic or probabilistic depending on the PHB) on the maximum delay incurred by aggregate traffic entering link l and belonging to h. and a bound on the loss probability p_l^h of the aggregate traffic entering link l and belonging to h.

The basic traffic model of network provisioning is the traffic trunk (see Sect. 4). The set of all traffic trunks is denoted by T. Each trunk $t \in T$ is associated with a bandwidth requirement, B_t. The following information about each traffic trunk is available because of the QoS class definition (see Table 1):

- The PHB $h \in H$ of the traffic carried on the trunk.
- The bound D_t (deterministic or probabilistic depending on the PHB) on maximum end-to-end delay expected between the ingress and egress nodes. This might be the minimum value of the delay range of the QoS-class definition.
- The maximum end-to-end loss probability, P_t required between the ingress and egress nodes. Similarly to the previous case this might be the minimum value of the loss range of the QoS class definition.
- The bandwidth B_t requirement.

5.2 Cost Definition and Optimisation Objectives

We need to provide a set of routes for each traffic trunk. For each ingress-egress pair, these routes are implemented as LSPs at the routers. We also need to provide the amount of bandwidth each route is expected to carry, and the amount of bandwidth that is to be allocated to the PHBs at the various interfaces in the network.

The *primary objective* of such an allocation is to ensure that the requirements of each traffic trunk are met as long as the traffic carried by each trunk is at its specified minimum bandwidth. This objective ensures that our SLS requirements are met. The objective is to provide a feasible solution (i.e., routes and route bandwidths respecting the link capacities) that satisfies the SLSes delay and loss constraints.

However, with the possible exception of heavily loaded conditions, there will generally be multiple feasible solutions. Hence, the design objectives can be further refined to incorporate other requirements such as:

(a) avoid overloading parts of the network while other parts are underloaded. This way, spare bandwidth is available at various parts of the network to accommodate unpredictable traffic requests. In addition, in case of link failures, smaller amounts of traffic will be disrupted and will need to be rerouted, and
(b) provide overall low network cost (load).

The last two requirements do not lead to the same optimisation objective. In any case, in order to make the last two requirements more concrete, the notion of *load* has to be quantified. Various definitions are possible. In general, the load (or cost) on a given link is an increasing function of the amount of traffic the link carries. This function may refer to link utilisation or may express an average delay, or loss probability on the link. In the context of this work, the QoS seen by the traffic using the different PHBs varies, so the link load induced by the traffic of each PHB may vary. This leads us to the following general form of the cost of link l.

Let x_l^h denote the capacity demand (flow) for PHB $h \in H$ satisfied by link l. The link cost induced by the load on PHB h is a convex function, $f_l^h(x_l^h)$, increasing in x_l^h. The total link cost per link is defined as:

$$F_l(\bar{x}_l) = \sum_{h \in H} f_l^h(x_l^h) \tag{1}$$

where $\bar{x}_l$ is the vector of demands for all PHBs of link l.

Provided that appropriate buffers have been provided at each router and the scheduling policy has been defined, then $f_l^h(x_l^h)$ may specify the *equivalent capacity* needed by PHB h on link l in order to satisfy the loss probability associated with that PHB. Hence, the total cost per link is the total equivalent capacity allocated on link l. Note that with this approach the link costs are very naturally defined. The drawbacks are: 1) The cost definition depends on the PHB implementation at the routers, 2) The cost functions may not be known, or may be too complex. Hence approximate cost functions must be used. An example may be linear function of x_l^h, $f_l^h(x_l^h) = a_l^h x_l^h$, where a_l^h is a constant. Other examples may be found in [42].

We express objectives (a) and (b) formally as follows:

(a) avoid overloading parts of the network, i.e.

$$\text{minimise } \max_{l \in E} F_l(\bar{x}_l) \tag{2}$$

(b) minimize overall network cost, i.e.

$$\text{minimise } \sum_{l \in E} F_l(\bar{x}_l) \tag{3}$$

We have proposed [42] a new objective function that compromises between the previous two. More specifically,

$$\text{minimise } \sum_{l \in E} [F_l(\bar{x}_l)]^n, \quad n \in [1, \infty) \tag{4}$$

When $n = 1$ the objective defined in (4) reduces to the one in (3). As n increases the objective defined by (4) increasingly favours solutions which adhere to the objective defined by (2). When $n \to \infty$, it can be shown that (4) reduces to (2), since the factor with the maximum F_l will grow much faster than the others, and thus it will be increasingly the most important factor of the sum in (4).

Equation (4) uses a non-linear combination of the cost functions (1), thus the resulting optimisation problem is of non-linear nature even if $f_l^h(x_l^h)$ is a linear cost function for each link for each PHB. The resulting optimisation problem is a network flow [20] problem and considering the non-linearity of the objective function (4), the optimal solution can be based on the general gradient projection method [43]. This is an iterative method, where we start from an initial feasible solution, and at each step we find the minimum first derivative of the cost function path and we shift part of the flow from the other paths to the new path, so that we improve our objective function. The details of how we use this procedure can be found in [42].

Delay and Loss Constraints. Each traffic trunk is associated with an end-to-end delay and loss probability, constraint of the traffic belonging to the trunk. Hence, the trunk routes must be designed so that these two QoS constraints are satisfied. Delay and loss constraints are both on additive path costs under specific link costs. However, the problem of finding routes satisfying these constraints is in general NP-complete [31], [44]. Given that this is only part of the problem we need to address, the problem in its generality is rather complex.

Fortunately, we can make a reasonable simplification. Usually the measured loss probabilities and delay for the same PHB on different nodes (routers) are of similar order. Therefore we use the maximum delay, and maximum loss probability of a particular PHB over the whole network, in order to translate the delay and loss constraints to an upper bound on the number of hops along a route. During network operation, the average delay and loss at each link/PHB are monitored and the maximum respective values are used in the next provisioning period. This is relatively conservative and assuming the network is not overloaded, the delay and loss constraints of the traffic trunks should be met.

Note that under normal operating conditions, the delay and loss at each link/PHB for "premium trunks" should be low, assuming correct network dimensioning and admission control functionality at the ingress nodes. Problems with dimensioning or admission control may cause increased delay and loss, and this will be realised by network monitoring and subsequently dimensioning, triggering in the short term different (stricter) operating policies and in the longer term different network planning with more physical resources.

5.3 Simulation Results

In the simulation results we present in this section we set the initial feasible solution (step 0) of the iterative procedure of the solution [42] to be the same as if the traffic trunks were to be routed with a shortest path first (SPF) algorithm. That corresponds to the case that all the traffic of a particular class from an ingress to an egress will be routed through the same shortest path according to some weight (routing metric). If there exist more than one such shortest paths, the load is distributed equally among them. The metric we are using for the SPF algorithm is set to be inversely proportional to the physical link capacity. The scenario we are using for step 0 described above is the norm in today's operational networks. The experiments shown in this section correspond to only one forecasting period, i.e. one run of the provisioning algorithm.

We define as total throughput of a network the sum of the capacities of the first-hop links of all the edge nodes (see Fig. 4). This is actually an upper bound of the throughput and in reality it is a much greater than the real total throughput a network can handle. In our experiments we used 70% load of the total throughput of the network, as the highly loaded condition, and a 40% load as a medium loaded one.

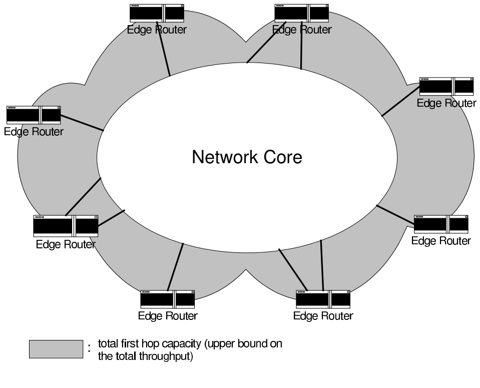

Fig. 4. Illustration of the total throughput as the sum of first hop links

Figure 5 shows the link loads for random 34-link topologies [45], at the first step and after the algorithm has run. It is clear that at step 0 solution (dark bars), which corresponds to the SPF with equal cost multi-path distribution enabled, parts of the network are over utilised while some links have no traffic at all. The final step (pale dry bars), which corresponds to the final output of our provisioning algorithm, balances the traffic over the whole network. Note that the largest over utilised link at step 0 is not necessarily the largest at the final solution.

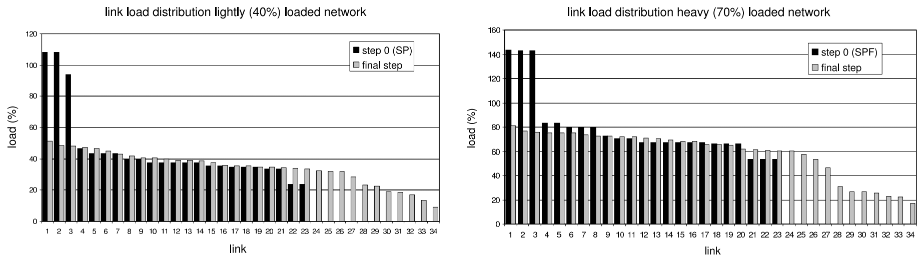

Fig. 5. Link load distribution of after the first and the last step

Figure 6 shows the mean and standard deviation of the link loads after each iteration of the algorithm. As we can see the load becomes more balanced over the network after each iteration (standard deviation reduces). We run those experiments with the exponent n, see (4), equal to 2. This value compromises between minimising the total (sum) of link costs and minimising the maximum link load. These two objectives generally lead to different solutions, with the first favouring path solutions with the least number of links while the other does not care about the number of links but only for the maximum link load (and therefore the deviation from the mean). We can observe the effect of these different objectives at the various ups and downs over the various steps of the algorithm.

The same behaviour observed with the network is also observed with larger random and transit-stub topologies [45]. We have experimented with large topologies up to 300 nodes under various loading scenarios and we observed the same behaviour as above. For the the details on more complex scenarios with large topologies and results on QoS constraints we refer the reader to reference [42].

Finally, as far as the the average running times of the various experiments conducted are concerned, we observed that even for quite large networks the running times were acceptable. For example for 300-node networks with more than 2000 links, for medium load the running time was on average about 17 minutes, and for high load about 25 minutes on a standard PC with 1GHz processor. These times are perfectly acceptable taking into account the timescale of the provisioning system operation (see Sect. 4.2).

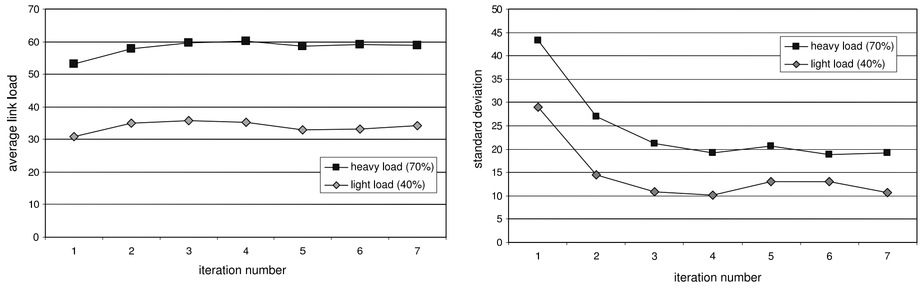

Fig. 6. Average and standard deviation of link load per iteration

6 Policy Extensions to Intra-domain Traffic Engineering

Policy-based Management has been the subject of extensive research over the last decade. Policies are seen as a way to guide the behaviour of a network or distributed system through high-level, declarative directives. The IETF has been investigating policies as a means for managing IP-based multi-service networks, focusing more on the specification of protocols (e.g. COPS) and the object-oriented information models for representing policies. We view policy-based management as a means of extending the functionality of management systems dynamically, in conjunction with pre-existing *hard-wired* logic [4]. Policies are defined in a high-level declarative manner and are mapped to low-level system parameters and functions, while the system intelligence can be dynamically modified added and removed by manipulating policies. Inconsistencies in policy-based systems are quite likely since management logic is dynamically being added, changed or removed without the rigid analysis, design, implementation, testing and deployment cycle of "hard-wired" long-term logic. Conflict detection and resolution is required in order to avoid or recover from such inconsistencies.

In the architecture shown in Fig. 1, Network Dimensioning besides providing long- term guidelines for sharing the network resources, it can also be policy influenced so that its behaviour can be modified dynamically at run-time reflecting high-level, business objectives. The critical issue for designing a policy-enabled resource management system is to specify the parameters influenced by the enforcement of a policy that will result in different allocation of resources in terms of business decisions. These policies that are in fact management logic, are not hard-wired in the system but are downloaded on the fly while the system is operating.

We extended [36], [46] the traffic engineering system shown in Fig. 1 to be able to drive its behaviour through policies. The resulting extended system architecture is depicted in Fig. 7. The Policy extensions include components such as the Policy Management Tool, Policy Repository, and the Policy Consumers.

A single Policy Management Tool exists for providing a policy creation environment to the administrator where policies are defined in a high-level declarative language and after validation and static conflict detection tests, they are translated into object-oriented representation (information objects) and stored

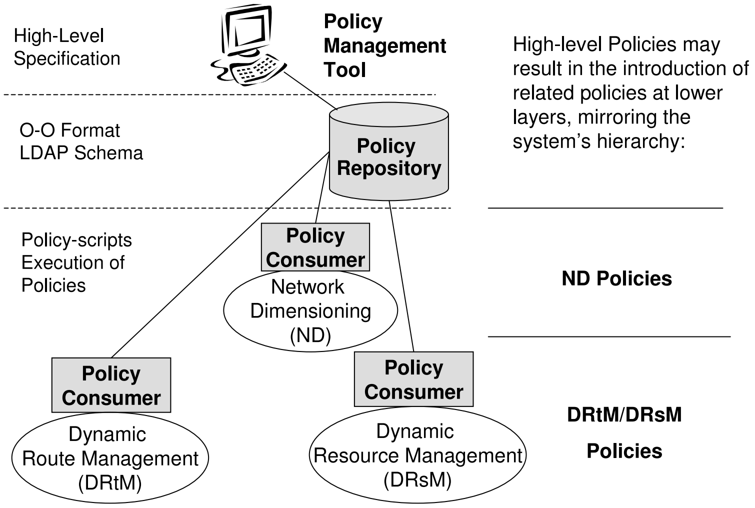

Fig. 7. Policy-driven Traffic Engineering Architecture

in a repository. The Policy Repository is a logically centralised component but may be physically distributed since the technology for implementing this component is the LDAP (Lightweight Directory Access Protocol) Directory. After the policies are stored, activation information may be passed to the responsible Policy Consumer in order to retrieve and enforce them. The Policy Consumer can be seen as a co-located Policy Decision Point (PDP) and Policy Enforcement Point (PEP) with regards to the IETF Policy Framework [47].

In Fig. 7 the representation of the policies at every level of the framework is also depicted, showing that every policy is going through two stages of translation/refinement in its life-cycle in order to be enforced. Starting from the high-level specification to the object-oriented format (LDAP objects) and then from the LDAP objects to a script that is interpreted on the fly, complementing this way conceptually the management intelligence of the above layer in the hierarchy.

For example, a policy enforced on the Dynamic Resource Management is actually enhanced management logic that conceptually belongs to the Network Dimensioning layer of our system. Although policies may be introduced at every layer of the system, higher-level policies may possibly result in the introduction of related policies at lower levels, forming a policy hierarchy mirroring the management system's hierarchy. This means that a policy applied to a hierarchical system might pass through another stage of translation and refinement that will generate the policies that are enforced in the lower levels of the system. It is questionable if the automation of this process is feasible without human intervention. A more detailed discussion on policy-based hierarchical management systems can be found in [48].

In the rest we will focus on the policies we identified for the specific domain defined by the functionality of Network Dimensioning as described in see Sect. 5.2.

6.1 Network Dimensioning Policies

We identify two categories of policies, *initialisation* and *resource provisioning* policies. The initialisation are policies that result in configuring with initial values the variables, which are essential for the functionality of Network Dimensioning and do not depend on any state but just reflect decisions of the policy administrator. The second category, the resource provisioning policies, are the policies that influence the way Network Dimensioning calculates the capacity allocation and the path creation configuration of the network. Such policies are those that their execution is based on the input from the traffic forecast module and on the resulting configuration of the network.

Since dimensioning is triggered mainly periodically, the policy administrator should specify this period. The priority of this policy should be specified in order not to cause any inconsistencies when re-dimensioning is triggered by notifications sent from the dynamic control parts of the system, that is when Dynamic Route Management and Dynamic Resource Management are unable to perform an adaptation of the network with the current configuration.

Another parameter that should be initialised by the policy system is the cost-function $f_l^h(x_l^h)$ used by the Network Dimensioning algorithm (see Sect. 5.2). The administrator should be able either to choose between a number of pre-specified cost functions and/or setting values to parameters in the desired function. For example, the approximate cost function used by Network Dimensioning could be linear function of the bandwidth allocated to a PHB, e.g. $f_l^h(x_l^h) = a_l^h x_l^h$, where a_l^h is a constant and x_l^h is the bandwidth allocated to PHB $h \in H$ on link l. In this case the initial value given to a_l^h could be configurable, and thus it could be specified by the policy administrator. Depending on the importance of a particular link l and of a particular PHB h, the constant can be increased and thus increasing the cost of using that PHB on this link.

Another constraint parameter which can be configurable by policies is the maximum number of alternative paths that Network Dimensioning defines for every traffic trunk for the purpose of load balancing. Finally, the exponent n of the objective function, as defined in (4), is another parameter that is specified by policies allowing the administrator to choose the relative merit between the two optimisation objectives, i.e. achieve low overall cost and avoid overloading parts only of the network.

Resource provisioning policies are more advanced compared to initialisation policies since their enforcement is not a simple parameter setting but requires the invocation of some logic. The policy administrator should be able to specify the amount of network resources (giving a minimum, maximum or a range) that should be allocated to each PHB. This will cause Network Dimensioning to take into account this policy when calculating the new configuration for this PHB. More specifically, Network Dimensioning should allocate resources in a way that does not violate the policy and then calculate the configuration taking into account the remaining resources.

A more flexible option is for the policy administrator to indicate how the resources should be shared in specific (critical) links. After the optimisation phase ends, Network Dimensioning enters a post-processing stage (see Fig. 3) where it will try to assign the residual physical capacity to the various PHBs. The way this distribution of spare capacity is done is left to be defined by policies that indicate whether it should be done proportionally to the way resources are already allocated or in a max-min fair way or it can be defined explicitly for every PHB. Another related policy is to specify the way the capacity allocated to each PHB should be reduced because the link capacity is not enough to satisfy the predicted requirements as given in the Traffic Matrix.

Network Dimensioning translates the delay and loss requirements on an upper bound on the number of hops per route, the way this translation is done can also be influenced by policy rules. For example, the safest approach to satisfy the traffic trunk requirements would be to assume that every link and node belonging to the route induces a delay equal to the maximum delay induced by an interface along the route. So, this policy rule will allow the administrator to decide if the maximum, average or some percentile of the actual measured delay or loss induced by an interface along the route should be used to derive the hop count constraint.

Policies that allow the administrator for a particular reason to explicitly specify an LSP that a traffic trunk should follow can also be defined. Of course, the resources for the administratively set LSP should be excluded from the available resources before the Dimensioning algorithm starts.

6.2 Policy Enforcement Example

The policy rule example concerns the effect of the cost function exponent n, see (4) in Sect. 5.2, on the capacity allocation of the network. As we discussed in Sect. 5.2 when we increase the cost function exponent n the optimisation objective that avoids overloading parts of the network is favoured compared to achieving overall low cost.

If it is required to keep the load of every link below a certain point then the administrator will have enter the appropriate policy in the Policy Management Tool using our proprietary policy language, which is then translated in LDAP objects according to an LDAP schema based on the Policy Core LDAP Schema [49] and stored in the Policy Repository. The syntax of our language as well as the extension to the Policy Core Information Model [50] with specific classes that reflect the policies described in the previous section are presented in [46]. The policy is entered with the following syntax:

```
if maxLinkLoad > 80% then decrease maxLinkLoad          (1)
```

After this rule is correctly translated and stored in the repository, the Policy Management Tool notifies the Policy Consumer which is associated with

Network Dimensioning, that a new policy rule is added in the repository. The Policy Consumer then retrieves all the associated objects with this policy rule. From these objects the consumer generates code that is interpreted and executed representing the logic added in our system by the new policy rule.

The pseudo-code produced by the Policy Consumer for policy (1) is shown in Fig. 8. The produced code will start by setting the exponent n to 1 and then will run the iterative optimisation algorithm, see `Optimisation phase` in Fig. 3. Then it will check if the value of `maxLinkLoad`, which represents the utilisation of the link with the maximum load as resulted from the optimisation algorithm, is less than the required value defined by the policy rule. If `maxLinkLoad` is above the aforementioned value, it will increase n an it will run the optimisation algorithm again until the policy is enforced. Note that as n increases the algorithm favours longer paths and thus we may not reach the desired solution as far as the hop-count constraint is concerned, but in any case if there is a feasible solution for enforcing the policy it will be found.

```
maxLinkLoad: maximum link load
 utilisation after optimisation
n=1: cost function exponent

run_optimisation_algorithm  n
while ( maxLinkLoad > 80 )
  n = n+1
  run_optimisation_algorithm n
```

Fig. 8. Pseudo-code produced when enforcing policy (1)

Figure 9 plots the maximum link load utilisation against the exponent of the objective function. The policy objective is achieved when $n = 4$, thus the enforcement of the policy rule caused the optimisation algorithm to run for 4 times until the maximum link load utilisation at the final step drops below 80%.

7 Interdomain Routing with BGP

Internet routing is handled by two distinct protocols with different objectives. Inside a single domain, link-state intradomain protocols such as OSPF or IS-IS distribute the entire network topology to all routers and select the shortest path according to a metric chosen by the network administrator. Across interdomain boundaries, the interdomain routing protocol is used to distribute reachability information and to select the best route to each destination according to the policies specified by each domain administrator. For scalability and business reasons, the interdomain routing protocol is only aware of the interconnections between distinct domains, it does not know any information about the structure of each domain.

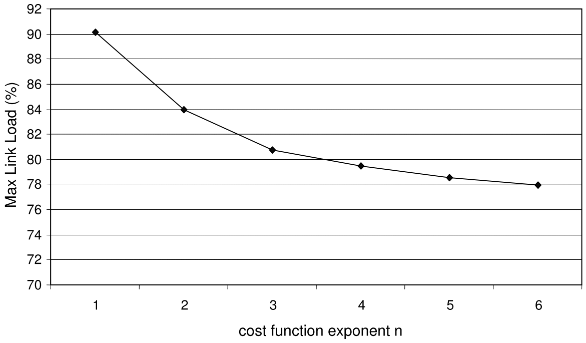

Fig. 9. Effect of the cost function exponent on the maximum link load utilisation

7.1 BGP Basics

The current de facto standard interdomain routing protocol is the Border Gateway Protocol (BGP) [51, 52]. In the BGP terminology, domains are called Autonomous Systems (AS) since these are usually managed by different independent companies. BGP is a *path-vector protocol* that works by sending *route advertisements*. A route advertisement indicates the reachability of a network which is a set of contiguous IP addresses represented by a *network address* and a *network mask* called a prefix. For instance, `192.168.0.0/24` represents a block of 256 addresses between `192.168.0.0` and `192.168.0.255`. A BGP router will advertise a route to a network because this network belongs to the same AS or because a route advertisement for this network was received from another AS. If a router of `ASx` sends a route advertisement for network N to a router of `ASy`, this implies that `ASx` accepts to forward IP packets with destination N on behalf of `ASy`.

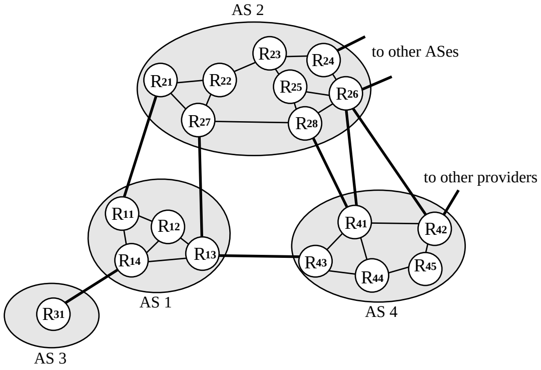

Fig. 10. A simple Internet

A route advertisement is mainly composed of the prefix of the network and the `next-hop` which is the IP address of the router that must be used to reach this network. A route advertisement also contains the `AS-path` attribute which contains the list of all the transit AS that must be used to reach the announced network. The `AS-path` has two important functions in BGP. First, it is used to detect routing loops. A BGP router will ignore a received route advertisement with an `AS-path` that already contains its AS number. Second, the length of the `AS-path` can be considered as a kind of route metric. A route with a shorter `AS-path` will usually be considered better than a route with a longer one.

Besides the `AS-Path`, a route advertisement may also contain several optional attributes such as `local-pref`, `multi-exit-discriminator` (`med`) or `communities` [51, 52].

7.2 Route Filtering

Inside a single domain, all routers are considered as "equal" and the intradomain routing protocol announces all known paths to all routers. In contrast, in the global Internet, all ASes do not play the same role and an AS will seldom agree to provide a transit service for all its neighbour ASes toward all destinations. Therefore, BGP allows a router to be selective in the route advertisements that it sends to neighbour eBGP routers. To better understand the operation of BGP, it is useful to consider a simplified view of a BGP router as shown in Figure 11.

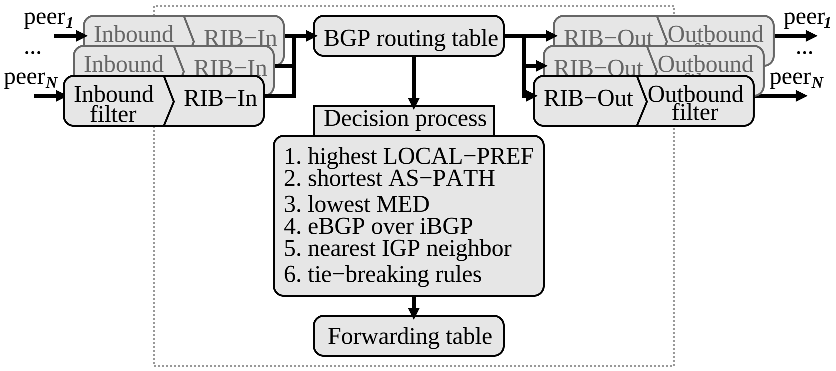

Fig. 11. Simplified operation of a BGP router.

A BGP router processes and generates route advertisements as follows. First, the administrator specifies, for each BGP peer, an input filter (Figure 11, left) that is used to select the acceptable advertisements. For example, a BGP router could only select the advertisements with an `AS-Path` containing a set of trusted ASes. Once a route advertisement has been accepted by the input filter, it is placed in the BGP routing table, possibly after having updated some of its attributes. The BGP routing table thus contains all the acceptable routes received from the BGP neighbours.

Second, on the basis of the BGP routing table, the BGP decision process (Figure 11, center) will select the best route toward each known prefix. Based on the `next-hop` of this best route and on the intradomain routing table, the router will install a route toward this network inside its forwarding table. This table is then looked up for each received packet and indicates the outgoing interface which must be used to reach the packet's destination.

Third, the BGP router will use its output filters (Figure 11, right) to select among the best routes in the BGP routing table the routes that will be advertised to each BGP peer. At most one route will be advertised for each distinct reachable prefix. The BGP router will assemble and send the corresponding route advertisements after a possible update of some of their attributes.

The input and output filters used in combination with the BGP decision process are the key mechanisms that allow a network administrator to support within BGP the business relationships between two ASes. Many types of business relationships can be supported by BGP. Two of the most common relationships are the customer-to-provider and the peer-to-peer relationships [53]. With the *customer-to-provider* relationship, a customer AS pays to utilise a link connected to its provider. This relationship is the origin of most of the interdomain cost of an AS. A stub AS usually tries to maintain at least two of these links for performance and redundancy reasons [53]. In addition, larger ASes typically try to obtain *peer-to-peer* relationships with other ASes and then share the cost of the link with the other AS. Negotiating the establishment of those *peer-to-peer* relationships is often a complicated process since technical and economical factors, as stated in [54], need to be taken into account.

To understand how these two relationships are supported by BGP, consider Figure 10. If `AS3` is `AS1`'s customer, then `AS3` will configure its BGP router to announce its routes to `AS1`. `AS1` will accept these routes and announce them to its peer (`AS4`) and upstream provider (`AS2`). `AS1` will also announce to `AS3` all the routes it receives from `AS2` and `AS4`. If `AS1` and `AS4` have a peer-to-peer relationship on the link between R_{13} and R_{43}, then router R_{13} will only announce on this link the internal routes of `AS1` and the routes received from `AS1`'s customer (i.e. `AS3`). The routes received from `AS2` will be filtered and thus not announced on the $R_{13} - R_{43}$ link by router R_{13}. Due to this filtering, `AS1` will not carry traffic from `AS4` toward `AS2`.

7.3 Decision Process

A BGP router receives from each of its peers one route toward each destination network. The BGP router must then identify the best route among this set of routes by relying on a set of criteria known as the *Decision Process*. Most BGP routers apply a decision process similar in principle to the one shown in Figure 11. The set of routes with the same prefix are analysed by the criteria in the order indicated in Figure 11. These criteria act as filters and the N^{th} criterion is only evaluated if more than one route has passed the $N-1^{th}$ criterion. It should be noted that most BGP implementations allow the network administrator to optionally disable some of the criteria of the BGP decision process.

In most BGP implementations, the set of criteria through which the router goes to select a best route toward a given destination is similar to what follows. First, the router checks that the routes received from its peers have a reachable `next-hop`, meaning that the IP routing table must contain a route toward this `next-hop`. If more than one route with a reachable next hop exists the router will then use preferences configured by the router administrator. Such preferences may be defined locally to a router with the `weight` parameter or shared over iBGP (interior BGP) sessions with the `local-pref` attribute. The router keeps routes with the highest `weight` and then routes with the highest `local-pref`. If after this criterion more than one route remain, the length of the `AS-Path` which acts as the BGP metric is used to compare routes. The length of the `AS-Path` is seen as a measure of the quality of the route and one usually expect that the route with the shortest `AS-Path` is the best.

If at this point the decision process has not yet identified the best route toward the given destination, that means that it has to select one among a set of equal quality routes. The remaining criteria were added for this purpose. The `multi-exit-discriminator` or `med` can be used to compare routes which were received from different routers of the same AS. The route with the lowest `med` is preferred. This criterion is not always enabled because the decision process can be influenced by the remote peers which set the value of the `med`. After the `med`, the decision process prefers routes learned over an eBGP session to routes learned over an iBGP session. The router gives then the preference to routes that can be reached by the closest BGP next hop. If after all these criteria, there is still more than one candidate route, tie-breaking rules are applied. Usual criteria are to keep the oldest route (this minimises route-flapping) or to prefer the route learned from the router with the lowest ID.

8 Characteristics of Interdomain Traffic

To obtain a better understanding of the characteristics of interdomain traffic, we have relied on Netflow [55] traces of two different ISPs. Netflow is a traffic monitoring facility supported by Cisco routers. When enabled, the router regularly transmits some information about all layer-4 flows (TCP connections and UDP flows) that passed through it to a close-by monitoring station. With Netflow, the monitoring station knows the starting and ending timestamps of all layer-4 flows as well as the flow volume (in bytes and packets) and the transport protocol and port numbers. Netflow is often used for billing purposes or by ISPs that need to better understand the traffic inside their network. Compared to the traditional packet-level traces that are often analysed, Netflow has the advantage of being able to monitor multiple links during long periods of time. The main drawback of Netflow is that it does not capture the very short-term variations of the traffic, but this is not a problem in our context of interdomain traffic engineering which tackles medium to long-term traffic variations.

The only characteristics common to both ISPs is that they do not offer transit service. Besides this, they serve very different customers and it can be expected that these customers have different requirements on the network. Due to technical reasons, it was unfortunately impossible to obtain traces from the two studied ISPs covering the same period of time.

The first trace was collected in December 1999 and covers 6 successive days of all the interdomain traffic received by BELNET. BELNET is the ISP that provides connectivity for the research and education institutions located in Belgium. At that time, BELNET was composed of a 34 Mbps star-shaped backbone linking the major universities. Its interdomain connectivity was mainly provided through 34 and 45 Mbps links to the transit service from two commercial ISPs. In addition, BELNET had a 45 Mbps link to the European research network, TEN-155, and was present at the BNIX and AMS-IX interconnection points with a total of 63 peering agreements in operation. Although some universities provided dialup access for their students, the typical BELNET user had a 10 Mbps access link to the BELNET network through their university LAN. During the 6 days period, BELNET received 2.1 terabytes of data. BELNET is representative of research networks and could also be representative of an ISP providing services to high bandwidth users with cable modem or ADSL. We will call BELNET the research ISP in the remainder of this section. The mean traffic over the six days period was slightly larger than 32 Mbps, with a one-minute maximum peak at 126 Mbps and a standard deviation of 21 Mbps. The trace begins around 1 AM on a Sunday and finishes six days later around 1 AM also.

The second trace was collected in April 2001 and covers a little less than 5 consecutive days of all the interdomain traffic received by Yucom. Yucom is a commercial ISP that provides Internet access to dialup users through regular modem pools. At that time, the interdomain connectivity of Yucom was mainly provided through high bandwidth links to two transit ISPs. In addition to this transit service, Yucom was also present at the BNIX interconnection point with 15 peering agreements in operation. During the five days of the trace, Yucom received 1.1 terabytes of data. Yucom is representative of an ISP composed of low bandwidth users. We will call Yucom the dialup ISP in the remainder of this section. The trace starts around $8:30$ AM on a Tuesday and finishes almost 5 days later at midnight. The total traffic also exhibits a daily periodicity with peak hours located during the evening, in accordance with the typical user profile, a dialup user. It had an average total traffic of about 23 Mbps over the measurements, with a one-minute maximum peak at 64 Mbps and a standard deviation of 12 Mbps.

Before analysing the collected traffic statistics, it is useful to have a first look at the BGP table of the studied ISPs. In this section, we assume that the BGP table of both ISPs was stable during the period of the measurements and perform all our analysis based on a single BGP table for each ISP. Using a single BGP table for each ISP is an approximation but since we rely on the BGP table of the studied ISPs our analysis is more precise than other studies [56, 57] that relied on a BGP routing table collected at a different place and time than the packet traces studied in these papers.

The routing table of the dialup ISP contained 102345 active prefixes, covering about 26 % of the total IPv4 address space. This coverage of the total IPv4 address space is similar for the research ISP, with about 24 %, but for 68609 prefixes only. Between late 1999 and mid-2001, 30 % more prefixes are necessary to cover a similar percentage of the IPv4 address space. This has already been analysed elsewhere [58]. Although having different numbers of prefixes in their BGP routing table, the two ISPs cover a similar percentage of the IPv4 address space. This is explained by the average address span per prefix for each ISP, which is about 11000 IP addresses for the dialup ISP and about 15200 addresses for the research ISP. The dialup ISP knew 10560 distinct AS while the research ISP 6298. This difference is mainly due to the large increase in the number of multi-homed sites during the last few years [58]. The average AS path length was 4.2 AS hops for the dialup ISP and 4.5 AS hops for the research ISP.

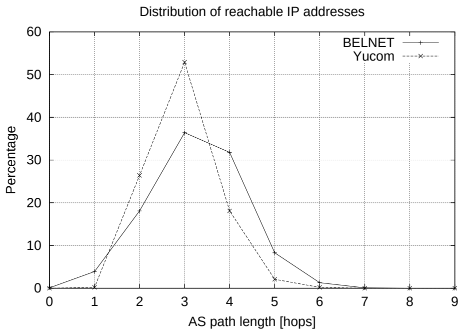

Fig. 12. Distribution of reachable IP addresses.

Figure 12 compares the distribution of the reachable IP addresses for the BGP routing tables of the research ISP and the dialup ISP. The main difference between the two is the more compact distribution for the dialup ISP around a distance of 3 AS hops. The research ISP has its reachable address space more spread over distances of 3 and 4 AS hops. The first 3 AS hops for the dialup ISP provide almost 80 % of the reachable address space while only about 60 % for the research ISP. The difference between the distribution of the reachable IP prefixes seen from the two ISPs is probably due mostly to the 16 months delay between the two traces.

To understand the topological variability of interdomain traffic and the possible levels of aggregation, we consider in this section two different types of interdomain flows. Generally, a flow is defined as a set of IP packets that share a common characteristic. For example, a micro-flow is usually defined as the set of IP packets that belong to the same TCP connection, i.e. the IP packets that share the same source address, destination address, IP protocol field, source and destination ports. In this section, we consider two different types of network-layer flows. A *prefix flow* is the set of IP packets whose source addresses belong to a given network prefix as seen from the BGP table of the studied ISP. An *AS flow* is defined as the set of IP packets whose source addresses belong to a given AS as seen from the BGP table of the studied ISP. We do not use explicitly the term "flow" to designate traffic coming from a traffic source, but rather the terms "prefix" and "AS" (or "source AS") to denote a *prefix flow* and *AS flow* respectively. Moreover we use the term *order statistics* throughout this paper to denote the traffic flows ordered by decreasing amount of total traffic sent during the all measurements.

Let us first study the amount of aggregation provided by the AS and prefix flows. Figure 13 shows the cumulative percentage of traffic for *order statistics* for prefixes and source AS. On this figure, we have thus ordered the prefixes and AS by decreasing order of the total amount of traffic sent by them over the measurements period, and we have computed their cumulative contribution to the total traffic over the measurements period. The x-axis uses a logarithmic scale to better show the low *order statistics.* Both ISPs seem to have a similar distribution for the most important interdomain traffic sources. The top 100 AS (resp. prefixes) capture 72 % of the total traffic (resp. 52 %) for the dialup ISP while a little less than 60 % (resp. a little more than 40 %) for the research ISP. 90 % of the total traffic is captured by 4.7 % of the AS and by 4.1 % of the prefixes for the dialup ISP. The research ISP required 9.8 % of the AS and 4.5 % of the prefixes to capture 90 % of the total traffic. These results are similar to the findings of earlier studies [59, 60] on the research Internet of the 1970s and the early 1990s. On the other hand, some AS and prefixes contribute to a very small fraction of the total traffic. For the dialup ISP, more than 4000 different AS contributed each to less than 1 megabyte of data during the measurement period and some AS only sent a single packet during this period. For the research ISP, 719 AS sent less than 1 megabyte of data during the six days measurement period.

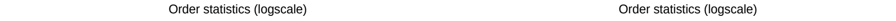

Fig. 13. Cumulative traffic distribution for traffic sources, research ISP (left) and dialup ISP (right).

Another interesting point to mention is that over the measurement period, the research ISP received IP packets from 5606 different AS and 35688 different network prefixes. This corresponds to 89 % of the AS present inside its routing table. Concerning the dialup ISP, it received IP packets from 7668 different AS and 35693 different network prefixes. This corresponds to 72.6 % of the AS present inside its routing table. These figures show that even relatively small ISPs receive traffic from a very large portion of the Internet during a one week period although some sources only send a few packets.

8.1 Interdomain Proximity of the Traffic

The amount of aggregation is not the only issue to be considered when studying interdomain traffic characteristics. Another important issue concerns the topological distribution of the traffic. By topological distribution, we mean the AS distance between the traffic sources and the studied ISP. This distance is important for two reasons. First, usually the performance of an Internet path decreases with the distance between the source and destination AS [61]. Second, if the distance between the source and the destination AS is large, it will be difficult for either the source or the destination to apply mechanisms to control the traffic flow in order to perform interdomain traffic engineering [1].

Another study of the topological distribution of the traffic [62] revealed that the studied ISPs only exchange a small fraction of their traffic with their direct peers (AS-hop distance on 1). Most of the packets are exchanged with ASes that are only a few AS hops away. For the BELNET trace, most of the traffic is produced by sources located 3 and 4 AS hops away while YUCOM mainly receives traffic from sources that are 2 and 3 AS hops away.

8.2 Distribution of the Interdomain Traffic

The previous section showed the amount of traffic generated by interdomain sources for each AS hop distance. Another concern for interdomain traffic engineering is how many sources send traffic at each AS hop distance. Figure 14 presents the cumulative traffic distribution for the top AS for each AS hop distance. In this figure, an AS is not seen as a traffic source from which a flow originates but also as an intermediate node through which a flow passes. This means that an AS located at an AS hop distance of n is seen as the source of the traffic it generates as well as of all the traffic it forwards when considering the AS_PATH information of the BGP routing table. This means that the traffic seen for all AS at an AS hop distance of n contains the traffic originating from all AS hop distances m with $m \geq n$. Because each AS hop distance does not contribute evenly to the total traffic, we have plotted the cumulative traffic percentage for every AS hop distance with respect to the total traffic seen during the measurements period, to show how many AS represent a large fraction of the traffic that crosses the interdomain topology at a given AS hop distance from the local ISP.

The rightmost part of teach curve of Figure 14 shows the uneven distribution of the total traffic among the different AS hop distances.

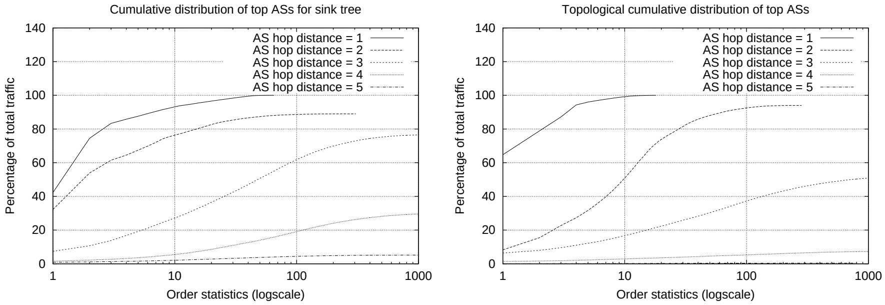

Fig. 14. Cumulative traffic distribution for sink tree, research ISP (left) and dialup ISP (right).

The most important AS at 1 AS hop carries 64 % of the total traffic in the case of the dialup ISP while 42 % for the research ISP. This difference is however lessened when considering the top 3 AS at an AS hop distance of 1, capturing 83 % and 87 % of the total traffic, for the research ISP and the dialup ISP respectively. This shows the predominance of a very small number of BGP peers that provide connectivity for almost all the interdomain traffic of the studied ISPs. At a distance of two AS hops, a few ASes also dominate the traffic with the top 10 carrying more than 77 % of total traffic for the research ISP while 54 % for the dialup ISP. Nevertheless, the traffic produced by AS at a distance of 2 or more AS hops corresponds to 89 % of the total traffic for the research ISP, and 94 % for the dialup ISP. Therefore, a very small fraction of the traffic comes from direct peers themselves. Subsequent distances in terms of AS hops require an increasingly important number of ASes to capture a large fraction of the traffic.

The first AS hop generates 11 % (resp. 6.1 %) of the total traffic, the second AS hop 12.4 % (resp. 42.1 %), the third 46.6 % (resp. 44.4 %), and the fourth 24.9 % (resp. 6.9 %) for the research ISP (resp. for the dialup ISP). The main difference between the two studied ISPs occurs at an AS hop distance of 2. The research ISP has its traffic for the first AS hops that is captured by very few ASes. The dialup ISP on the other hand requires a relatively large number of AS at a distance of 2 AS hops to account for an important fraction of the traffic. This should be compared with the routing table of the dialup ISP (Figure 12) where 52 % of the reachable IP addresses are located at 3 AS hops, and 26 % and 18 % at levels 2 and 4. This means that traffic is unevenly distributed between levels 2 and 4, with more traffic coming from level 2 in comparison to its reachable address space relatively to level 4.

9 BGP-based Interdomain Traffic Engineering

At the interdomain level, ASes have to face various sometimes conflicting issues. On one hand, the traffic is unevenly distributed because BGP seldom takes the right decision on its own and this can cause links to be unevenly loaded and congestion to occur. Moreover, depending on the type of its business, an AS will be more concerned by its incoming or outgoing traffic and thus will use the appropriate traffic engineering techniques. On the other hand, ASes try to maintain as much connections as they can with other ASes for performance and redundancy reasons. If an AS selects a single provider, then all its interdomain traffic will be sent and received from this provider and the only traffic engineering activity will be to balance the traffic if several physical links are used. However, in practice many ASes prefer, for both performance and economical reasons, to select at least two different upstream providers. Since this connectivity is expensive, another concern of ASes will often be to favour the cheapest links.

Moreover, an AS will want to optimise the way traffic enters or leaves its network, based on its business interests. Content-providers that host a lot of web or streaming servers and usually have several customer-to-provider relationships with transit ASes will try to optimise the way traffic leaves their networks. On the contrary, access-providers that serve small and medium enterprises, dialup or xDSL users typically wish to optimise how Internet traffic enters their networks. And finally, a transit AS will try to balance the traffic on the multiple links it has with its peers.

9.1 Control of the Outgoing Traffic

To control how the traffic leaves its network an AS must be able to choose which route will be used to reach a particular destination through its peers. Since an AS controls the decision process on its BGP routes, it can easily influence the selection of the best path. Before looking at the BGP-based techniques later, we first comment the results of an analysis of routing tables collected by Route-Views [63] which show that an hypothetic stub AS connected to two ISPs often receives two routes toward the same destination. The analysis also shows that the selection of a best route in this set is non-deterministic in many cases (because the lengths of the `AS-Path` are equal). Later in this subsection, we briefly describe two techniques that are frequently used to influence the way the traffic leaves the network.

Implications of the BGP decision process As shown earlier, the length of the `AS-Path` is used by BGP to rank routes. Some studies [64, 61] have indicated some correlation between the quality of a route and the length of its `AS-Path`. To evaluate the importance of `AS-Path` attribute in the selection of BGP routes, we have simulated the behaviour of a dual-homed stub ISP. For this, we relied on the BGP routing tables collected by route-views [63]. Route-views is a BGP router that maintains multi-hop BGP peering sessions with 20 different ISPs.

We used the BGP routing table recorded on 01 September 2002 at 00:38. We have extracted from the table the routes advertised by each peer of route-views and only consider the peers that advertise their full BGP routing table in this study and used a single BGP peer from each AS.

Based on the BGP routing tables announced by each AS, we have simulated the various possibilities of being dual-homed for a candidate ISP. For this purpose, we performed an experiment with three routers. We used three BGP routers and two BGP sessions. Two routers are used to simulate candidate providers and the third router simulates the multi-homed stub AS. Each of the two candidate AS advertises a BGP routing table from one of the providers. The BGP router of the stub AS runs *GNU Zebra 0.92a* [65] with a default configuration. This implies that this router selects the best route toward each destination advertised by the candidate ASes based on the normal BGP decision process. We modified *Zebra* to allow us collect statistics on the number of routes selected by each criteria of its decision process.

We used this setup to evaluate all possible pairs of upstream providers based on the route-views data. The first result of this evaluation concerns the size of the routing table of the stub ISP. On average, there were 107789 prefixes inside its routing table with a minimum of 95428 and a maximum of 112842. The upper line of figure 15 shows, for each candidate upstream provider, the average number of routes for the 19 experiments where this provider was considered together with another upstream provider. This figure shows that the average number of routes does not vary significantly.

The second element that we considered is the selection of the routes by the BGP decision process of the stub AS. Two cases are possible in our experiment. First, if a route toward a given prefix is only announced by a single candidate AS, this route will automatically be selected. In our experiment, between 87 and 96.5% of the routes received by the stub AS were advertised by both candidate upstream providers. The second line on figure 15 shows, for each candidate upstream provider, the average number of common prefixes between this provider and the other 19 providers. The difference between the routes advertised by different providers can be caused by several factors such as the differences in reachability of each provider and the utilisation of prefix-length filters by some AS as discussed in [66].

The second, and more interesting case to consider is when both candidate providers advertise a route toward each prefix. In this case, the BGP decision process of the stub AS needs to select the best route for each prefix. With the default configuration used by the router of our stub AS, it will first check the `AS-Path` of the received routes. If their `AS-Path` differ, the route with the shortest `AS-Path` will be selected. Otherwise, the tie-breaking rules will be used to select the best route. The bottom line of figure 15 shows, for each candidate upstream provider, the average number of routes from this provider that are selected on the basis of their shorter `AS-Path` by the BGP decision process of our stub AS. For example, concerning AS16150, this figure shows that on average, it advertises 5427 routes with a shorter `AS-Path` than other candidate upstream providers on

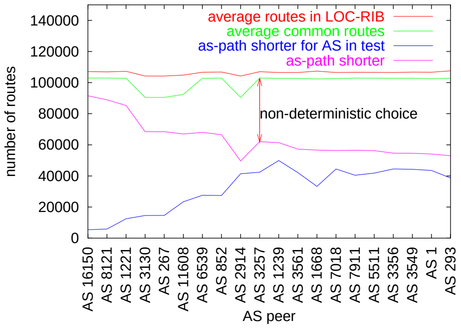

Fig. 15. Quality of the routes announced by an AS — Tests on the 20 peers from Route-Views

an average of 102945 routes in common. This is not surprising since this AS is a much smaller than that the tier-1 ISPs that appear on the right hand side of figure 15.

The second line starting from the bottom in figure 15 shows the average number of routes that were selected by the BGP decision process due to a shorter `AS-Path` in the 19 experiments that involved each candidate AS. For example, concerning AS16150, this line shows that on average, 86104 routes were chosen for their shorter `AS-path` when this provider was confronted with the other providers. The difference between the two bottom lines in figure 15 corresponds to the average number of routes received from the 19 other providers with a shorter `AS-Path` than the considered candidate AS. For AS16150, the other providers advertised on average 80677 routes with a shorter `AS-Path` than this provider. Finally, the difference between the number of common routes and the total average number of shorter routes, shows the average number of routes that were selected in a non-deterministic manner by using the tie-breaking rules of the BGP decision process. For the experiments with AS16150, only 11413 routes were chosen in such a non-deterministic manner.

If we consider a larger AS in figure 15 such as AS1, we find several interesting results. First, on average, AS1 advertises for 43532 prefixes a route with a shorter `AS-Path` than the routes to the same prefixes advertised by any of the other 19 studied ASes. Second, the difference between the bottom line and the line above shows that on average another upstream provider only advertises 10546 shorter routes than AS1. Finally, among all the pairs where AS1 was one of the candidate upstream providers, on average 48473 routes were selected by relying on the tie breaking rules of the BGP decision process. This means that on average 45% of the received routes have the same quality based on their `AS-Path`. A closer look at those common prefixes reveals that 50% of the common prefixes with an `AS-Path` length of three AS-hops are chosen in non deterministic manner. Furthermore, 26.5% of the routes chosen by the tie-breaking rules have an `AS-Path` length of

3 or 4. This indicates that the large ASes advertise short routes towards most destinations.

BGP-based techniques Two techniques are often used to control the flow of the outgoing traffic. The first technique that can be used by an AS to control its outgoing traffic is to rely on the `local-pref` attribute. This optional BGP attribute is only distributed inside an AS. It can be used to rank routes and is the first criterion used in the BGP decision process (Figure 11). For example, consider a stub AS with two links toward one upstream provider : a high bandwidth and a low bandwidth link. In this case, the BGP router of this AS could be configured to insert a low `local-pref` to routes learned via the low bandwidth link and a higher value to routes learned via the high bandwidth link. A similar situation can occur for a stub AS connected to a cheap and a more expensive upstream provider. In practice the manipulation of the `local-pref` attribute can also be based on passive or active measurements [67]. Recently, a few companies have implemented solutions [68] that allow multi-homed stub ASes and content-providers to engineer their outgoing interdomain traffic. These solutions usually measure the load on each interdomain link of the AS and some rely on active measurements to evaluate the performance of interdomain paths. Based on these measurements and some knowledge of the Internet topology (either obtained through a central server or from the BGP router to which they are attached), they attach appropriate values of the `local-pref` attribute to indicate which route should be considered as the best route by the BGP routers. We will evaluate the impact of `local-pref` by using simulations in section 9.5.

A second technique, often used by large transit ISPs, is to rely on the intradomain routing protocol to influence how a packet crosses the transit ISP. As shown in Figure 11, the BGP decision process will select the nearest IGP neighbour when comparing several equivalent routes received via iBGP. For example, consider in Figure 10 that router R_{27} receives one packet whose destination is R_{45}. The BGP decision process of router R_{27} will compare two routes toward R_{45}, one received via R_{28} and the other received via R_{26}. By selecting router R_{28} as the exit border router for this packet, `AS2` will ensure that this packet will consume as few resources as possible inside its own network. If a transit AS relies on a tuning of the weights of its intradomain routing protocol as described in [69], this tuning will indirectly influence its outgoing traffic.

9.2 Control of the Incoming Traffic

In contrast to the outgoing traffic, it is much more difficult to control the incoming traffic with BGP. Nevertheless access providers can utilise some techniques to influence how the interdomain traffic enters their AS. We first briefly describe these techniques, then we present the simulation model that we used to evaluate one of those techniques and discuss the simulation results in section 9.4.

The first method that can be used to control the traffic that enters an AS is to rely on selective advertisements and announce different route advertisements on different links. This method suffers from an important drawback: if a link fails, the prefixes that were only announced through the failed link will not be reachable anymore.

A variant of the selective advertisements is the advertisement of more specific prefixes. This technique relies on the fact that an IP router will always select in its forwarding table the most specific route for each packet (i.e. the matching route with the longest prefix). For example, if a forwarding table contains both a route toward `16.0.0.0/8` and a route toward `16.1.2.0/24`, then a packet whose destination is `16.1.2.200` would be forwarded along the second route. This fact can be used to control the incoming traffic by advertising a large aggregate on all links for fault-tolerance reasons and specific prefixes on some links. The advantage of this solution is that if a link fails, the less specific prefix remains available on the other link. Unfortunately, a widespread utilisation of this technique is responsible for a growth of the BGP routing tables. To reduce this growth, many large providers have implemented filters that reject advertisements for too long prefixes [66].

Another method consists in allowing an AS to indicate a ranking among the various route advertisements that it sends. Since the length of the `AS-Path` appears as the second criteria in the BGP decision process, a possible way to influence the selection of routes by a distant ASes is to artificially increase the length of the `AS-Path` of less preferable routes. This is typically done by inserting several times its own AS number in the `AS-Path`. Based on discussions with network operators, it appears that the number of times the own AS is prepended to achieve a given goal can only be found on a trial and error basis.

The last method to allow an AS to control its incoming traffic is to rely on the `multi-exit-discriminator` (MED) attribute. This optional attribute can only be used by an AS multi-connected to another AS to influence the link that should be used by the remote AS to send packets toward a specific destination. It should however be noted that the utilisation of the `MED` attribute is usually subject to a negotiation between the two peering ASes and some ASes do not accept to take the `MED` attribute into account in their decision process. Furthermore, the utilisation of this attribute may cause persistent oscillations [70].

9.3 Simulation Model

The first element of our simulation model is our simulation environment: Javasim [71]. Javasim is a scalable event-driven simulator developed by Hung-Ying Tyan and many others at Ohio-State University. Javasim is written in Java for portability reasons and contains realistic models of various Internet protocols. Although Javasim supports several routing protocols, it did not contain any BGP model. Instead of developing a BGP model from scratch, we choose to port[8] and enhance the BGP implementation developed by B. J. Premore [72] for SSFNet [73]. This model has been extensively validated and tested and has

[8] Our modifications to Javasim are available from `http://www.javasim.org`.

already been used for several simulation studies [74, 75]. We have enhanced it to better support the routing policies that are often used by ISPs as shown earlier.

The second element of our simulation model is the network itself. Since our goal is to evaluate the performance of interdomain traffic engineering, we need a realistic model of the Internet. To evaluate `AS-Path` prepending, we choose to build each AS as composed of a single router that advertises a single IP prefix. This router runs the BGP protocol and maintains BGP sessions with routers in neighbouring ASes. The second element that we needed to specify is the topology of the interdomain links.

An interdomain topology could be obtained from a snapshot of the current Internet, such as the one analysed in [53]. However, a drawback of this approach is that then it is difficult to perform simulations with various topologies to evaluate the impact of the topology on the results. A second method is to rely on a topology built by topology generators. Various topology generators have been proposed and evaluated in the last few years (see [76, 77] and the references therein). It is admitted that two classes of generators can be used: structural and degree-based generators. Structural generators attempt to reproduce the real Internet hierarchy (i.e. tiers, transit ASes and stubs) while degree-based generators approximate a specific property of the real topology, the node degree distribution. It has been shown in [76] that the Internet hierarchy can be better approximated with topologies produced by degree-based generators. Moreover, [77] indicates that degree-based generators are also better suited to approximate the structure of the Internet. Indeed, such generators implicitly create hierarchies closely related to the current Internet hierarchy.

We relied on a degree-based topology generator to produce the various Internet topologies used for the simulations. Our topologies have been generated with Brite [78] which is a highly configurable generator. One of its interesting features is the ability to produce topologies with ASes only, intended to simulate the interdomain level. Brite is able to rely on various mathematical models to generate a topology. We have chosen the Barabasi-Albert model [79] because it is degree-based. This model builds the topology sequentially by adding one AS at a time while relying on two simple principles[80]:

- *Growth*: each node that must be added to the topology is connected to m existing nodes (where m is a parameter of the generator).
- *Preferential Attachment*: when a new interdomain link is created, it connects the AS being added to an existing AS. This AS is selected with a probability which depends on the number of links already attached to each AS. This means that an AS with a lot of interdomain links will be attached to other ASes with a high probability.

A consequence of these two principles is that the ASes which are generated first (i.e. those with a low identifier) have a greater connectivity than the ASes generated last (i.e. those with a large identifier).

In our simulations we use an interdomain topology with two types of ASes. The *core* of the network is composed of a few hundred transit ASes. This core is generated by using Brite. Note that we do not consider hierarchy in the *core*. For this reason, all *core* ASes all advertise their full routing table to their neighbouring ASes and no routing policies have been defined for the *core* ASes.

In addition to the *core*, our topologies also contain a few hundred stub ASes. Those stub ASes are added to the topology generated by Brite by following a *preferential attachment* principle (the probability for an AS in the *core* to be connected to a stub is function of its current connectivity). Each stub has exactly two connections to two different transit ASes in the *core*. These connections represent *customer-to-provider* links where the stub is the customer and the ASes in the *core* are the providers. We configured BGP policies on the stub ASes to ensure that those ASes do not provide any transit. In the following, we call `lowID` provider (resp. `highID` provider), the provider of the considered stub AS with the lowest (resp. highest) AS number and `lowID` link (resp. `highID` link), the link that leads to this provider.

We have introduced the stub ASes in our network topologies for two reasons. First, they represented 85.6% of the number of ASes on the Internet in October 2002. Second, they will serve as measurement points to evaluate the impact of `AS-Path` prepending on the routes selected by the BGP decision process of each simulated router.

We have performed simulations with several topologies. Due to space limitations, we restrict our analysis to two representative topologies. The first topology is composed of a lightly connected *core* with 200 ASes. This topology was produced by Brite with the value of the m parameter set to 2. 400 dual-homed stub ASes were attached to the *core* ASes with preferential attachment. In total, when considering both the stub and the transit ASes, this topology contains 1594 interdomain links.

The second topology is composed of a dense *core* with 200 ASes. This *core* was produced by Brite with the value of the m parameter set to 4 to model a *core* composed of ASes with a higher connectivity. We have connected 200 dual-homed stub ASes with preferential attachment to this dense core. In total, this second topology contains 1980 interdomain links.

Due to the memory constraints we were not able to perform simulations with more than about 2500 BGP peering sessions. This is one order of magnitude less than the number of interdomain relations reported in [53], but more than one order of magnitude more links than in existing traffic engineering studies.

Another element that should be considered in such a model is the amount of traffic sent by each AS towards each remote AS. In section 8, we have shown that a small number of ASes were responsible for a large fraction of the traffic received by the two studied ISPs. However, those measurement are not sufficient to allow us to determine the behaviour of hundred different ASes and how their traffic would be distributed among the Internet. Developing such a model is outside the scope of this chapter and for the simulations described below, we will consider the interdomain paths between all AS pairs without considering the amount of traffic exchanged.

To evaluate the impact of this BGP traffic engineering technique, we use the stub ASes as measurement points. After a sufficient time to allow the BGP routes to converge, each router sends a special IP packet with the `record route` option toward each remote stub AS. The stub ASes collect the received IP packets and by analysing the `record route` option of each received packets, we can determine all the interdomain paths followed by IP packets. The analysis of all these interdomain paths allows us to study the impact of BGP traffic engineering techniques.

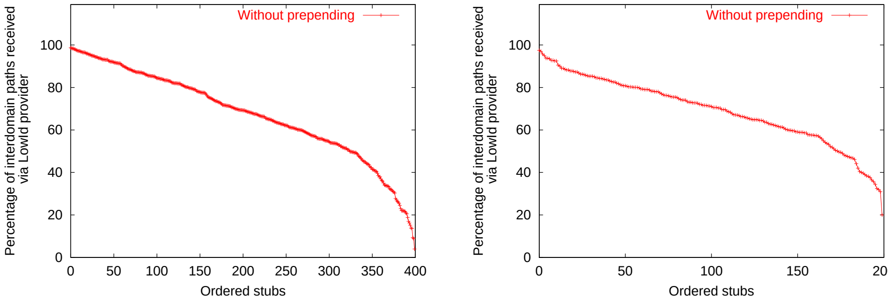

Fig. 16. Distribution of interdomain paths without prepending for lightly (left) connected and dense (right) core

For our first simulation, we configured the stub ASes to send their BGP announcements without any `AS-Path` prepending. Figure 16 shows, for each stub AS, the percentage of the interdomain paths that end on this stub AS and are received via its `lowId` provider. To plot this figure, we have ordered the stub ASes that appear on the x-axis in decreasing percentage of the interdomain paths received via their `lowID` provider. This ordering, determined for each topology, is used for all simulation results described in the remainder of this section.

Several points need to be mentioned concerning Figure 16. First, the distribution of the interdomain paths is not uniform. For both core networks, some stub ASes receive almost all their interdomain packets via one of their providers. The stub ASes that receive almost all their interdomain packets via their `lowID` provider are usually attached to a dense `lowID` provider and a `highID` provider with a very weak connectivity. For the stub ASes that receive almost all their interdomain packets via their `highID` provider, the reason is that this provider is closer to the core ASes with the higher connectivity than their `lowID` provider. Note that with the dense core this situation occurs less often.

A second point to be mentioned is that for the lightly connected core, about 66% of the stub ASes receive more than 60% of their interdomain packets via their `lowID` provider. This is due to the fact that, thanks to its connectivity, the `lowID` provider is, on average, closer to most destinations than the `highID` provider. For the dense core, results are similar: 72.5% of stub ASes receive more than 60% of their traffic through their `lowID` link.

9.4 Evaluation of `AS-Path` Prepending

As described earlier, `AS-Path` prepending can be used by an ISP to control the flow of its incoming traffic by announcing on some links routes with an artificially long `AS-Path`. Although this technique is used today in the Internet ([64] reports that `AS-Path` prepending affected 6.5 % of the BGP routes in November 2001), there has not been any analysis of its performance to the best of our knowledge.

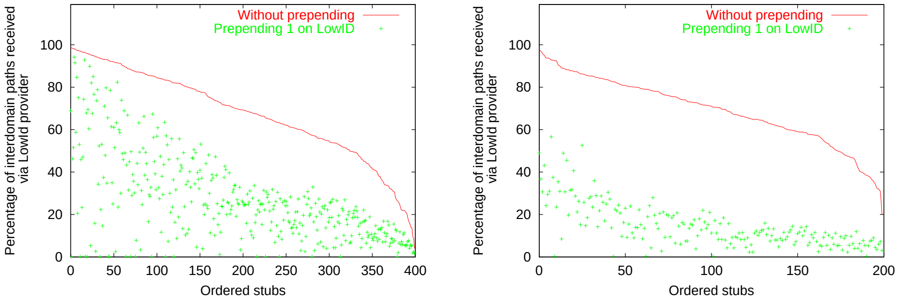

Fig. 17. Distribution of interdomain paths with prepending of 1 on the lowID link for lightly connected (left) and dense (right) cores

To evaluate the impact of `AS-Path` prepending we performed several simulations with the stub ASes configured to send prepended routes to one of their upstream providers. In all simulations, we configured all stub ASes in the same manner to ease comparisons. For our first simulation, each stub AS sent routes with its own AS number prepended once to its `lowID` provider and without prepending to its `highID` provider. In practice, a stub AS could use this prepending to better balance its traffic between its two upstream providers if it receives more traffic via its `lowID` provider.

Figure 17 shows the impact of this prepending on the distribution of the interdomain paths. In this figure, we show the distribution without prepending that was presented in Figure 16 as a reference and use the ordering from this figure to plot the distribution of the interdomain paths with prepending.

The analysis of the simulation with the lightly connected core (Figure 17, left) reveals several interesting results. First, as expected, the distribution of the interdomain paths is affected by the utilisation of `AS-Path` prepending. One can see on Figure 17 that with an `AS-Path` prepending of one on the `lowID` link, the distribution of the interdomain paths has changed for almost all stub ASes. With this amount of prepending, 79% of the stub ASes receive now less than 40% of their interdomain paths via their *lowID* provider.

However, a second important point to mention is that the influence of `AS-Path` prepending is different for each stub AS: some receive all their traffic through the *highID* link while other ASes seem not to be affected. This implies that it can be difficult for a stub AS to predict the impact of the utilisation of `AS-Path` prepending on the distribution of its incoming traffic. This difference is due to the structure of the topology. In our topology as in the Internet, there exists a path between the two upstream providers of a stub AS. The length of the path between these two upstream providers determines the distribution of the interdomain paths after prepending. Let us first consider what happens when the two upstream providers of a stub AS are directly connected. In this case, the `lowID` provider will receive a direct route of two AS-hops and a route of two AS-hops through the `highID` provider. When comparing these two routes, the BGP decision process in our model relies on its random tie-breaker to select one over the other since no routing policies have been configured for *core* ASes. If the BGP decision process of the `lowID` provider selects the route via the `highID` provider, then the stub will receive all the interdomain paths via its `highID` provider. When the two providers are not directly connected, then the impact of the distribution of the interdomain paths depends on their respective connectivity.

In the topology with the dense core, the utilisation of `AS-Path` prepending has a stronger influence on the distribution of the interdomain paths as shown in Figure 17 (right). After prepending, most stub ASes receive less than 40% of their interdomain paths via their `lowID` provider. As with the lightly connected *core*, after prepending some stub ASes do not receive traffic via their `lowID` provider anymore. The difference between the lightly and the dense core topologies can be explained by the connectivity of the providers. In the dense core, there are more direct links between providers and the providers with the lower connectivity have a much better connectivity than the less connected providers in the lightly connected core.

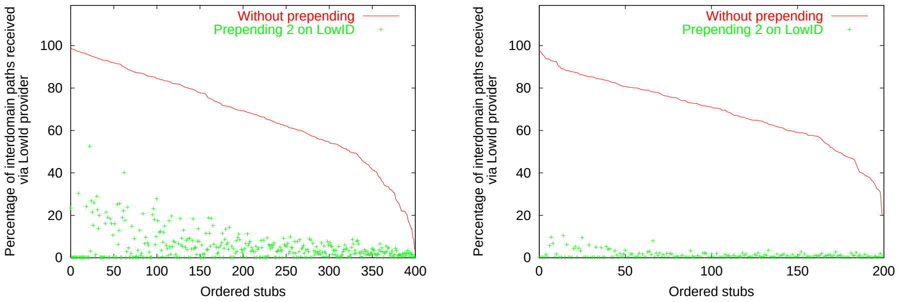

Fig. 18. Traffic distribution after prepending twice on the `lowID` link for the lightly connected (left) and the dense (right) cores

Prepending twice modifies significantly the distribution of the interdomain paths as shown by the simulation results in Figure 18. For the lightly connected core, only 3 stub ASes still receive more than 30% of the interdomain paths via their `lowID` provider after prepending. Furthermore, 86% of the stub ASes receive less than 10% of their traffic through the prepended provider. This means that after prepending twice on the `lowID` link, almost all the interdomain paths have been shifted to the other link. The 3 stub ASes for which the effect is less important have actually a very good connectivity via their `lowID` provider and a very weak connectivity via their `highID` provider. On the dense *core*, prepending twice on the `lowID` link moves almost all the interdomain paths on the `highID` link.

Prepending 7 times, or more, is often used on backup links that should only be used in case of failures. Our simulations with this amount of prepending show that all the interdomain paths are received via the `highID` provider. This is because the topologies we used do not contain routes longer than 6 AS hops. This is similar to the current Internet, where most routes have a length between 2 and 4 AS hops and very few routes a longer than 6 AS hops.

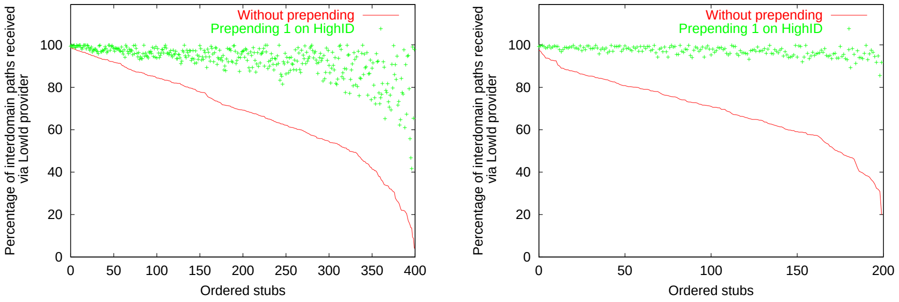

Fig. 19. Distribution of the interdomain paths after prepending once on the `highID` link for the lightly (left) and dense (right) cores

We have also studied the effect of prepending on the link to the `highID` provider. Figure 19 shows that in this case, the distribution of the interdomain paths is much more affected than when prepending was used on the link to the `lowID` provider. This result was expected since the `highID` provider has a weaker interdomain connectivity than the `lowID` provider. As when prepending was used on the `lowID` link, the effect of prepending is not the same for all stub ASes. For the lightly connected *core*, 97 % of the stub ASes receive less than 30% of their traffic through their `highID` provider. A few stub ASes still receive a large part of their interdomain paths via their `highID` link provider despite the prepending. This is due to the good connectivity of the `highID` providers connected to these stub ASes.

For the dense core, the impact of `AS-Path` prepending on the distribution of the interdomain paths is even more important as shown in the right part of Figure 19. Indeed, 84% of the stub ASes receive more than 95% of their interdomain traffic through their lowID provider.

Simulations with larger amounts of `AS-Path` prepending on the `highID` link have shown that most of the interdomain paths are received via the `lowID` provider. For example, for the dense core, all stub ASes receive less than 10 % of the interdomain paths via their `highID` provider when the stub ASes prepend twice the routes announced to this provider. For the dense core, all stub ASes receive less than 2% of the interdomain path via their `highID` provider after prepending twice.

9.5 Influence of `local-pref`

In the previous section, we have studied the impact of `AS-Path` prepending by studying the distribution of the interdomain paths starting from each AS and ending inside each stub AS. This study has been performed by assuming that each source of an interdomain path sends its packets along the best path selected by its decision process. However, as discussed in section 9.1, each AS can easily control its outgoing traffic by using the `local-pref` attribute which is evaluated first in the BGP decision process.

To evaluate the impact of the utilisation of the `local-pref` attribute by the stub ASes, we performed the same simulations as above, but by first configuring `local-pref` on each stub AS to force it to send all its packets via its `lowID` provider. Figure 20 compares the distribution of the interdomain paths in this simulation with the distribution obtained (see Figure 16) when each stub AS sent its packets along its best path toward each destination.

Surprisingly, based on this simulation result, the upstream provider selected by the stub ASes has only a minor influence on the distribution of the interdomain paths. This result is confirmed when we configured each stub AS to send their packets via their `highID` provider as shown in Figure 20 (right). A closer look at the results shows that 68% of the stub ASes receive more than 60% of their interdomain paths through their `lowID` provider when each stub AS sends its packets via its `lowID` provider. On the other hand, 66% of the stub ASes receive more than 60% of the their interdomain paths through their `lowID` provider when each stub AS sends its packets via its `highID` provider. Similar results were obtained with the dense core and when prepending was used.

This result can be explained by analysing all the interdomain paths. A closer look at those paths reveals that a small number of *core* ASes appear in a large fraction of those paths. In Figure 21, we show for the number of appearance of each *core* AS in these interdomain paths. Since each AS sent an IP packet with the `record route` option to each of the other 599 ASes reachable in our topology, there were 358801 interdomain paths. The analysis of those paths reveals that some ASes of the core appear in a very large number of interdomain paths and that many ASes only appear in a small number of paths. In the lightly connected core, one AS appeared in 100031 interdomain paths when the stub

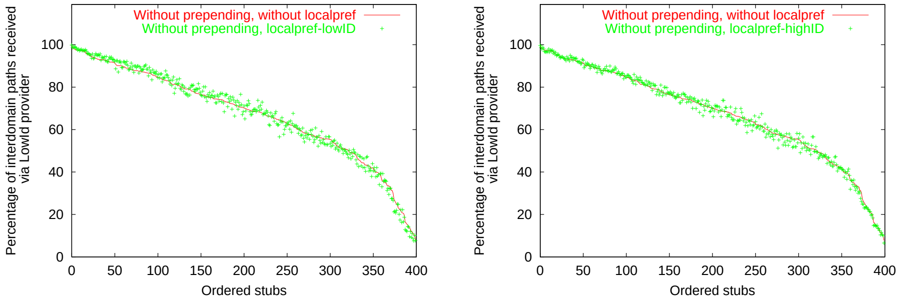

Fig. 20. Impact of `local-pref` on `lowID` link (left) and `highID` link (right)

ASes sent their packets along their best path. This number changed to 109527 (resp. 105236) when the stub ASes sent all their packets via `lowID` (resp. `highID`) provider. Figure 21 shows that the number of interdomain paths passing via each *core* AS does not change significantly when the stub ASes select one upstream provider or another.

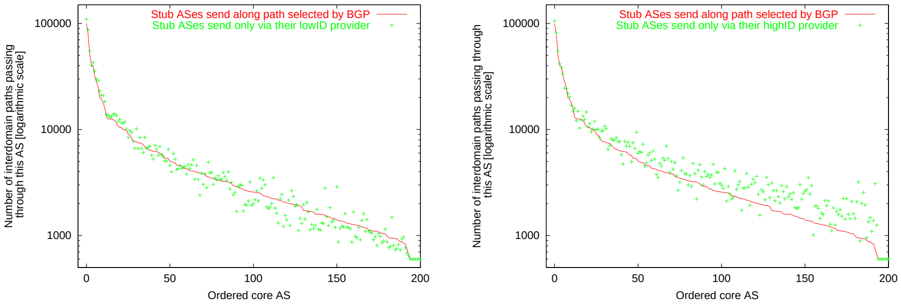

Fig. 21. Impact of the provider selected by the stub ASes on the interdomain paths

Based on these simulation results, it appears that the coupling between the traffic engineering techniques that allow stub ASes to control the flow of their incoming and the outgoing traffic is very weak. This weak coupling implies that the first hops of the path used by a stub AS to send its packets do not significantly influence the last hops of the path that these packets will take to reach their final destination.

10 An Approach for the Propagation of Quality of Service Information

Value-added IP service offerings that are likely to be deployed over the Internet often imply the negotiation of QoS requirements between a customer and a provider, customers possibly being providers themselves. Such QoS information includes parameters like one-way transit delays, inter-packet delay variations, loss rates, *etc.* [81]. This information can also be inferred by a DiffServ Code Point (DSCP) [2], marking indication, so that a given destination (an IP prefix or a host) could be reachable by different routes, depending on the level of quality both customer and provider have agreed upon.

The enforcement of end-to-end QoS policies is therefore conditioned by the provisioning of resources across domains. The characteristics of these resources will have to comply as much as possible with contractual QoS requirements, which means that the routers involved in the forwarding of the corresponding traffic should be aware of this QoS information, so that it might influence their routing decisions accordingly.

Generally speaking, the BGP attributes that have been specified so far enable the enforcement of a high-level kind of inter-domain routing policy, where service providers can influence the selection of the adjacent domain to reach a given destination. Nevertheless, this is the kind of information that cannot be propagated across domains, and that currently lacks of "QoS-related" knowledge, made of the valued parameters that have been introduced in the beginning of this section.

We believe there is a need for a finer granularity, where service providers would have the ability to exchange information about the classes of service they currently support, the destination prefixes that can be reached by means of traffic-engineered routes that have been computed and selected within their domain, as well as any kind of QoS indication that may contribute to the enforcement of end-to-end QoS policies.

From this perspective, the BGP4 protocol is one of the possible vectors for conveying QoS information between domains. A different way of extending BGP to convey QoS information has been described in [82].

10.1 Propagating Quality of Service Information between Domains

The choice of using the BGP4 protocol for exchanging QoS information between domains is not only motivated by the fact this is currently the only inter-domain (routing) protocol activated in the Internet, but also because the manipulation of attributes is a powerful means for service providers to announce QoS information with the desired level of precision. The approach relies upon the use of an additional attribute, and has identified the following requirements.

First, the approach must be kept scalable. The scalability of the approach can be defined in many ways that include the convergence time to reach a consistent view of the network connectivity, the number of route entries that will have to be maintained by a BGP peer, the dynamics of the route announcement mechanism (*e.g.*, how frequently and under which conditions should an UPDATE message containing QoS information be sent?), *etc.* Second, the operation of the BGP4 protocol must be kept unchanged. The introduction of a new attribute should not impact the protocol machinery, but the information contained in this attribute may very well influence the BGP route selection process. Third, the approach must allow a smooth migration. The use of a specific BGP attribute to convey QoS information should not constrain network operators to migrate the whole installed base at once, but rather help them in gradually deploying the QoS information processing capability.

The QOS_NLRI attribute To propagate QoS-related information across domains by means of the BGP protocol, an additional BGP4 attribute, named the QOS_NLRI (QoS Network Layer Reachability Information) attribute is specified in [83].

The QOS_NLRI attribute can be used for two purposes. First, it can be used to advertise a QoS route to a peer. A QoS route is a route that meets one or a set of QoS requirement(s) to reach a given (set of) destination prefixes. Such QoS requirements can be expressed in terms of *minimum one-way transit delay* to reach a destination, the experienced *delay variation* for IP datagrams that are destined to a given destination prefix, the *loss rate* experienced along the path to reach a destination, and/or the identification of the traffic that may use this route (identification means for such traffic include DSCP (DiffServ Code Point) marking). These QoS requirements can be used as an input for the route calculation process embedded in the BGP peers.

Second, the QOS_NLRI attribute can be used to provide QoS information along with the NLRI information in a single BGP UPDATE message. It is assumed that this QoS information will be related to the route (or set of routes) described in the NLRI field of the attribute.

From a service provider's perspective, the choice of defining the QOS_NLRI attribute as an optional transitive attribute is basically motivated by the fact that this kind of attribute allows for gradual deployment of QoS extensions to BGP4: not all the BGP peers are supposed to be updated accordingly, while partial deployment of such QoS extensions can already provide an added value as a means to enforce traffic engineering policies across domains that belong to the same network operator. For example, this would yield QoS-aware BGP route computation and selection for the range of value-added IP services offerings provided by the ISP, as well as an enhanced network resource optimisation and planning within the corresponding domains.

The contents of the QOS_NLRI attribute have been designed so that: *(i)* there is enough room to convey any kind of QoS information. The QoS information that has been identified so far includes transit delay information, loss rate, bandwidth information and Per Hop Behaviour (PHB) identification, *(ii)* the QoS information is tightly related to the destination prefixes that are contained in the NLRI field of the attribute, so as to allow for a route-level granularity whenever appropriate(instead of an AS-level granularity), as mentioned earlier, as one application example of [84] and *(iii)* The QoS information can be associated to other network protocols than IPv4, including its version 6 [85] whose header format includes implicit traffic engineering capabilities that should allow for an even finer level of granularity.

Basic operation The QOS_NLRI attribute is conveyed in BGP UPDATE messages that are exchanged between peers of different domains. BGP routers that are capable of processing the information contained in the QOS_NLRI attribute can provide this indication to their peers by means of the Capabilities Advertisement mechanism, as defined in [86].

By *processing* the QoS information contained in the QOS_NLRI attribute, we basically mean two different operations. First, the QoS information can be altered as long as it crosses the Internet. For example, one-way transit delay information can be modified by means of additive operation, whenever the information is propagated hop-by-hop between domains. And second, the QoS information can actually influence the BGP route selection process.

Because it is an optional and transitive attribute, the QOS_NLRI attribute will be propagated across domains, even though there may be peers along the path that are unable to process the information conveyed in the attribute. In this case, the Partial bit of the attribute will be set by such peers, meaning that the information propagated by the attribute is incomplete.

10.2 Simulation work

The simulation work aims at validating the technical feasibility of the approach (hence qualifying the added value introduced by the use of the QOS_NLRI attribute). It is also intended to focus on the scalability of the approach, within the context of the enforcement of inter-domain traffic engineering policies.

The simulation work has been organised as a step-by-step approach, which consists in the following four phases.

First, an IP network composed of several autonomous systems is modelled. Since this simulation effort is primarily focused on the qualification of the scalability related to the use of the QOS_NLRI attribute for exchanging QoS-related information between domains, it has been decided that the internal architecture of such domains be kept very simple, i.e. without any specific IGP interaction

Second, inside this IP network, some BGP peers that are QOS_NLRI aware, i.e. they have the ability to process the information conveyed in the attribute, while the other routers do not recognise the QOS_NLRI attribute by definition. Those routers will forward the information to other peers, by setting the Partial bit in the attribute, meaning that the information conveyed in the message is incomplete. This approach allows to elaborate on the added value introduced by a gradual deployment of the QoS extensions to BGP4.

Third, as far as QOS_NLRI aware BGP peers are concerned, they will process the information contained in the QOS_NLRI attribute to possibly influence the route decision process, thus yielding the selection (and the installation) of distinct routes towards a same destination prefix, depending on the QoS-related information conveyed in the QOS_NLRI attribute. From this implementation perspective, the BGP routing tables have been modelled so that they contain a "sub-section" where QOS_NLRI-capable peers will store the information conveyed in the attribute.

The last phase is to modify the BGP route decision process: at this stage of the simulation, the modified decision process relies upon the one-way delay information and it also takes into account the value of the Partial bit of the attribute.

Once the creation of these components of the IP network has been completed (together with the modification of the BGP route selection process), the behaviour of a QOS_NLRI-capable BGP peer is as follows. Upon receipt of a BGP UPDATE message that contains the QOS_NLRI attribute, the router will first check if the corresponding route is already stored in its local RIB, according to the value of the one-way delay information contained in both QoS Information Code and Sub-code fields of the attribute.

If not, the BGP peer will install the route in its local RIB. Otherwise (i.e. an equivalent route already exists in its database), the BGP peer will select the best of both routes according to the following criteria:

If both routes are said to be either incomplete (Partial bit has been set) or complete (Partial bit is unset), the route with the lowest delay will be selected, Otherwise, a route with the Partial bit unset is always preferred over any other route, even if this (complete) route reflects a higher transit delay.

If ever both Partial bit and transit delay information are not sufficient to make a decision, the standard BGP decision process (according to the breaking ties mechanism depicted in [5]) is performed.

A case study The current status of the simulation work basically relies upon the one-way transit delay information only, as well as the complete/incomplete indication of the Partial bit conveyed in the QOS_NLRI attribute.

The IP network has been modelled so that it is composed of 6 autonomous systems and 11 BGP peers. Future scenarios will be composed of more ASs. Figures /reffigure:case-study depicts the actual processing of the QoS-related information conveyed in the QOS_NLRI attribute, depending on whether the peer is QOS_NLRI-aware or not.

Figure 22 depicts the modelled network for a first case study. In this figure, the routers marked with an asterisk support the QOS_NLRI attribute while the others do not and the arrows indicate the delays known by each QOS enabled router. Let us consider the propagation of a BGP UPDATE message that contains the QOS_NLRI attribute, in the case where the contents of the attribute are changed, because of complete/incomplete conditions depicted by the Partial bit of the QOS_NLRI attribute.

Router S is a QOS_NLRI-capable speaker. Assume that it takes 20 milliseconds to node S to reach network 192.0.20.0: this information will be conveyed in a QOS_NLRI attribute that will be sent by node S in a BGP UPDATE message with the Partial bit of the QOS_NLRI attribute unset.

Router A is another QOS_NLRI BGP peer, and it takes 3 milliseconds for A to reach router S. Node A will update the QoS-related information of the QOS_NLRI attribute, indicating that, to reach network 192.0.20.0, it takes 23

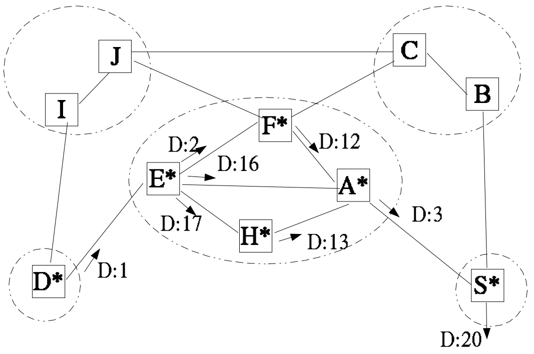

Fig. 22. A case study

milliseconds. Router A will install this new route in its database, and will propagate the corresponding UPDATE message to its peers.

On the other hand, router B is not capable of processing the information conveyed in the QOS_NLRI attribute, and it will therefore set the Partial bit of the QOS_NLRI attribute in the corresponding UPDATE message, leaving the one-way delay information detailed in both QoS Information Code and Sub-code unchanged.

Upon receipt of the UPDATE message sent by router A, router E will update the one-way delay information since it is a QOS_NLRI-capable peer. Finally, router D receives the UPDATE message, and selects a route with a 40 milliseconds one-way delay to reach network 192.0.20.0.

This example shows that the selection of a delay-based route over a BGP route may not yield an optimal decision. In the above example, the 40 ms-route goes through routers D-E-A-S, while a "truly optimal" BGP route would be through routers D-E-F-A-S, hence a 38 ms-route. This is because of the iBGP rule that does not allow router F to send an UPDATE message towards router E, because router F received the UPDATE message from router A thanks to the iBGP connection it has established with A.

These basic observations confirm that the enforcement of a QoS policy between domains by using the BGP4 protocol is obviously conditioned by the BGP4 routing policies that are enforced within each domain.

Preliminary simulation results Figure 23 reflects the first results obtained from a simulation network composed of 9 autonomous systems and 20 BGP peers. It shows the impact of the introduction of QOS_NLRI-capable peers in the network, where the enhanced route selection process results in an increase of the percentage of satisfied QoS requirements, especially in the region where these requirements are stronger (simulation units are arbitrary units so as to provide a better readability of the chart. In practice, such units would be expressed in milliseconds if the one-way transit delay were the key differentiator

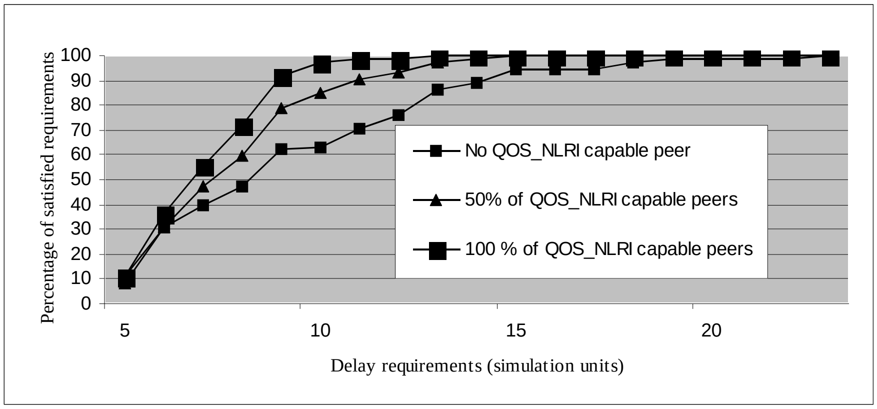

Fig. 23. Preliminary results of QOS_NLRI-capable peers.

for route computation). Furthermore, the chart clearly shows that the efficiency of a QOS_NLRI-inferred BGP route selection process is critical for the strongest requirements (e.g. for real-time traffic) at the sole expense of introducing a new attribute (and possibly influencing the breaking ties' mechanism logic), without questioning the BGP-based exchange of information between the domains of the Internet, as it's been done for several decades.

11 Improving Communication Qualities through Multi-homing

Since global connectivity has become a critical resource for organisations, sites attempt to improve the qualities of their communication facilities obtaining Internet connectivity through two or more providers. However, actual benefit obtained depends on how end-hosts and the routing system cope with this multiple path information, i.e. the multi-homing solution available. In the IPv4 Internet, several multi-homing solutions have been deployed, providing different levels of benefit. While some of them can be directly adopted in the IPv6 Internet, others do not seem to be aligned with some key design criteria of IPv6, such as routing system scalability or aggressive route aggregation, making them unsuitable for direct adoption. In this section we will review and evaluate existent and proposed solutions for the improvement of communication qualities through multi-homing.

11.1 Currently Available Solutions

In this section we will describe currently available multi-homing solutions. All the solutions described are available for IPv4 and while it is also possible to deploy them for IPv6, some of them are deemed unacceptable since they would jeopardise routing system scalability features.

The Incumbent Solution: Indiscriminate Route Injection The most widely deployed multi-homing solution in IPv4 is based on the announcement of the site prefix through all its providers, as illustrated in figure 24.

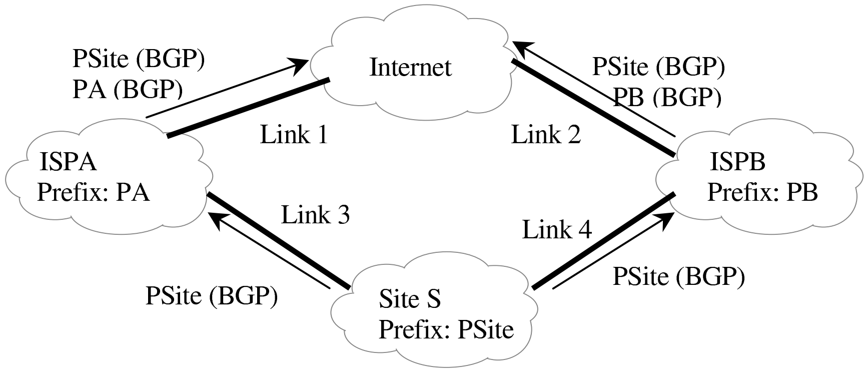

Fig. 24. Unrestricted route injection

In this solution, the site S obtains a prefix assigned directly from the RIR (Regional Internet Registry) or from one of its providers (ISPA or ISPB). Then, the site announces this prefix (PSite) to its providers using BGP. ISPA and ISPB announce the prefix to its providers, and this process continues until the route is announced into the Default Free Zone (DFZ). It must be noted that the PSite prefix can be part of one provider aggregate or it can be obtained directly from a RIR, although in any case the PSite prefix has to be announced separately in order to avoid that the longest prefix match rule diverts all the traffic to the provider announcing the more specific route.

This mechanism presents many desirable properties, such as optimal fault tolerance capabilities that includes preserving established connections throughout an outage, since alternative routes are managed without end-node perception. However, each multi-homed site using this solution contributes with two routes to the DFZ routing table, imposing additional stress to already over-sized routing tables. A resulting concern is the growing time taken for route withdrawal, that depending on the case can be long enough to cause retransmission from end-hosts [87]. So, this solution that originally provided optimal functionality and preserved almost unchanged established communications, currently is presenting poor performance because of its own success. For this reason, this solution is not considered to be acceptable for IPv6, for which some sort of provider aggregation is to be adopted, as it is described next.

Available Solutions Compatible with the Provider Aggregation Scheme In this section we will present solutions that are compatible with the usage of provider aggregatable addresses. We will first describe the configuration of a multi-homed site with provider aggregatable addresses and then we will present several current available multi-homing solutions compatible with this configuration.

Provider Aggregatable Addresses and Multi-homed Sites In order to reduce the routing table size, the usage of some form of provider aggregation is recommended, meaning that sites obtain prefixes which are part of the allocation of their provider, so that its provider only announce its own aggregate. In this scheme, the most beneficial aggregation is achieved by aggregating end-site prefixes into the prefix allocated to their direct provider [88], so that direct provider aggregation of end-sites is deemed necessary for scalability. Further aggregation, i.e. the aggregation of provider prefixes into their upstream provider prefix, leads to moderate aggregation benefits but it presents deployment challenges, making its adoption uncertain.

When provider aggregation of end-site prefixes is used, end-site host interfaces obtain one IP address from each allocation in order to be able to receive traffic through both providers. Note that ISPs only announce their aggregates to their providers, and they do not announce prefixes belonging to other ISPs aggregates.

This configuration presents several concerns:

- Additional address consumption, which may or may not be an issue depending on the protocol version used.
- Increased connection establishment time in case of failure. When Link 1 or Link 3 becomes unavailable, addresses containing the *PASite* prefix are unreachable from the Internet. New incoming (from the site perspective) connections addressed to *PASite* addresses will suffer from an increased establishment time, since the connection request to the unavailable address will timeout and alternative address, containing PBSite prefix will be used.
- Established connections will not be preserved in case of an outage. If Link 1 or Link 3 fails, already established connections that use addresses containing the *PASite* prefix will fail, since packets addressed to the *PASite* aggregate will be dropped because there is no route available for this destination. Note that an alternative path exists, but the routing system is not aware of it.
- Ingress filtering incompatibility. Ingress filtering is a widely used technique for preventing the usage of spoofed addresses. However, in this configuration, additional difficulties arise from the interaction between source address selection mechanism and intra-site routing systems, since the packet exit path and the source address carried in the packet must be coherent, in order to bypass ingress filtering mechanisms.
- Source address selection difficulties. When Link 1 or Link 3 fails, site hosts should not use addresses containing *PASite* prefix for initiating external communications, since reply packets will be dropped because there is no route available to the source address. If Link 3 is the one that has failed, the site exit router connecting with ISPA can be aware of the outage and it can

propagate the information to the hosts, so that the unavailable addresses is no longer used (this can be done using Router Advertisement [89] and Router Renumbering [90]). However, if Link 1 is the unavailable one, the situation cannot be solved using only these tools.

In brief, this configuration present better availability than the single-homed configuration but it does not improve the qualities of established connections during an outage. Therefore, additional mechanisms are needed to allow the usage of provider aggregatable addresses while still obtaining benefits equivalent to the incumbent multi-homing solution.

Auto-route Injection A multi-homing mechanism compatible with provider aggregation is presented in [91] and it is based on the outage-triggered injection of routes. In normal operation, the site S, that obtains one address block per provider, announces via BGP *PASite* to ISPA and *PBSite* to ISPB. In case of an outage, which can be detected by S by comparing the routing announcement obtained from both ISPs, the site injects both prefixes to the still working ISP.

This solution preserves established communications in any failure mode in which at least one path is available, and also allows new communications to be established using all advertised addresses. The impact of the outage in communications depends on the location of the outage. If the outage is topologically close to the site, route injection will be fast and routing system can converge fast enough so that the communications remains unaware of the failure. However, as the topological distance from the site to the faulty device increases, the time needed for route injection and routing system convergence grows, affecting communication. The impact will be an increased delay, which can even imply retransmissions from the source.

The presented mechanism preserves aggregation during normal operation and it only injects additional routing information during outages. However, considering the size of the Internet, probably there will be many simultaneous outages at any given time, so that additional routing information will always be present in the global routing tables. This solution is already deployed in IPv4 and it is susceptible to be deployed in IPv6. Whether the amount of extra information is acceptable or not for IPv6 remains to be evaluated.

Multi-homing Support at Site Exit Routers. This mechanism, presented in [91] and developed for IPv6 in [92], is aimed to provide last-hop link fault tolerance which is achieved through tunnels between site egress routers (RA, RB) and ISP border routers (BRB, BRA) as it is depicted in figure 25. In normal operation tunnelled paths are advertised with a low priority, to guarantee that traffic is routed through direct links whenever possible. In case of link failure, the link that is down is no longer advertised as a valid route, so the tunnelled path becomes the preferred route in spite of its low priority.

This mechanism has already been implemented in both IPv4 and IPv6 and it is fully compatible with the provider aggregation scheme. In this case, the solution provides better response times than the previous one in the case that the

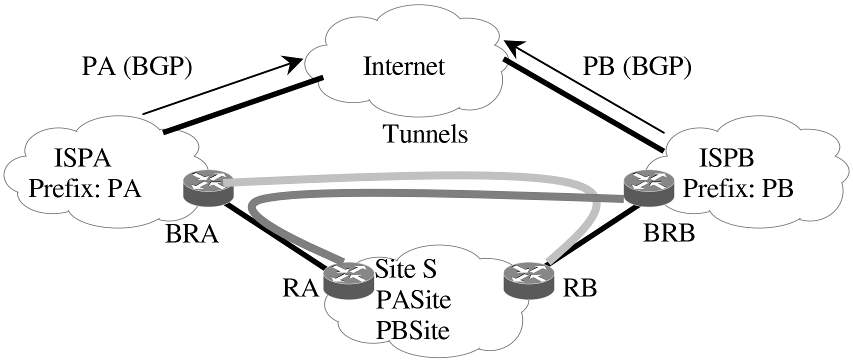

Fig. 25. Multi-homing support at site exit routers

outage is caused by a direct link failure. New connections can still be established using all addresses, and established connections are preserved. The only impact perceived by end-hosts is an increase of the transit delay due to the extra hops introduced, and a reduction of the path MTU that can originate fragmentation or packet loss.

The main drawback is the limited fault tolerance achieved by the solution, since it only preserves connectivity through a direct link failure episode, while other faults such as ISP breakdown remain uncovered.

11.2 Alternative Multi-homing Solutions

In this section we will present multi-homing solutions that have been proposed but that have not been widely accepted yet. Some of these mechanisms are IPv6-only, since they lie on special IPv6 features, such as the extended address range. Others have been deployed for IPv6 but also apply for IPv4. We will classify the presented solutions into two groups: host centric mechanisms and router centric mechanisms.

Host Centric Solutions Host centric solutions are mechanisms residing on hosts that deal essentially with address management to provide multi-homing capabilities. So, the host is capable of discovering multiple alternative addresses of the other end and it is also capable of discerning when to use them. In host centric solutions, interaction with routing system may provide additional information so that if it is available, the multi-homing mechanism performance improves, but if it is not, the solutions still work properly. Host solutions can be implemented at three levels, namely network level, transport level and application level, as it will be presented next.

Network Level Solutions A host network level solution is presented in [93]. In this solution, identifier and locator functions of IP addresses are explicitly separated by defining a new type of address, called *LIN6 generalised ID*, which is constructed concatenating a reserved prefix (the LIN6 prefix) with the Interface Identifier part of IPv6 addresses. These addresses are used by transport and upper layers for identifying endpoints of communications, but they are never transmitted in packets, since they are mapped into regular routable addresses by the LIN6 module of the host network layer. Routable addresses are formed by concatenating regular network prefixes delegated by ISPs and the Interface Identifier of the interfaces.

LIN6 generalised Identifiers are stored in the DNS and mapping information is stored in Mapping Agents whose location can be found in newly defined DNS records. Fault tolerance is supported by changing the routable address to which the LIN6 generalised ID is mapped to, i.e. when an ICMP error message is received, an alternative routable address is used for the same LIN6 generalised ID. It should be noted that the IP layer is not connection oriented and no connection timeout available to indicate that an error has occurred, so the only available mechanism for fault detection is based on ICMP error messages.

Transport Level Solutions Transport layer solutions propose that a transport layer connection can be identified not just by one IP address on each end, but by a set of addresses on each end. An extension to TCP has been proposed in [94], so that SYN packets carry several IP addresses available to reach the source. This information can be transported either in a TCP option or in a new IPv6 Extension Header. Fault tolerance is provided by switching addresses used to reach the other end, and address switching is triggered by TCP connection timeout.

A more sophisticated transport level solution is provided by the Stream Control Transport Protocol (SCTP) [95]. In this case, several IP addresses can be also used to reach the other end of the communication, but more sophisticated fault tolerance support is available based on a heartbeat mechanism to probe correspondent endpoint reachability through every address. This improved mechanism enables an intelligent choice of the alternative address to use.

Application Level Solutions Another possible approach is to delegate multihoming support to the application level. This implies that applications must deal with multiple addresses per endpoint i.e. they must be able to recognise that packets coming from different source addresses belong to the same communication. This provides maximum flexibility to applications. However, this also imposes extra complexity to application developers, who must deal with routing and reachability issues, being this not always desirable.

Final Considerations on Host Centric Solutions Support for fault tolerance in host centric solutions is based on retransmissions i.e. when an outage occurs, packets are lost and optionally ICMP errors packets are sent or timeouts occur, so the packet is retransmitted with an alternative address. In this type of solutions, all failure modes are covered, since whenever a path is available through one of the advertised addresses, the communication will be preserved. However, the host becomes aware of the outage at some communication layer, since packets must be retransmitted, introducing an increased delay in certain packets. It should be noted that this delay will probably be higher than the one introduced by the previous solution, in the case of a direct link outage.

Router Centric Solutions Currently deployed solutions are essentially router centric solutions i.e. routers provide the capabilities needed for multi-homing support while hosts remains unaware. This type of solutions has already proven to provide good fault tolerance, but it also has exhibited scalability limitations. Alternative mechanisms have been proposed which do not introduce additional information into the global routing table and therefore preserve aggregation. The Multi-Homing Aliasing Protocol (MHAP) [96] is a router mechanism that proposes the usage of two address spaces, one for multi-homed endpoint identification and another one for routing. In MHAP, multi-homed hosts are identified by a special type of IPv6 addresses, called *Multi-homed addresses* which are formed by a prefix specially reserved for these purposes concatenated with the corresponding Interface Identifier. However, these addresses are not used to route the packet through the Internet, but they are translated into regular Provider Aggregatable addresses. So the typical communication between a single-homed source and a multi-homed destination would be as follows: The source obtains the multi-homed address of the destination host through a DNS query. Then it sends the packet with the obtained multi-homed as destination address. When the packet reaches the site exit router, a process within the router called the *MHTP client* obtains the Provider Aggregatable addresses, and switches the multi-homed address with one of the obtained addresses. Afterwards, the packet is routed to the destination site using this address. Finally, the ingress router at the destination site translates the destination address back to the original multi-homed one. In this scheme, traffic addressed to multi-homed sites is translated twice, so that end-hosts remain unaware of changes. Mapping information about multi-homed addresses and Provider aggregatable addresses are exchanged on demand by concerned routers using BGP. Fault tolerance capabilities are provided by exchanging keepalives between both end-site exit routers. If keepalives are not received, an alternative address is used. It should be noted that keepalive timeout can be tuned so that established communications do not timeout.

12 Summary and Conclusions

In this chapter, we have discussed several Internet traffic engineering techniques. We have first addressed the needs of single autonomous systems by using Diffserv and MPLS. We have proposed an automated provisioning system, targeting to support demanding traffic requirements (SLSes), while at the same time optimising the use of network resources. We seek to place the traffic demands to the network in such a way as to avoid overloading parts of the networks and minimise the overall network cost. We devised a non-linear programming formulation and we proved though simulation that we achieve our objectives.

Moreover, we presented how this intra-domain traffic engineering and provisioning system can be policy-driven. We described the components of the necessary policy-based system extensions that need to be deployed in order to enhance or modify the functionality of the policy influenced components reflecting high-level business decisions. We then enlisted the policies which can be used to influence the behaviour of Network Dimensioning and presented an example of the enforcement of such a policy.

In the second part of this chapter we have discussed the traffic engineering techniques that are applicable between autonomous systems. We have first described the behavior of BGP and explained several techniques that can be used to control the flow of interdomain traffic. We have also discussed the characteristics of interdomain traffic and have shown that although an AS will exchange packets with most of the Internet, only a small number of ASes are responsible for a large fraction of the interdomain traffic. This implies that an AS willing to engineer its interdomain could move a large amount of traffic by influencing a small number of distant ASes. Second, the sources or destinations of interdomain traffic are not direct peers, but they are only a few ASes hops away. This implies that interdomain traffic engineering solutions should be able to influence ASes a few hops beyond their upstream providers or direct peers.

We have then presented a detailed evaluation of techniques that can be used to control the flow of the incoming traffic. Our detailed simulations of `AS-Path` prepending has shown that it is difficult to utilize this technique to achieve a given goal. Our simulations with `local-pref` have shown that the utilization of this technique had only a small influence on the traffic received by remote stub ASes.

We have then described the QoS_NLRI attribute that can be used to distribute QoS information between domains and have shown preliminary simulation results. However, this BGP-based approach to inter-domain traffic engineering raises issues that remain un-addressed, like the scalability of the approach and the QoS route aggregation capabilities of enhanced BGP peers.

Finally, we have discussed IPv6 multi-homing solutions. A set of current multi-homing solutions have been presented, starting with currently deployed ones and then presenting alternative new approaches that attempts to provide multi-homing benefits without jeopardizing overall scalability. We have then analyzed the improvements that each one of the described solutions provides from the communications point of view. As a conclusion, no solution that has been proposed yet combines scalability with the same benefits as the incumbent solution, although the host-centric approach could be an acceptable one.

References

1. D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. *Overview and Principles of Internet Traffic Engineering.* IETF, Informational RFC-3272, May 2002.
2. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *An Architecture for Differentiated Services.* IETF, Informational RFC-2475, December 1998.

3. E. Rosen, A. Viswanathan, and R. Callon. *Multi-protocol Label Switching Architecture.* IETF, Standards Track RFC-3031, January 2001.
4. M. Sloman. Policy Driven Management For Distributed Systems. *Journal of Network and Systems Management*, 2(4):333–360, December 1994.
5. Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). Internet Engineering Task Force, RFC 1771, March 1995.
6. Internet Engineering Task Force (IETF). Traffic Engineering Working Group (tewg). information available at: www.ietf.org/html.charters/tewg-charter.html.
7. D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. *Requirements for Traffic Engineering Over MPLS.* IETF, Informational RFC-2702, September 1999.
8. F. Le Faucheur and W. Lai (eds.). *Requirements for support of Diff-Serv-aware MPLS Traffic Engineering.* Internet draft, <draft-ietf-tewg-diff-te-reqts-07.txt>, work in progress, February 2003.
9. D. Katz, K. Kompella, and D. Yeung. *Traffic Engineering Extensions to OSPF Version 2.* IETF Internet draft, <draft-katz-yeung-ospf-traffic-10.txt>, work in progress, June 2003.
10. A. Feldman and J. Rexford. IP Network Configuration for Intra-domain Traffic Engineering. *IEEE Network Magazine*, 15(5):46–57, September/October 2001.
11. P. Aukia, M. Kodialam, P. V. Koppol, T. V. Lakshman, H. Sarin, and B. Suter. RATES: A Server for MPLS Traffic Engineering. *IEEE Network Magazine*, 14(2):34–41, September/October 2000.
12. B. Fortz and M. Thorup. Internet Traffic Engineering by Optimizing OSPF Weights. In *Proc. of IEEE INFOCOM'00*, pages 519–528. Israel, March 2000.
13. M. Kodialam and T. V. Lakshman. Minimum Interference Routing with Applications to Traffic Engineering. In *Proc. of IEEE INFOCOM'00*, pages 884–893. Israel, March 2000.
14. P. Trimintzios, I. Andrikopoulos, G. Pavlou, P. Flegkas, P. Georgatsos D. Griffin, D. Goderis, Y. T'Joens, L. Georgiadis, C. Jacquenet, and R. Egan. A Management and Control Architecture for Providing IP Differentiated Services in MPLS-based Networks. *IEEE Communications Magazine*, 39(5):80–88, May 2001.
15. K. Nichols, V. Jacobson, and L. Zang. *A Two-bit Differentiated Services Architecture for the Internet.* IETF, Informational RFC-2638, July 1999.
16. J. De Clercq, S. Van den Bosch, and A. Couturier. *An Architecture for a Gradual Deployment of end-to-end QoS on an Internet-wide Scale.* IETF Internet draft, <draft-declercq-vsn-arch-01.txt>, work in progress, June 2003.
17. P. Pan, Hahne E, and H. Schulzrinne. BGRP: A Tree-Based Aggregation Protocol for Inter-domain Reservations. *Journal of Communications and Networks*, 2(2):157–167, June 2000.
18. E. Mykoniati, C. Charalampous, P. Georgatsos, T. Damilatis, D. Goderis, P. Trimintzios, G. Pavlou, and D. Griffin. Admission Control for Providing QoS in IP DiffServ Networks: the TEQUILA Approach. *IEEE Communications Magazine*, 41(1):38–44, January 2003.
19. G. Ventre (ed.). *Chapter 8: Engineering of Future Internet Services (this book).* Springer, 2003.
20. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms and Applications.* Prentice Hall, 1993.
21. D. Mitra and K. G. Ramakrishnan. A Case Study of Multi-service, Multi-priority Traffic Engineering Design for Data Networks. In *Proc. IEEE GLOBECOM'99*, pages 1087–1093. Brazil, December 1999.

22. D. Mitra, J. A. Morrison, and K. G. Ramakrishnan. Virtual Private Networks: Joint Resource Allocation and Routing Design. In *Proc. IEEE INFOCOM'99*, pages 884–893. USA, March 1999.
23. S. Van den Bosch, F. Poppe, H. De Neve, and G. Petit. Choosing the Objectives for Traffic Engineering in IP Backbone Networks based on Quality-of-Service Requirements . In *Proc. of 1st Workshop on Quality of future Internet Services (QofIS'00)*, pages 129–140. Germany, September 2000.
24. S. Suri, M. Waldvogel, and P. R. Warkhede. Profile-Based Routing: A New Framework for MPLS Traffic Engineering. In *Proc. of 2nd Workshop on Quality of future Internet Services (QofIS'01)*, pages 138–157. Portugal, September 2001.
25. D. Mitra and Q. Wang. Stochastic Traffic Engineering, with Applications to Network Revenue Management. In *Proc. of IEEE INFOCOM'03*. USA, March/April 2003.
26. N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K.K. Ramakrishnan, and J. Van der Merwe. A Flexible Model for Resource Management in Virtual Private Networks. In *Proc. of ACM SIGCOMM'99*, Massachussetts, USA, August/September 1999.
27. A. Kumar, R. Rastogi, A. Silberschatz, and B. Yener. Algorithms for Provisioning Virtual Private Networks in the Hose Model. In *Proc. of ACM SIGCOMM'01*, San Diego, USA, August 2001.
28. A. Jüttner, I. Szabó, and Á Szentesi. On Bandwidth Efficiency of the Hose Resource Management Model in Virtual Private Networks. In *Proc. of IEEE INFOCOM'03*. USA, March/April 2003.
29. Z. Wang, Y. Wang, and L. Zhang. Internet Traffic Engineering without Full Mesh Overlaying. In *Proc. IEEE INFOCOM'01*. Alaska, April 2001.
30. Y. Breitbart, M. Garofalakis, A. Kumar, and R. Rastogi. Optimal Configuration of OSPF Aggregates. In *Proc. IEEE INFOCOM'02*. USA, June 2002.
31. P. Van Mieghem (ed.). *Chapter 3: Quality of Service Routing (this book)*. Springer, 2003.
32. S. Chen and K. Nahrstedt. An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions. *IEEE Network Magazine*, 12(6):64–79, November/December 1998.
33. A. Elwalid, C. Jin, S. H. Low, and I. Widjaja. MATE: MPLS Adaptive Traffic Engineering. In *Proc. IEEE INFOCOM'01*, pages 1300–1309. Alaska, April 2001.
34. A. Sridharan and R. Guérin. Achieving Near-Optimal Traffic Engineering Solutions for Current OSPF/IS-IS Networks. In *Proc. of IEEE INFOCOM'03*. USA, March/April 2003.
35. Z. Cao, Z. Wang, and E. Zegura. Performance of Hashing-based Schemes for Internet Load Balancing. In *Proc. of IEEE INFOCOM'00*, pages 332–341. Israel, March 2000.
36. P. Trimintzios, P. Flegkas, and G. Pavlou. Policy-driven Traffic Engineering for Intra-domain Quality of Service Provisioning. In *Proc. of 3rd Workshop on Quality of future Internet Services (QofIS'02)*, pages 179–193. Switzerland, October 2002.
37. J. Boyle and V. Gill and A. Hannan and D. Cooper and D. Awduche and B. Christian and W.S. Lai. *Applicability Statement for Traffic Engineering with MPLS*. IETF, Informational RFC-3346, August 2002.
38. D. Goderis (ed.). *Service Level Specification Semantics and Parameters*. Internet draft, <draft-tequila-sls-01.txt>, work in progress, December 2001. available at: www.ist-tequila.org/sls.
39. A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving Traffic Demands for Operational IP Networks: Methodology and Experience. *IEEE/ACM Transactions on Networking (TON)*, 9(3):265–279, June 2001.

40. A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic Matrices Estimation: Existing Techniques and Future Directions. In *Proc. of ACM SIGCOMM'02*, Pittsburgh, USA, August 2002.
41. K. Papagiannaki, Z.-L. Zhang N. Taft, and C. Diot. Long-Term Forecasting of Internet Backbone Traffic: Observations and Initial Models. In *Proc. of IEEE INFOCOM'03*. USA, March/April 2003.
42. P. Triminztios, T. Baugé, G. Pavlou, P. Flegkas, and R. Egan. Quality of Service Provisioning through Traffic Engineering with Applicability to IP-based Production Networks. *Computer Communications Journal, Elsevier Science*, 26(8):845–860, May 2003.
43. D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.
44. Z. Wang and J. Crowcroft. Quality of Service Routing for Supporting Multimedia Applications. *IEEE J. Selected Areas in Communications (JSAC)*, 14(7):1128–1234, September 1996.
45. E. W. Zegura, K. L. Calvert, and S. Bhattacharjee. How to model an internetwork. In *Proc. of IEEE INFOCOM'96*, pages 594–602. USA, March 1996.
46. P. Flegkas, P. Trimintzios, and G. Pavlou. A Policy-based Quality of Service Management Architecture for IP DiffServ Networks. *IEEE Network Magazine*, 16(2):50–56, March/April 2002.
47. R. Yavatkar, D. Pendarakis, and R. Guérin. *A Framework for Policy Based Admission Control*. IETF, Informational RFC-2753, January 2000.
48. P. Flegkas, P Trimintzios, G. Pavlou, I. Andrikopoulos, and C. Cavalcanti. On Policy-based Extensible Hierarchical Network Management in QoS-enabled IP Networks. In *Proc. of 1st IEEE Workshop on Policies for Distributed Systems and Networks (Policy '01)*, pages 230–246. U.K., January 2001.
49. J. Strassner, B. Moore, R. Moats, and E. Ellesson. *Policy Core LDAP Schema*. IETF Internet draft, <draft-ietf-policy-core-schema-16.txt>, work in progress, October 2002.
50. B. Moore, E. Ellesson, J. Strassner, and A. Westerinen. *Policy Core Information Model – Version 1 Specification*. IETF, Standards Track RFC-3060, February 2001.
51. Y. Rekhter and T. Li. A border gateway protocol 4 (bgp-4). Internet draft, draft-ietf-idr-bgp4-17.txt, work in progress, May 2002.
52. J. Stewart. *BGP4 : interdomain routing in the Internet*. Addison Wesley, 1999.
53. L. Subramanian, S. Agarwal, J. Rexford, and R. Katz. Characterizing the internet hierarchy from multiple vantage points. In *INFOCOM 2002*, June 2002.
54. S. Bartholomew. The art of peering. *BT Technology Journal*, 18(3), July 2000.
55. Cisco. NetFlow services and applications. White paper, available from `http://www.cisco.com/warp/public/732/netflow`, 1999.
56. W. Fang and L. Peterson. Inter-AS traffic patterns and their implications. In *IEEE Global Internet Symposium*, December 1999.
57. P. Pan, E. Hahne, and H. Schulzrinne. BGRP: A Tree-Based Aggregation Protocol for Inter-domain Reservations. *Journal of Communications and Networks*, 2(2), June 2000.
58. G. Huston. Analyzing the Internet's BGP routing table. *Internet Protocol Journal*, 4(1), 2001.
59. L. Kleinrock and W. Naylor. On measured behavior of the ARPA network. In *AFIS Proceedings, 1974 National Computer Conference*, volume 43, pages 767–780. John Wiley & Sons, 1974.
60. K. Claffy, H. Braun, and G. Polyzos. Traffic characteristics of the T1 NSFNET backbone. In *INFOCOM93*, 1993.

61. P. McManus. A passive system for server selection within mirrored resource environments using as path length heuristics. Available from `http://www.gweep.net/~mcmanus/proximate.pdf`, April 1999.

62. B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O.Bonaventure. Interdomain traffic engineering with bgp. *IEEE Communications Magazine*, May 2003.

63. University of Oregon. Route-views. Available from `http://antc.uoregon.edu/route-views`.

64. A. Broido, E. Nemeth, and K. Claffy. Internet expansion, refinement and churn. *European Transactions on Telecommunications*, January 2002.

65. K. Ishiguro. Gnu zebra 0.92a. Available from `http://www.zebra.org`.

66. S. Bellovin, R. Bush, T. Griffin, and J. Rexford. Slowing routing table growth by filtering based on address allocation policies. preprint available from `http://www.research.att.com/~jrex`, June 2001.

67. S. Uhlig, O. Bonaventure, and B. Quoitin. Interdomain Traffic Engineering with minimal BGP Configurations. In *Proc. of the* 18^{th} *International Teletraffic Congress, Berlin*, September 2003.

68. S. Borthick. Will route control change the internet ? *Business Communications Review*, September 2002.

69. B. Fortz, J. Rexford, and M. Thorup. Traffic engineering with traditional IP routing protocols. *IEEE Communications Magazine*, October 2002.

70. T. Griffin and G. Wilfong. Analysis of the MED oscillation problem in BGP. In *ICNP2002*, 2002.

71. Hung-Ying Tyan. *Design, Realization and Evaluation of a component-based compositional software architecture for network simulation.* PhD thesis, The Ohio State University, 2002.

72. B. J. Premore. Ssf implementations of bgp-4. available from `http://www.cs.dartmouth.edu/~beej/bgp/`, 2001.

73. J. H. Cowie, D. M. Nicol, and T. Ogielski. Modeling the Global Internet. *Computing in Science & Engineering*, (1):42–50, Jan/Feb 1999.

74. T. Griffin and B. Presmore. An experimental analysis of BGP convergence time. In *ICNP 2001*, pages 53–61. IEEE Computer Society, November 2001.

75. Z. M. Mao, R. Govindan, G. Varghese, and R. Katz. Route flap damping exacerbates internet routing convergence. In *ACM SIGCOMM'2002*, 2002.

76. M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-Law Relationships of the Internet Topology. In *ACM SIGCOMM*, Sept. 1999.

77. H. Tangmunarunkit, R. Govindan, and S. Jamin. Network Topology Generators: Degree-Based vs Structural. In *ACM SIGCOMM*, 2002.

78. A. Medina, A.Lakhina, I. Matta, and J. Byers. BRITE: An Approach to Universal Topology Generation. In *MASCOTS 2001*, August 2001.

79. A.L. Barábasi and R. Albert. Emergence of Scaling in Random Networks. *Sciences*, (286):509–512, October 1999.

80. R. Albert and A. Barabasi. Statistical mechanics of complex networks. *Review of Modern Physics*, pages 47–97, January 2002.

81. C. Demichelis and P. Chimento. IP Packet Delay Variation Metric for IPPM. Internet Engineering Task Force, RFC3393, August 2002.

82. L. Xiao, K. Lui, J. Wang, and K. Nahrstedt. QoS extensions to BGP. In *ICNP 2002*, Paris, France, November 2002.

83. G. Cristallo and C. Jacquenet. Providing Quality of Service Indication by the BGP-4 Protocol: the QOS_NLRI Attribute. Internet draft, draft-jacquenet-qos-nlri-04, Work in progress, September 2002.

84. D. Walton, D. Cook, A. Retana, and J. Scudder. Advertisement of multiple paths in BGP. Internet draft, draft-walton-bgp-add-paths-01.txt, work in progress, November 2002.
85. S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. Internet Engineering Task Force, RFC 2460, December 1998.
86. R. Chandra and J. Scudder. Capabilities Advertisement with BGP-4. Internet Engineering Task Force, RFC 2842, May 2000.
87. C. Labovitz, A. Ahuja, A. Bose, and J. Jahanian. Delayed Internet Routing Convergence. August 2000.
88. T. Li Y. Rekhter. An Architecture for IPv6 Unicast Address Allocation. Internet Engineering Task Force, RFC 1887, December 1995.
89. W. Simpson T. Narten, E. Nodmark. Neighbor Discovery for IP Version 6 (IPv6). Internet Engineering Task Force, RFC2461, December 1998.
90. M. Crawford. Router Renumbering for IPv6. Internet Engineeing Task Force, RFC2894, August 2000.
91. Y. Rekhter T. Bates. Scalable Support for Multi-homed Multi-provider Connectivity. Internet Engineering Task Force, RFC2260, January 1998.
92. J. Hagino. IPv6 multihoming support at site exit routers. Internet Engineering Task Force, RFC3178, April 2001.
93. F. Teraoka, M. Ishiyama, M. Kunishi, and A. Shionozaki. LIN6: A Solution to Mobility and Multi-Homing in IPv6. Internet draft, draft-teraoka-ipng-lin6-02. Work in progress, November 2001.
94. P. Tattam. Preserving active TCP sessions on multihomed. IPng Working Group Meeting Minutes, Tokio. Work in progress, September 1999.
95. R. Stewart and al. Stream Control Transmission Protocol. Internet Engineering Task Force, RFC2960, October 2000.
96. M. Py. Multi Homing Translation Protocol. Internet draft, draft-py-multi6-mhtp-02. Work in progress, 2001.