PAPER

Inter-Domain QoS Routing: Optimal and Practical Study

Rui PRIOR[†a)], *Student Member and* **Susana SARGENTO**[††b)], *Nonmember*

SUMMARY This paper addresses the problem of inter-domain QoS routing with Service Level Agreements (SLA) for data transport between peering domains, using virtual-trunk type aggregates. The problem is formally stated and formulated in Integer Linear Programming. As a practical solution, we define the QoS_INFO extension to the BGP routing protocol, conveying three different QoS metrics (light load delay, assigned bandwidth and a congestion alarm), and a path selection algorithm using a combination of these metrics. We present simulation results of QoS_INFO, standard BGP, and BGP with the QoS_NLRI extension, and compare them with the optimal route set provided by the ILP formulation. The results show that our proposal yields better QoS than standard BGP or BGP with the QoS_NLRI extension, since it is able to efficiently avoid congested paths, and that the impact of QoS_INFO in route stability is relatively low.

key words: *inter-domain QoS routing, linear optimization, BGP extension*

1. Introduction

The provision of multimedia services with real-time requirements through the Internet is conditioned by its ability to ensure that certain Quality of Service (QoS) requirements are met. The introduction of QoS routing mechanisms able to select paths with the required characteristics is of major importance towards this goal. Though much attention has been paid to QoS in IP networks, most of the effort has been centered on intra-domain; much less has been done in the scope of inter-domain, which is a much more complex problem, for a number of reasons. The Internet is a complex entity, comprised of Autonomous Systems (AS) managed by very diverse operators. If it is to be widely deployed, an inter-domain QoS routing mechanism must be capable of handling the heterogeneity of the Internet and impose minimum requirements on intra-domain routing, in order to be appealing to the different operators. The introduction of QoS metrics should not disrupt the currently existing inter-domain routing: the QoS and non-QoS versions should interoperate, allowing for incremental deployment, and the stability of the routes should not be overly affected by the QoS mechanisms. Conciliating the required stability with the dynamic nature of QoS information is a major challenge in inter-domain QoS routing. A related issue is that of scalability: a solution that does not scale to the dimension of the Internet cannot be deployed widely enough to be useful.

Manuscript received August 7, 2006.
Manuscript revised October 1, 2006.
[†]The author is with LIACC, University of Porto, Portugal.
[††]The author is with IT, University of Aveiro, Portugal.
a) E-mail: rprior@dcc.fc.up.pt
b) E-mail: ssargento@det.ua.pt
DOI: 10.1093/ietcom/e90–b.3.549

In this paper we formally state the problem of inter-domain QoS routing with virtual trunks and formulate it as an Integer Linear Programming (ILP) optimization problem, proving that routes thus obtained are cycle-free. Using this formulation in a Mixed Integer Programming (MIP) code, we may obtain the optimal solution to the inter-domain QoS routing with virtual trunks problem in a given topology and traffic demand matrix. We propose a practical solution for inter-domain QoS routing based on both static and coarse-grained dynamic metrics: it uses the light load delay and assigned bandwidth (both static) in order to improve the packet QoS and make better use of network resources, and a coarse-grained dynamic metric for path congestion to avoid overloaded paths. We define the QoS_INFO extension to the Border Gateway Protocol (BGP) [1] to transport these QoS metrics and the algorithm to use them for path selection. Using the ns-2 simulator [2] we compare the proposed protocol with standard BGP and with BGP with the QoS_NLRI extension [3] conveying static one-way delay information (expected route delay in light load conditions). The optimal solution for the same topology and traffic matrix is also used as a baseline for the comparison. Results show that the QoS parameters of the route set obtained with QoS_INFO are the closest to those of the optimal route set. Specifically, we show that congestion and packet losses are much lower with QoS_INFO than with standard BGP or with QoS_NLRI.

The paper is organized as follows. The next section briefly describes related work. Section 3 contains the formal description of the problem and its formulation in ILP. Section 4 describes the QoS_INFO extension to BGP and the associated path selection algorithm. In Sect. 5 we compare the optimal results with simulation results from BGP, QoS_NLRI and QoS_INFO. Finally, Sect. 6 draws our conclusions.

2. Related Work

A framework for QoS-based Internet routing, adopting the traditional separation between intra- and inter-domain routing, was defined by Crawley et al. [4]. They discussed the goals of an inter-domain QoS routing and the associated issues that must be addressed, and provided general guidelines that should be followed by any viable solution to QoS routing in the Internet. However, they did not specify the set of QoS metrics to be transported or the algorithms for using such metrics in the choice of inter-domain routes.

A series of statistical metrics for QoS information ad-

vertisement and routing, tailored for inter-domain QoS routing, though also applicable to intra-domain routing, were defined by Xiao et al. [5], along with algorithms to compute them along the path. These metrics, the Available Bandwidth Index (ABI), the Delay Index (DI), the Available Bandwidth Histogram (ABH) and the Delay Histogram (DH), convey information expressed in terms of one or more probabilistic intervals. Simulation results show that by using these metrics, selected routes are closer to optimality than when using static metrics; moreover, the overhead is lower and the stability higher than when using the corresponding instantaneous (purely dynamic) metrics. However, these approaches consider only a single QoS parameter, making it difficult to simultaneously satisfy different requirements. When optimizing by bandwidth, paths with large delay may be chosen while others with less, yet sufficient, available bandwidth and much lower delay may be available. Conversely, when optimizing by delay, a route with low available bandwidth may be selected; switching to this route may cause congestion, increasing the delay. When the delay information is updated, the previous route might be selected again, and so on, causing route flapping (though on longer time scales than with dynamic metrics).

Cristallo and Jacquenet proposed an extension to BGP with a new optional and transitive attribute, QoS_NLRI, for the transport of several types of QoS information [3]. This work is focused on the specification of the attribute, including the formats for transporting different parameters, such as reserved data rate or minimum one-way delay, and does not specify how the information is to be used in path selection. Simulation results demonstrating its use with (static) information on one-way packet delay are provided, though.

3. Inter-Domain QoS Routing with Virtual Trunks

In this section we formally describe the problem of inter-domain routing with virtual trunks and formulate it as an Integer Linear Programming (ILP) problem.

3.1 Virtual Trunk Model of the Autonomous Systems

Though the use of some inner information of the ASs is important for inter-domain QoS routing, the exact topology and configuration of the ASs should not be used for two reasons: (1) the level of detail would be excessive, complicating the route computation task and, most important, (2) network operators usually want to disclose the minimum possible amount of internal information about their networks.

In this work, we use a "black box" model where only externally observable AS information is disclosed. The intra-domain connections between edge routers are replaced by virtual trunks with specific characteristics interconnecting the peering ASs. Each virtual trunk corresponds to a particular (*ingress link*, *egress link*) pair, and has a specific amount of assigned bandwidth and an expected delay. These values depend on the internal topology of the AS, on the intra-domain routing and on resource management

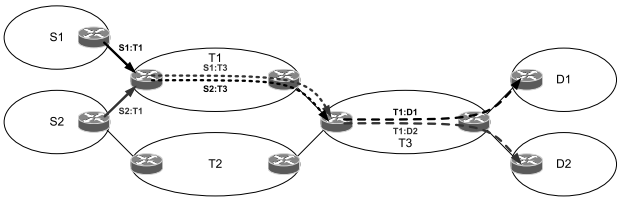


Fig. 1 Virtual trunk-type SLSs.

performed by the operators, and usually reflect SLAs established between the operator of the AS providing the edge-to-edge transport service and the operators of the peering ASs. The virtual trunk model can be implemented with DiffServ, and is frequently found in Multiprotocol Label Switching (MPLS) networks, where a given amount of bandwidth is assigned to each Label Switched Path (LSP); however, it is especially suited for Dense Wavelength Division Multiplexing (DWDM) transit networks, where lightpaths are established between pairs of edge routers, providing physical data pipes with given capacities, corresponding to the virtual trunks.

The virtual trunk model of ASs is illustrated in Fig. 1. A Service Level Specification (SLS) between domain S1 and domain T1 states that X traffic may flow between S1 and domain T3; an SLS between domain T1 and domain T3 states that Y traffic may flow between T1 and domain D1. Aggregates are managed internally within each (transit) domain, ensuring that enough resources are assigned, and no imposition is made regarding the mechanisms used to this end.

The configuration of the virtual trunks must be consistent with the inter-domain links. In particular, the summed bandwidth of all virtual trunks traversing AS j and going to AS k must be less than the bandwidth of the inter-domain link connecting ASs j and k; similarly, the summed bandwidth of all virtual trunks coming from AS i and traversing AS j must be less than the bandwidth of the inter-domain link connecting ASs i and j.

3.2 Problem Statement

Let $G = (V, E)$ be an undirected graph with edge capacities $c_{i,j}$ and edge delays $w_{i,j}$. Each node represents an AS, and the edges correspond to the inter-domain links. Additionally, we define a set F of aggregate flows between pairs of nodes and a corresponding matrix of traffic demands $a_{s,d}$ for all $(s, d) \in V^2$, where s and d denote the source and destination nodes, respectively.

Given any three nodes i, j and k, where i is directly connected to j and j is directly connected to k, there may be a traffic contract (SLS) stating that j provides a virtual trunk between i and k with assigned (reserved) capacity $r_{i,j,k}$. The amount of data transported from i to k via j is, therefore, bounded by $r_{i,j,k}$. If no such contract exists, we say that $r_{i,j,k} = 0$. Since each virtual trunk is mapped to an actual path inside the AS, it has an associated delay $y_{i,j,k}$, corresponding to the delay of that path. We denote by L the set

of all virtual trunks (i, j, k).

The virtual trunks must satisfy the conditions $o_{j,k} + \sum_i r_{i,j,k} \leq c_{j,k}$, where $o_{j,k}$ is the minimum capacity for traffic originated at node j and destined to or traversing node k, and $t_{i,j} + \sum_k r_{i,j,k} \leq c_{i,j}$, where $t_{i,j}$ is the minimum capacity for traffic destined to node j and originated at or traversing node i.

The expected total delay suffered by packets of a given flow is the sum of the $w_{i,j}$ and $y_{i,j,k}$ parameters along the path followed by the flow. Our goal is to find the set of hop-by-hop routes that minimize the delay while guaranteeing that inter-domain link and virtual trunk capacities are not exceeded.

3.3 Problem Statement Transform

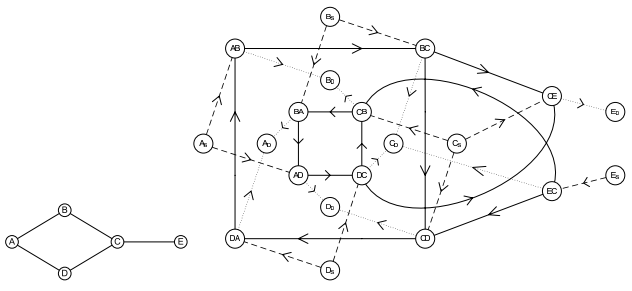
In order to formulate the stated problem as an ILP problem, we first transform the original graph into a transformed graph where the virtual trunks are explicitly accounted for.

3.3.1 Transform Graph

It is possible to transform the graph into a directed multigraph where edges correspond to virtual trunks; however, it is difficult to account for the delays of all links in the original graph (inter-domain links) without counting some of them twice. Therefore, we add virtual nodes to the directed multigraph in order to obtain a resulting directed graph.

Virtual trunks are established between an entry link and an exit link. Therefore, we add two virtual vertices per link of the original graph, one for each direction, and virtual trunks are represented by edges connecting these virtual nodes. Moreover, in order to forbid a node of the original graph from being traversed directly (instead of via a virtual trunk), we split each original node into two: one source virtual node with outgoing edges only, and one destination virtual node with incoming edges only. Flows on the transform graph exist between source and destination virtual nodes.

Figure 2 provides an example of a cyclic graph and its transform containing all possible virtual trunks. The solid edge connecting the virtual nodes ij and jk corresponds to the virtual trunk for sending traffic from node i to node k via node j, and has capacity $r_{i,j,k}$ (that of the virtual trunk), and delay $y_{i,j,k} + w_{j,k}$, where $y_{i,j,k}$ is the internal delay of the virtual trunk and $w_{j,k}$ the delay of the inter-domain exit link. Each dashed edge (j_S, jk) corresponds to the inter-domain exit link from node j to node k, and has delay $w_{j,k}$ and infinite capacity. Each dotted edge (ij, j_D) corresponds to the inter-domain entry link in node j from node i, and has zero delay and infinite capacity. Though the transform graph looks overly complex when compared to the original one, the number of variables and constraints in the ILP formulation is not increased, since a formulation based on the original graph would require variable unfolding in order to be linear. Also keep in mind that an undirected graph has half the number of edges of the equivalent directed graph.



(a) Original graph (b) Transform graph

Fig. 2 Cyclic network with 5 nodes.

3.3.2 Generation of the Transform Graph

In this section we present an algorithm for the generation of the transform graph $G' = (V', E')$ from the original graph G and the set of virtual trunks, informally described above. The algorithm is as follows:

1. For each node $i \in V$
   a. Add node i_S to the set S of sources and to the set V' of nodes; add node i_D to the set D of destinations and to V'
2. For each (undirected) edge $\{i, j\} \in E$
   a. Add node ij to V'; add node ji to V'
   b. Add edge (ij, j_D) to the set E' of edges, with capacity $c'_{ij,j_D} = \infty$ and delay $w'_{ij,j_D} = 0$; add edge (ji, i_D) to E', with capacity $c'_{ji,i_D} = \infty$ and delay $w'_{ji,i_D} = 0$
   c. Add edge (i_S, ij) to E', with capacity $c'_{i_S,ij} = \infty$ and delay $w'_{i_S,ij} = w_{i,j}$; add edge (j_S, ji) to E', with capacity $c'_{j_S,ji} = \infty$ and delay $w'_{j_S,ji} = w_{j,i}$
3. For each (directed) virtual trunk $(i, j, k) \in L$
   a. Add edge (ij, jk) to E' and to the set L' of virtual trunk edges, with capacity $c'_{ij,jk} = r_{i,j,k}$ and delay $w'_{ij,jk} = y_{i,j,k} + w_{j,k}$
4. For each flow $(i, j) \in F$
   a. Add flow (i_S, j_D) to the set F' of flows; set traffic demand $a'_{i_S,j_D} = a_{i,j}$

When the algorithm finishes, we have the transform graph $G' = (V', E')$, the associated edge capacity and edge delay matrices C' and W', a set $L' \subset E'$ of virtual trunk edges, a set $S \subset V'$ of source nodes, a set $D \subset V'$ of destination nodes, a set F' of flows, and the respective traffic demand matrix A'.

3.3.3 Complexity of the Transform Graph

The number of nodes and edges of the transform graph G' is related to the original (undirected) graph G and the set of virtual trunks in the following way. The number of nodes is two per node of the original graph (one source and one destination, e.g., AS and AD) plus two per edge of the original

graph (one for each direction, e.g., AB and BA). The number of edges is four per edge of the original graph (combinations of source/destination and transmission/reception, e.g., (A_S, AB), (AB, B_D), (B_S, BA) and (BA, A_D)) plus one per virtual trunk (e.g., (AB, BC)). In the example of Fig. 2, the original graph has 5 nodes, 5 edges and 12 possible virtual trunks. The transform, therefore, has 20 nodes ($2\times5+2\times5$) and 32 edges ($4\times5+12$).

3.3.4 Backward Conversion of the Routes

A route p' on the transform graph may be converted back to a route p on the original graph by analyzing the traversed edges: edge (i_S, ij) on the transform graph corresponds to node i on the original, edge (ij, jk) to node j, and edge (jk, k_D) to node k. For example, the route (B_S, BC, CE, E_D) on the transform graph of Fig. 2.(b) corresponds to the route (B, C, E) on the original graph.

3.4 Formulation as ILP Problem

We now formulate our bandwidth-constrained global route delay optimization hop-by-hop routing problem as an ILP problem with boolean variables using the transform graph. Formulation is simpler in the transform graph, as some constraints are already enforced by the topology: since there are no incoming edges in source nodes, it is not necessary to use a constraint disallowing incoming traffic for flows originated at those nodes (similarly for destination nodes).

Our objective is to minimize the global delay while respecting the bandwidth limits, assuming that the network has enough capacity to satisfy all demands. In addition to the transform data obtained by the above described algorithm, let us define a set of positive flow weights $b_{s,d}$ for all $(s,d) \in F'$. Two different optimizations may be obtained by using different weight values. The first alternative uses $b_{s,d} = 1, \forall (s,d) \in F'$, stating that all flows have equal importance — optimization is performed on a per-route basis. The second alternative uses $b_{s,d} \propto a'_{s,d}, \forall (s,d) \in F'$, stating that a flow's importance is proportional to its traffic demand — optimization is performed on a traffic volume basis. We define the boolean decision variables $x_{i,j}^{s,d}$ which take the value 1 if the flow $(s,d) \in F'$ is routed through the edge $(i,j) \in E'$ and 0 otherwise.

The problem can, thus, be formulated as follows:

$$\text{Minimize} \sum_{(i,j)\in E'} \sum_{(s,d)\in F'} b_{s,d} w'_{i,j} x_{i,j}^{s,d} \quad \text{subject to}$$

$$x_{i,j}^{s,d} \in \{0,1\}, \forall (s,d) \in F', (i,j) \in E' \tag{1}$$

$$\sum_{(s,d)\in F'} a'_{s,d} x_{i,j}^{s,d} \le c'_{i,j}, \forall (i,j) \in L' \tag{2}$$

$$\sum_{i\in V':(i,j)\in E'} \sum_{(s,d)\in F'} a'_{s,d} x_{i,j}^{s,d} \le c_i, \forall i \in (V' - S - D) \tag{3}$$

$$\sum_{(j,k)\in E'} x_{j,k}^{s,d} - \sum_{(i,j)\in E'} x_{i,j}^{s,d} = 0, \forall (s,d) \in F', j \in (V' - \{s,d\}) \tag{4}$$

$$\sum_{(s,j)\in E'} x_{s,j}^{s,d} = 1, \forall (s,d) \in F' \tag{5}$$

$$\sum_{(i,d)\in E'} x_{i,d}^{s,d} = 1, \forall (s,d) \in F' \tag{6}$$

$$\sum_{j\in V':(i,j)\in E'} \sum_{\substack{t\in S \\ t\ne d}} x_{i,j}^{t,d} \le |S| \cdot x_{s,i}^{s,d}, \quad \forall (s,d) \in F', i \in S : (s,i) \in E' \tag{7}$$

$$\sum_{(i,j)\in E':(j,d)\in E'} \sum_{\substack{s\in S \\ s\ne d}} x_{i,j}^{s,d} = \sum_{j\in V':(j,d)\in E'} \sum_{\substack{s\in S \\ s\ne d}} x_{j,d}^{s,d}, \forall d \in D \tag{8}$$

Constraint set (1) imposes boolean decision variables, meaning that flows cannot be split over multiple paths; (2) states that the sum of all flows traversing a virtual trunk edge will not exceed its capacity; (3) states that the sum of all flows traversing (leaving) a virtual node i corresponding to an inter-domain link in the original graph must be less than c_i, the capacity of the inter-domain link.

Constraint sets (4), (5) and (6) are the mass balance equations: (4) means that each flow entering a node that is neither source nor destination for that flow must leave it and vice-versa; (5) means that each flow leaves the source node once and, conversely, (6) means that each flow enters the destination node once.

Constraint set (7) means that if a flow from a source to a destination traverses a given virtual node directly connected to that source, no other flows to the same destination may traverse a different virtual node connected to the same source. On the original graph it means that if the flow from a given node to a certain destination leaves that node by a given link, no flow to the same destination traversing that node may leave it by a different link — in other words, it imposes hop-by-hop routing.

Finally, (8) prevents routing loops at the destination nodes of flows in the original graph, by forcing flows arriving at a node directly connected to their destination virtual node to use that direct path. Failing this, a flow would be counted twice (or more) on the left hand side and only once on the right hand side, invalidating the equality.

Theorem 1: Paths obtained through this optimization process are guaranteed to be cycle-free on the transform graph.

Proof: Satisfaction of conditions (5) and (6) implies that each flow leaves the source virtual node exactly once (5) and arrives at the destination virtual node exactly once (6), therefore the source and destination virtual nodes belong to the path. There are no incident edges to source virtual nodes, therefore these nodes cannot be in a cycle. Conversely, there are no incident edges from destination virtual nodes, therefore these nodes cannot be part of a cycle either.

Now let p be a path with a cycle from a given source $s \in S$ to a given destination $d \in D$, and p^* the same path with the cycle removed. The cycle may only include intermediate nodes (source and destination nodes cannot be part of cycles). If the above constraints (notably the capacity constraints) are satisfied with path p from s to d, then they are also satisfied with path p^*, from s to d. Since $b_{s,d} >$

$0, \forall(s,d) \in F'$ and $w'_{i,j} > 0, \forall(i,j) \in E' : i \notin S \wedge j \notin D$, the cost of using p^* would be lower than the one using p, therefore p could not be in the optimal route set (it would not minimize the cost function). □

Theorem 2: Paths obtained through this optimization process are guaranteed to be cycle-free on the original graph.

Proof: The proof is based on the following three lemmas concerning different types of cycles.

Lemma 1: A path satisfying the above conditions cannot contain cycles that do not include the destination node on the original graph.

Proof: Let p_1 be a path on the original graph from source s to destination d containing a cycle that does not include node d, and p'_1 the equivalent path in the transform graph. Since the cycle on p_1 does not contain the destination d, it must contain a node i left by the flow twice by different edges, (i, j) towards the cycle and (i, k) towards the destination node. Therefore, in the transform graph, p'_1 must contain virtual nodes ij and ik. Constraint (7) implies that $x_{i,j}^{i,d} = 1$ and $x_{i,k}^{i,d} = 1$. However, this violates constraint (5); therefore, p_1 can only contain cycles that include the destination node d. □

Lemma 2: A path satisfying the above conditions cannot contain cycles that include the destination node and the preceding edge on the original graph.

Proof: Let p_2 be a path on the original graph from source s to destination d containing a cycle that includes node d and the edge incident to d, (i, d); let p'_2 be the equivalent path in the transform graph. In this case, the flow enters d twice by the edge (i, d), from the source and from the cycle. Therefore, in the transform graph, p'_2 encloses a cycle containing the virtual node id. This contradicts theorem 1, which states that p'_2 is guaranteed to be cycle-free on the transform graph, meaning that a cycle containing node d and the edge incident to d cannot exist in the returned route set. □

Lemma 3: A path satisfying the above conditions cannot contain cycles that include the destination node but not the preceding edge on the original graph.

Proof: Let p_3 be a path on the original graph from source s to destination d containing a cycle that includes node d but not the edge incident to d; let p'_3 be the equivalent path in the transform graph. The flow enters node d twice by different edges, (i, d) and (j, d), from the source and from the cycle. Therefore, in the transform graph, p'_3 contains virtual nodes id and jd, contributing two units to the left hand side of constraint (8). According to constraint (6), the destination virtual node can only be entered once; therefore this flow's contribution to the right hand side of constraint (8) can only be one unit. Since the same applies to all flows, no flow on the original graph can have a cycle containing the destination node d but not the edge incident to d. □

The preceding lemmas cover all possible cycles on the original graph, proving that paths satisfying the constraints are guaranteed to be cycle-free on the original graph. □

4. Proposed Protocol and Associated Algorithms

While the ILP formulation of the inter-domain QoS routing problem presented in the previous section is useful as a baseline for comparison with real protocols in controlled environments where all the input data is known, it cannot be used in the implementation of a real protocol itself for several reasons: first, the problem of 0-1 integer programming is known to be NP-complete [6]; second, because it requires knowledge of the traffic matrix, which is not easy to obtain in real utilization scenarios; and third, because it requires global knowledge of the virtual trunk SLAs, which are usually disclosed only to the involved peers. In this section we propose a virtual-trunk-aware inter-domain QoS routing protocol, based on an extension of BGP, for practical implementation in real internetworks.

4.1 QoS Routing

Currently, version 4 of BGP has become the *de facto* standard for inter-domain routing in the Internet. BGP is a path vector protocol, exchanging reachability information between connected ASs through UPDATE messages. Besides the destination prefix, information on the traversed ASs (AS_PATH) and the next hop (NEXT_HOP) is provided for advertised routes. The most common policy for path selection is the minimum number of hops in the AS_PATH. Even though the AS_PATH length metric bears only a very loose relation to QoS parameters, BGP can easily be extended to convey and use virtually any kind of relevant QoS information without breaking backward compatibility. We extended BGP to use three QoS metrics: assigned bandwidth (static), path delay under light load (static) and a dynamic metric for path congestion described below.

4.2 Metrics

Virtual trunk information is explicitly included using BGP to carry information on the amount of bandwidth contracted between two domains regarding data transport to a third one. The assigned bandwidth, reflecting traffic contracts, is essentially static. It is updated along the path to be the minimum, that is, the bottleneck bandwidth (concave metric). Notice that our model does not require explicit and quantified agreements, only that transport operators assign a certain capacity for data transport between their connected peers; explicit SLAs are just a means to guarantee that reasonable virtual trunk assignments are performed.

Information on the expected delay in light load conditions (a lower bound for the expected packet delay) is also carried. Minimization of this metric allows not only for better packet QoS, but also for a more rational use of network resources, since in high capacity links with significant

length, such as those found in today's transit networks, it consists mostly on the sum of propagation delays [7], directly proportional to the traversed span of fiber, as long as there is no congestion. The light load delay metric is static, and is summed along the path (additive metric).

The third QoS metric conveyed by our proposed extension is a path congestion alarm. The concept of congestion is deliberately vague and may, therefore, be translated into a coarse objective metric, minimizing the overhead in message exchange and path recomputation typical of dynamic metrics. The congestion alarm is expressed by an integer with three possible values: 0 — not congested; 1 — very lightly congested; 2 — congested. This metric is updated along the path to the maximum value (convex metric). In a basic version, congestion is inferred from the utilization of the aggregates; a more advanced version would also use other parameters, such as packet loss, average length of traversed router queues or measured delay. The main requirement for the congestion alarms, the sole dynamic metric in our proposal, is that changes should be infrequent, for scalability and stability reasons; hysteresis and related techniques may be used in assigning the alarm levels to this end.

An effective value of the congestion alarm is used for path selection instead of the received value, aiming at reducing the fluctuations in virtual trunk usage; it is the same as the received value, unless the received value is 1 and the route is already in use, in which case the effective alarm is 0. This means that when level 1 (light congestion) is reached, the route should not be used to replace another one, but if it is already in use, a switch to a different one should only occur if a higher congestion level is reached. This behavior is meant to avoid synchronized route flapping.

4.3 Path Selection Algorithm

The three above mentioned QoS metrics are conveyed in the UPDATE messages by a newly defined Path Attribute, QoS_INFO, which is optional and transitive (meaning that ASs which do not yet support the extension simply forward the received value), and are updated by the BGP-speaking routers at each transit domain, taking into account the virtual trunks between the domain to which the route is advertised and the "next hop domain" for the route. Notice that these virtual trunks are shared among different source to destination routes: in Fig. 1, for example, all traffic transported from T1 to D1 via T3 shares the T1:D1 virtual trunk, independently of being originated at S1 or S2.

Figure 3 illustrates the propagation of the delay, bandwidth and congestion alarm metrics in the QoS_INFO attribute of UPDATE messages. When the destination AS (AD2) first announces the route to an internal network, it may omit the QoS_INFO attribute if this network is directly connected to the announced NEXT_HOP. On receiving the UPDATE, the edge router at transit domain TD2 creates (or updates, if already present) the QoS_INFO attribute with metrics of the outgoing link, for route selection purposes (this step is omitted in the figure). If the route is selected,

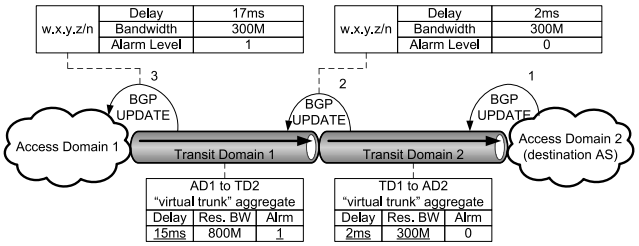


Fig. 3 Propagation of metrics in the QoS_INFO attribute.

```
set Traffic to dest = Local traffic to dest + Transit traffic to dest
for both routes
   if Alarm_rcv = 1 and route in use then set Alarm_eff = 0
   else set Alarm_eff = Alarm_rcv
if both routes have Assigned BW < traffic to dest,
   choose the one with larger Assigned BW
else if one route has Assigned BW < traffic to dest,
   choose the other one
else if Alarm_eff is different,
   choose the route with lower Alarm_eff
else if Delay is different,
   choose the route with least Delay
else if Assigned BW is different,
   choose the route with larger Assigned BW
else use normal BGP rules (AS_PATH length, etc.)
```

Fig. 4 Route comparison/selection function.

it is propagated to all peering domains; the QoS_INFO attribute sent to the different upstream domains is different, since the metrics are updated with respect to the virtual trunk aggregates. The same process is repeated at transit domain TD1. Notice that in the UPDATE sent from TD1 to AD1, the delay metric (17 ms) is the sum of the delays of the concatenated virtual trunks (2 ms and 15 ms), the assigned bandwidth is the minimum along the path (300 Mbps) and the congestion alarm the maximum (1). The virtual trunk values that contribute to the final values received by AD1 for this route are underlined in the figure.

Figure 4 shows the algorithm for route comparison used in the decision processes in pseudo-code. Delay information is used to select the fastest/shortest route. Information on the assigned bandwidth is used to eliminate, from the set of possible choices, routes with insufficient bandwidth to support the current outgoing traffic aggregate from the local AS to the destination (measured by monitoring at the edge routers, including flows generated at the local AS and flows traversing it); it is also used as tie breaker when two routes for the same destination have the same announced delay. The alarm levels are used to eliminate congested routes from the set of possible choices. Elimination of routes with insufficient capacity prevents congestion of those routes to a certain degree, contributing to lower message and processing overhead and to route stability.

4.4 Route Aggregation

A very important aspect in inter-domain routing is the pos-

sibility of aggregating routes. Without the deployment of route aggregation and Classless Inter-Domain Routing (CIDR) [8] in the 1990s, the routers would not have been able to support the increasing number of advertised routes. Paradoxically, little attention is given to aggregation in inter-domain QoS routing proposals, in general.

The use of a metric so coarse-grained as the congestion alarm in this proposal is aggregation-friendly. While the introduction of new metrics reduces the possibilities of aggregation compared to the standard, non-QoS-aware BGP, congestion alarm values will almost always be either 0 or 1, meaning that much aggregation is still possible. This is particularly true if congestion is introduced in transit domains, since it is common to all routes sharing the congested virtual trunk. The light load expected delay metric may easily be made compatible with aggregation if assumed to be an indicator rather than an exact value: in this case, two routes may be aggregated if the smaller delay is more than a certain fraction (say 75%) of the larger one, announcing the larger value for the aggregated route. The hierarchical structure of the Internet allows for a large degree of aggregation with this approach. The bandwidth metric, however, is more difficult to deal with, even with a hierarchical structure. We are currently working on how to conciliate the bandwidth metric with high levels of aggregation. We are also evaluating the use of the congestion metric alone and of a combination of congestion and light load delay metrics without the bandwidth metric. The evaluation of the different aggregation possibilities is left for further work.

5. Simulation Results

In this section we present simulation results obtained in ns-2 [2] of the QoS_INFO proposal for inter-domain QoS routing. These results concern the performance, in terms of delay, loss probability and inter-domain links congestion, of QoS_INFO when compared to standard BGP, to BGP with the QoS_NLRI extension conveying static one-way delay information (the expected delay of the route in light load conditions), and to optimal solutions obtained using the ILP formulation of Sect. 3.4 with a MIP code (Xpress-MP from Dash Optimization [9]). They also concern the number of updates required to provide inter-domain QoS and the stability of the routes. Note that the QoS_NLRI extension can be used to convey QoS parameters other than delay, and that the extension does not specify whether the delay information is static or dynamic. In fact, [3] is focused on the BGP extension for the transport of QoS information, not specifying the way that information is to be used by BGP in the path selection process. Therefore, in this comparison we used the scenario illustrated in [3].

The amount of traffic in inter-domain scenarios is extremely high, making it very difficult to complete simulations with realistic parameters within a reasonable time span. For this reason, we have chosen to simulate the signaling protocol normally at the packet level, but not the data traffic, which was mathematically simulated using well-known M/G/1 queuing model with three different packet sizes: 50% of packets with 40 bytes (representing 4% of the traffic volume), simulating SYN, ACK, FIN and RST TCP segments; 20% of packets with 80 bytes, simulating packetized voice (3% of traffic volume); and 30% of packets with 1500 bytes, simulating full size TCP segments (93% of traffic volume). These packet sizes reflect the bimodality currently observed in internet traffic [10], complemented with voice packets, whose frequency tends to increase. Queuing delays were obtained using the Pollaczek-Khintchine formula [11], $W_Q = \frac{\lambda E[S^2]}{2(1-\lambda E[S])}$ where W_Q is the queuing delay, λ is the traffic arrival rate and S is the service time, and computation of total packet delays was based on the Kleinrock independence approximation [11].

5.1 Simulation Scenario

To have meaningful results, a realistic topology and traffic matrix is required. We have used a hierarchical topology (Fig. 5) with two large transport providers with broad geographical coverage, four regional providers and 19 local providers. Abstracted at the AS level, the topology has 25 nodes (ASs) and 36 inter-domain links. The traffic demand for each route (source-destination pair) is constant during the simulation. The distribution of traffic demand values for the different routes is summarized in Fig. 6, having a maximum of 1.1 Gbps, an average of 45 Mbps and a standard deviation of 90 Mbps. The link bandwidth was assigned based on expected demands. The configuration of the virtual trunk type SLSs in our proposed model was performed automatically, based on the link bandwidth, the traffic matrix and a set of feasible routes (proportional distribution of link bandwidth). Not all triplets (a, b, c) such that a is connected to b and b to c have a corresponding SLS — whenever this is the case, traffic between a and c should use intermediate nodes other than b. Traffic that does not match an established SLS or that exceeds its assigned capacity is discarded at the ingress routers of the ASs.

Thresholds for setting alarm levels on path usage were 35% of the SLS bandwidth for level 1 and 80% for level 2, except where stated otherwise. We ran simulations for 8200 simulated seconds, discarding data for the first 1000 in or-

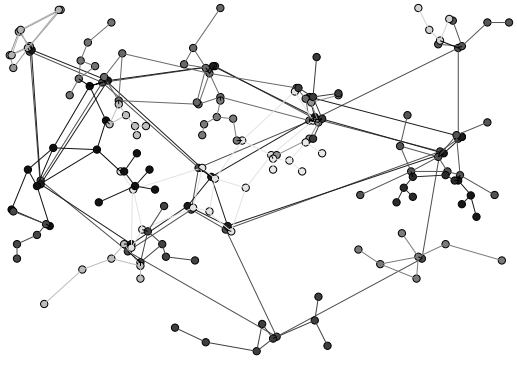


Fig. 5 Simulated topology.

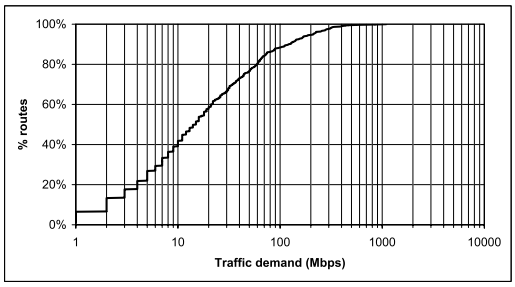


Fig. 6 Cumulative distribution function (CDF) of traffic demands.

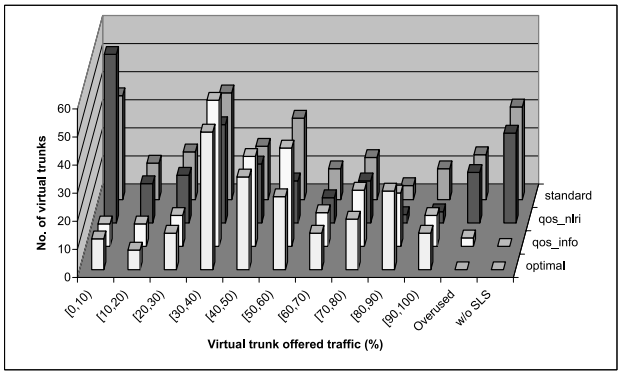


Fig. 7 Offered traffic distribution per virtual trunk.

der to filter out transient effects, and evaluated link usage, route optimality, route stability, QoS parameters and signaling overhead.

5.2 Link Usage, Route Optimality and QoS Parameters

In the first experiment we compare the three inter-domain routing mechanisms: standard BGP, BGP with QoS_NLRI and our proposed QoS_INFO with respect to link usage, route optimality and QoS parameters.

Figure 7 shows histograms with the distribution of the offered traffic for the links and the virtual trunks in the three approaches, averaged out of the 7200 useful simulation seconds. The same results are also provided for the optimal route set. The *overused* class corresponds to virtual trunks having an offered load above their capacity, and the *w/o SLS* class to AS triplets (a, b, c) with traffic but without an established SLS; in both cases, a significant portion of packets is consistently discarded due to link capacity limitation or SLS policing (not only a very small portion due to sporadic queue overload).

With standard BGP, routes are normally chosen based on the lowest number of elements in the AS Path, not taking into consideration path delay or congestion. As a result, 22% of the routes were sub-optimal in terms of expected light load delay. Regarding utilization, 16 out of the 211 virtual trunks (7.6%) were overused and, even worse, there was traffic on 33 triplets without established SLSs (15.6% compared to the number of SLSs). As a consequence, packet losses were 17.1% of the total traffic demand.

With the QoS_NLRI BGP extension carrying light load path delay information (static), routes are optimal in terms of expected light load delay. Congestion, however, is even worse than with standard BGP: 18 of the virtual trunks (8.5%) are congested, and there is traffic on 32 triplets without established SLSs (15.2% compared to the number of SLSs). As a result, the overall packet loss figure was 28.2%. The fact that congestion is worse in QoS_NLRI than in standard BGP is probably related to the fact that by minimizing the number of AS hops, standard BGP tends to exploit the hierarchical character of the network by preferring a more logical path comprising a small number of transport operators with broad geographical coverage[†] to a path consisting on a large number of operators with small coverage that may, nevertheless, have a lower light load delay value.

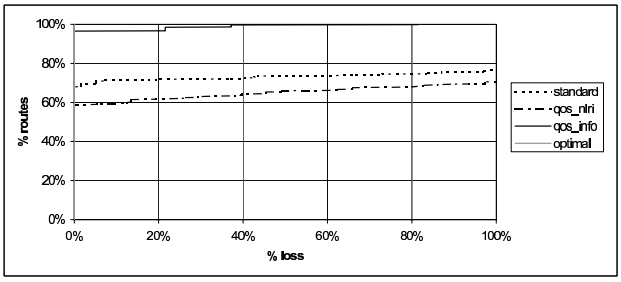


Fig. 8 Percentage of routes with loss probability $\leq x$.

With our proposed QoS_INFO approach, there was no traffic on AS triplets without a corresponding SLS, and only 3 SLSs were overused (1.4%). The overall packet loss, of only 0.4%, was much lower than in both of the previous cases. The reason for this is that the system reacts to congestion by changing the affected routes. Obviously, optimized results had no overused virtual trunks or traffic on AS triplets without corresponding SLSs.

Figure 8 shows the packet loss probability CDF for the routes at the end of the simulation[††] in the different scenarios. Again, our proposed QoS_INFO approach yields better results, with 96.5% of the routes having a negligible packet loss probability, contrasting to only 58.8% in QoS_NLRI and 68.0% in the standard BGP. In the optimal case, 100% of the routes had no packet losses.

Figure 9 shows CDFs of the expected packet delay for the routes (sum of propagation and transmission delays with the expected queuing delays along the path). Since policing is performed on the virtual trunks and their assigned capacity is consistent with the capacity of the inter-domain links they traverse, there was no link congestion in most cases, therefore the route delays were kept low. Nevertheless, 2.2% of the routes in QoS_NLRI traversed a congested link and suffered large delays. Except for these routes, the delays are close in all cases, with the QoS_INFO curve practically overlapping that of the optimization.

[†]In non-hierarchical topologies standard BGP performed worse than QoS_NLRI with respect to congestion.

[††]Since routing with QoS_INFO is based on dynamic information, routes do change in the course of the simulations; in standard BGP and BGP with QoS_NLRI all routes are stable during the useful simulation period.

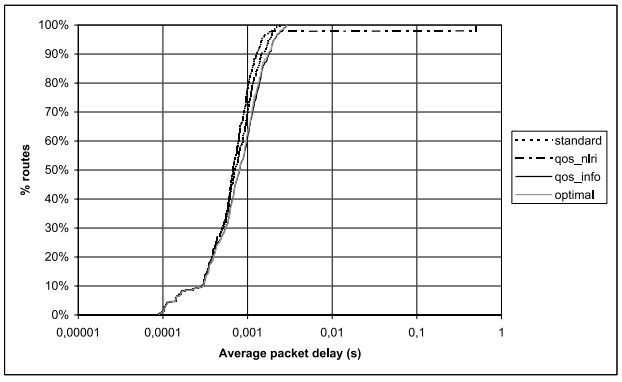


Fig. 9 Percentage of routes with packet delay $\leq x$.

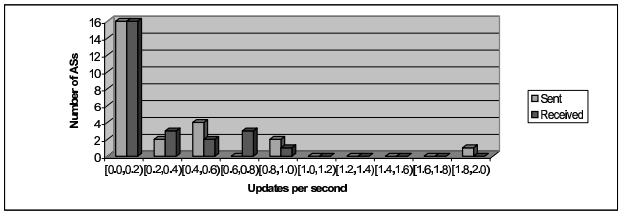


Fig. 10 Distribution of the number of updates per second in ASs.

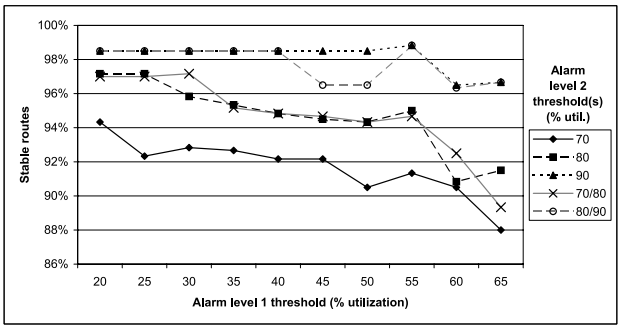


Fig. 11 Route stability vs. alarm level thresholds.

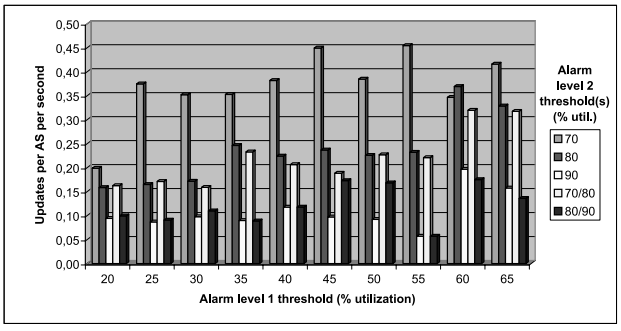


Fig. 12 Frequency of BGP updates.

5.3 Signaling Overhead and Route Stability

The drawback of the QoS_INFO approach, as usual with dynamic QoS routing approaches, is increased signaling load and decreased route stability. In the performed simulations we measured an average of 6.16 updates per second for the whole topology, or 0.246 per AS. These updates, however, do not affect all ASs equally, since some routes are stable, while others oscillate. The distribution of the frequency of sent and received updates is shown in Fig. 10. With the other models all routes are stable as long as there are no topology changes (due, e.g., to link failures). It is worth noting that if the delay information conveyed in the QoS_NLRI extension was dynamic, based on measurements, then route oscillations would also occur in this model; on the other hand, link overloads would be reduced. Regarding route stability, with the QoS_INFO approach, 572 out of a total of 600 routes in the topology (ca. 95%) were stable, meaning that they did not change during the useful simulation period; the other 5% did change, though with varying frequency.

Since the choice of a new route is triggered by changes in the alarm levels, the SLS utilization thresholds used to assign a given alarm level have strong influence in the stability of the routes. In order to evaluate this influence, we evaluated route stability with simulations using utilization values from 20% to 65% (x axis) of the bandwidth assigned to the SLSs as threshold for alarm level 1, and from 70% to 90% as threshold for alarm level 2 (different curves). In an attempt to increase route stability, we have also introduced hysteresis by using two different values for th_2[†] — a change from alarm level 1 to 2 occurs only when the high value is crossed, but in order to return to level 1, the utilization must drop below the lower level. The results are shown in Fig. 11. We may see that relatively low values of threshold for alarm level 1 (th_1) tend to improve the route stability, especially for lower values of th_2. As th_1 gets closer to th_2, route stability decreases. Higher values of th_2 also tend to improve route stability: without hysteresis, the highest value achieved was 98.8% for $th_1 = 55\%$ and $th_2 = 90\%$. However, though such values did not lead to increased packet losses (0.3%) nor to the use of non-established virtual trunks, and only increased the number of oversused virtual trunks to 4 in 211 (1.9%), they would have to be lowered in a practical deployment, since traffic demand is more variable. Contrary to our expectations, hysteresis did not seem to improve route stability: the results are better than using only the lower level of th_2, but comparable to using only the higher level.

In Fig. 12 we show the average number of updates per second per node, without and with hysteresis. Similarly to route stability, the number of updates with hysteresis is generally comparable to using only the higher level. However, a more conclusive comparison would require simulations using many different topologies, which is left for further work. In any case, the interest of hysteresis in a practical deployment with dynamic traffic demands is higher than in these simulations using a static traffic demand matrix, as the average thresholds would have to be lower.

[†]The introduction of hysteresis in th_1 has much lower relevance due to the fact that selected routes with $th_1 = 1$ are treated as if $th_1 = 0$.

6. Conclusions

This paper addressed the problem of inter-domain QoS routing. Our proposal is based on the use of virtual trunk type aggregates for the indirect transport of traffic between two different administrative domains across a peering third one, usually (though not necessarily) defined by means of SLSs between the respective operators. We formally stated the problem of SLA-aware inter-domain QoS routing and formulated it as an Integer Linear Programming (ILP) optimization problem, providing a proof that routes thus obtained do not contain cycles.

As a practical solution, we proposed the QoS_INFO extension to BGP, using a combination of three different metrics (assigned bandwidth, expected light load delay and congestion alarm) in order to simultaneously achieve different and conflicting goals: finding non-congested paths that satisfy the QoS requirements of the data flows, minimizing the network resources used to transport the flows, and minimizing the message exchange, path computation overheads and route instability. Though one of the metrics (congestion alarm) is dynamic, its coarse granularity and the rules for its use in the path selection algorithm are such that the impact overhead in message exchange and in path recomputation is minimized, and route stability increased.

Simulations were performed to evaluate our proposal and compare it to standard, QoS-unaware BGP and to the QoS_NLRI extension. The results show that though it represents an improvement over standard BGP, routing using only static QoS parameters is also unable to avoid path congestion. With our QoS_INFO proposal, congested paths and their consequences on QoS are avoided. Although there is a penalty in overhead and route stability in doing this, most of the routes are stable, especially if the thresholds for alarm setting are appropriately selected. The introduction of hysteresis in the alarm level assignment did not seem to improve stability and overhead, though it is expected to have some impact with dynamic traffic demands.

Some topics for further improvment of this work are devising a more advanced algorithm for the assignment of alarm levels to aggregates, the introduction of route aggregation and its conciliation with accurate QoS information, in order to improve scalability, and the simulation of the proposed QoS routing mechanism with a large number of randomly generated topologies to better ascertain its behavior in the real word.

Acknowledgments

The authors would like to thank João Pedro Pedroso and Ana Paula Tomás for some useful suggestions and for helping us get started with the optimization software. We would also like to thank Dash Optimization for the use of an Academic Partnership Program license of Xpress-MP.

This work is based on results of the IST FP6 Integrated Project DAIDALOS [12], funded by the European Community. It reflects the authors' views, and the EC is not liable for any use that may be made of this information.

References

[1] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol 4 (BGP-4)," RFC 4271 (Draft Standard), Jan. 2006.
[2] The Network Simulator — ns-2. http://www.isi.edu/nsnam/ns/
[3] G. Cristallo and C. Jacquenet, "The BGP QOS_NLRI attribute," IETF Internet Draft, Feb. 2004.
[4] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, "A framework for QoS-based routing in the Internet," RFC 2386 (Informational), Aug. 1998.
[5] L. Xiao, J. Wang, K.S. Lui, and K. Nahrstedt, "Advertising inter-domain QoS routing information," IEEE J. Sel. Areas Commun., vol.22, no.10, pp.1949–1964, Dec. 2004.
[6] R.M. Karp, "Reducibility among combinatorial problems," in Complexity of Computer Computations, ed. R.E. Miller and J.W. Thatcher, pp.85–103, Plenum Press, 1972.
[7] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot, "Measurement and analysis of single-hop delay on an IP backbone network," IEEE J. Sel. Areas Commun., vol.21, no.6, pp.908–921, Aug. 2003.
[8] V. Fuller, T. Li, J. Yu, and K. Varadhan, "Classless inter-domain routing (CIDR): An address assignment and aggregation strategy," RFC 1519 (Proposed Standard), Sept. 1993.
[9] Dash Optimization. http://www.dashoptimization.com
[10] R. Sinha, C. Papadopoulos, and J. Heidemann, "Internet packet size distributions: Some observations," Oct. 2005. http://netweb.usc.edu/ rsinha/pkt-sizes/
[11] D. Bertsekas and R. Gallager, Data Networks, 2nd ed., ch. 3, pp.149–270, Prentice-Hall, 1992.
[12] DAIDALOS Project (IST-2002-506997) Homepage. http://www.ist-daidalos.org

Rui Prior received the Lic and MSc degrees in Electrical and Computer Engineering from the Faculty of Engineering of the University of Porto in 1997 and 2001, respectively. He has worked as a researcher in INESC Porto, and is currently an Assistant Lecturer and PhD student at the Department of Computer Science of the Faculty of Sciences of the University of Porto, doing research at the Information Networks Group of the Laboratory of Artificial Intelligence and Computer Science (LIACC).

Susana Sargento graduated in electronics and telecommunications engineering from the University of Aveiro in 1997, and concluded her Ph.D. in 2003. From September 2002 to January 2004 she was an Assistant Professor at the Computer Science Department of the University of Porto, and leading the Networks and Communications group at LIACC. Since February 2004 she has been an assistant professor at the University of Aveiro. Her main research interests are in the areas of next-generation heterogeneous networks, with emphasis on QoS, mobility, multicast, and charging issues.