Resilient Availability and Bandwidth-aware Multipath Provisioning for Media Transfer Over the Internet

Sahel Sahhaf, Wouter Tavernier, Didier Colle, Mario Pickavet
Ghent University - iMinds
Email: sahel.sahhaf@intec.ugent.be

***Abstract*—Traditional routing in the Internet is best-effort. Path differentiation including multipath routing is a promising technique to be used for meeting QoS requirements of media-intensive applications. Since different paths have different characteristics in terms of latency, availability and bandwidth, they offer flexibility in QoS and congestion control. Additionally protection techniques can be used to enhance the reliability of the network.**

This paper studies the problem of how to optimally find paths ensuring maximal bandwidth and resiliency of media transfer over the network. In particular, we propose two algorithms to reserve network paths with minimal new resources while increasing the availability of the paths and enabling congestion control. The first algorithm is based on Integer Linear Programming which minimizes the cost of the paths and the used resources. The second one is a heuristic-based algorithm which solves the scalability limitations of the ILP approach. The algorithms ensure resiliency against any single link failure in the network.

The experimental results indicate that using the proposed schemes the connections availability improve significantly and a more balanced load is achieved in the network compared to the shortest path-based approaches.

***Index Terms*—Multipath routing; Availability; MPLS tunnels**

I. Introduction

Live streaming is becoming very popular as more enterprises stream on the Internet. Examples include radio and television broadcasts, sport events and multimedia conferencing. Based on the industry forecasts, multimedia streaming over packet-switched IP networks will be the dominant Internet traffic in 2019 (80 to 90 %) [1]. Besides, both broadcasters and end-users expect high-quality live viewing experience which is comparable with high-definition (HD) television broadcast. However, traditional routing in the Internet is best-effort and often suffers from different network impairments such as packet loss, jitter and outages of unknown duration without any QoS guarantees. Path differentiation techniques, multipath routing in particular, seems to be a promising method to meet QoS requirements of media-intensive applications.

Our contribution. In this work, we propose algorithms for network path reservation over the Internet. We use multipath routing to meet QoS requirements of media applications, enable congestion control, enable protection against failures and enhance network reliability.

We first propose an Integer Linear Programming (ILP) model to find optimal multipaths for media transfer. This model assigns a cost to each network link which is defined based on the combination of: i) link's availability and ii) link's available bandwidth. Component's availability refers to the probability that the component is in a functional state at any arbitrary time which is a significant QoS metric for describing reliability. The objective is to minimize the cost of the paths reserved for each media transfer. The model finds two sets of paths as primary and secondary. The paths within each set are not necessarily edge-disjoint. However, the primary and secondary resources can not be reserved on the same link.

With the large number of variables required in the ILP formulation of the problem, finding the optimal solution might not be feasible in a reasonable time. We therefore propose a heuristic algorithm to overcome the scalability limitation of the ILP solution. Similar to the ILP, this algorithm assigns the explained cost to each link and finds least-cost paths as primary and secondary. The algorithm iterates several times to find multiple paths to meet the (bandwidth) demands of the transfer. In order to reduce the number of required secondary paths and the overhead caused by finding two sets of paths, the algorithm tries to minimize the consumed resources by finding secondary paths which fulfill the maximum bandwidth allocated on the edges of the primary paths. Since we target resiliency against single link failure, it is sufficient to find such paths as backup. The proposed scheme differs the existing approaches in the sense that: i) the path selection is based on two metrics: reliability of the connections in terms of availability and available bandwidth of the links, ii) the reserved resources for the secondary paths is reduced. As the paths in each set are not necessarily edge-disjoint, more links can be shared while resiliency against single link failure is guaranteed.

In order to use the proposed schemes we rely on Inter-AS (G)MPLS tunnels [2], [3]. Inter-AS Path Computation (PC) is more challenging than the solutions used within a single domain. The reason is that there is a limitation in the visibility of the Traffic Engineering (TE) information of different domains. We rely on the architectural framework proposed in [4]. In this framework they introduce a 'service plane', working on abstract representations of inter-domain relationships. The functional features of a multi-domain service plane supporting the advertisement of providers network capabilities are modeled. This service plane is not necessarily designed to be extended to the whole Internet. However, it is suitable for a limited number of neighboring providers to jointly offer inter-AS services. This framework relies on a Path Computation Element (PCE)-based method for end-to-end path computation and signaling. The tunnel is signaled across the inter-AS path via the RSVP-TE protocol [3].

The rest of the paper is organized as follows. Section II describes the related work. In section III, we briefly describe the architectural framework proposed in [4] and our extension to advertise availability information. Section IV describes the reliability performance parameters and Section V details the problem statement. Sections VI and VII explain the ILP model and the proposed heuristic algorithm respectively. The performance evaluations of the proposed schemes are reported in Section VIII and finally Section IX concludes the paper.

II. Related Work

The two surveys [5] and [6] detail many of the multipath routing protocols in the current Internet which can be used for Traffic Engineering (TE) and fulfilling the QoS requirements. The authors review several protocols, from application to link and physical layers. There exist several IETF activities in support of inter-domain TE such as [3] and [2]. In addition, several research projects focused on this topic. Some of which are focusing on PCE-based framework. [7] provides a survey on the PCE architecture. The Dragon project, [8], relied on this architecture to implement multi-domain TE paths. There are several proposals for BGP extensions to advertise TE information [9], [10]. The main challenge in such approaches is that they require changing BGP. Furthermore, exchanging TE information in addition to the reachability information in BGP limits the scalability.

Other studies focused on multipath provisioning in overlay networks [11]. The combination of overlay architecture and BGP extensions was proposed in [12]. A complementary approach is the peer-to-peer (P2P) approach. Although P2P live streaming is known to be cost-effective, it is not clear whether it can provide the same level of scalability and QoS as provided by dedicated overlay networks. In this context, authors in [13] proposed an architecture referred to as P4P (Proactive Provider Participation for peer-to-peer applications) which enables an efficient cooperation between applications and network providers. Such an architecture enables efficient resource allocation and better performance. [14] surveys several algorithms and mechanisms considered in P2P overlay networks.

In a more recent work, authors proposed solutions to maximize the availability-weighted capacity in elastic optical networks [15]. Since the focus of their work is on elastic optical networks the constraints are quite different than the ones considered int this work for inter-domain settings.

In this work, we consider a PCE-based platform relying on a user-centric model proposed in [4] which is restricted to some providers willing to cooperate for inter-AS services. We briefly describe this model in Section III and the rest of the paper focuses on finding multiple paths fulfilling service demands.

III. Architectural framework

In [4], a provider alliance linked by a common service plane is considered. An inter-AS tunnel request starts at the source domain asking for the computation of inter-domain paths considering the cost and the service requirements. The involved providers agree on a route which is then passed to the PCE-based control planes to compute the MPLS tunnel.

It is assumed that a service broker which is responsible for managing the inter-AS tunnel transactions is independent of the provider. The different entities in this framework include: i) AS Selection Agent (ASA), which is responsible for inter-AS route calculation. The ASAs receive requests, query the service repository, make selections and check if the service can be instantiated. Once the service is activated, the source ASA triggers the path computation at the PCE-based control plane. The tunnel is signaled across the inter-AS path via the RSVP-TE protocol [3]. ii) Network Service Broker (NSB), which is a centralized entity providing for ASAs: Partial Internet Topology (PIT), with ASes as nodes and interconnection between ASes as links, Transit Capabilities and Costs (TCC) of every AS and statistics about previous requests.

We modify the TCC model to include the availability information between ASes and the directional capabilities between the adjacent ASes. These capabilities can be function of different QoS requirements (e.g., minimum bandwidth).

We rely on the availability model of a bidirectional line proposed in [16]. Based on this model, fiber optic cable is the dominant component since the cable cuts are frequent and repair times are very long. Therefore, the availability of a line is dependent on its length. Accordingly the availability of the connection between two ASes can be estimated based on its length. To this end, we require an estimation on the physical distance between adjacent ASes. We propose to use 2 databases: i) GeoLite ASN which provides the IP addresses to AS number mapping and ii) GeoLite City which provides IP addresses to geolocations mapping. Using these databases, the approximate location of the ASes can be found which can be used to estimate the physical distance between them.

There are some limitations in this approach: i) ASes might be connected to each other through Internet Exchange Points (IXP), while in PIT, these ASes are considered to be adjacent, ii) It is assumed that the extracted locations represent the geolocation of the border routers directly connected between adjacent ASes. As part of the future work, we will focus on solving these limitations to have a more accurate calculation.

We refer the interested readers to [4] for detail description of this framework and the corresponding functional architecture.

IV. Reliability performance parameters

We detail the reliability performance parameters which are used in the design and evaluation of the proposed schemes.

A. Component availability

The probability that a component is functional at an arbitrary moment is called availability of that component. Availability is calculated based on: i) components mean time to repair (MTTR) which is the time needed to restore a component and ii) mean time between failures (MTBF) which is defined as the time between two consecutive failures of the component [16]. The availability A is defined as follows:

$$A = 1 - \frac{MTTR}{MTBF} \tag{1}$$

B. Protected/unprotected path availability

The availability of a path is calculated based on the availability of the components in the network. For an unprotected path, all the nodes and links along that path should be available in order to have an available path. This is defined as:

$$A_{unprotected} = \prod_{i \in components} A(i) \tag{2}$$

In this formula, $A(i)$ indicates the availability of the ith component along the path. Note that recovery techniques improve the availability of a path because a protected path is available if primary path or the secondary path is available.

$$A_{protected} = A_p + A_p^{'} A_s \tag{3}$$

A_p and $A_p^{'}$ indicate the availability and unavailability of the primary path respectively while A_s represents the availability of the secondary path. This formula can be extended to present the availability of a system, A_{system}. In the following formula, single failure and multiple paths are considered:

$$A_{system} = \prod_{i=1}^{M} A_p(i) + \sum_{j=1}^{M} \left(\prod_{k=1; k \neq j}^{M} A_p(k) \right) A_p^{\prime}(j) A_s(j) \tag{4}$$

The number of working paths in the system is M.

V. Problem Statement

We present a formal model for our proposed reliable multipath provisioning. This model can be used to find multiple paths for a given request to transfer media-intensive applications. The network (PIT) is represented as a graph $G(V, E)$ with ASes as vertices V and the set of inter-AS logical connections as edges E. Each edge has a certain capacity in terms of bandwidth B_e which is the residual capacities after the previously admitted requests. The ASA of the source domain associates to each edge of the PIT a cost $cost_e$ which is calculated based on the availability A_e and the available bandwidth B_e of that edge retrieved from the TCC:

$$cost_e = -ln(A_e) + \alpha \cdot B_e^{-\beta} \;\; \alpha, \beta \in \Re^+ \tag{5}$$

We take the availability of the links into account to enhance the reliability of the connections. In our proposed schemes, we target minimizing the cost of the paths. Since product of the components' availability should be considered when calculating a path availability (see Formula (2)), we used the *log* of the availability in the cost function. Additionally we consider the available bandwidth of the edges to give higher cost to the edges with less bandwidth. This way we prefer paths with more available bandwidth which enables congestion control. α and β are defined to tune the impact of the factors in the cost function.

In this model a request is defined as a tuple $r = (s^r, d^r, b^r)$ in which s^r and d^r are the source and destination of the request respectively. b^r indicates the bandwidth demand of this request. Since requests arrive over time, we do not know about the future requests. Therefore, we try to find paths for each request optimally by: i) using fewer new resources, ii) using less popular edges by giving higher cost to the edges with less bandwidth and iii) using edges with higher availability.

We use protection techniques to further enhance the paths reliability. Therefore, two sets of paths should be found. The first set includes the primary paths and the second set comprises the secondary paths. These two sets should be edge-disjoint however, the paths within each set can share links among themselves. This increases the chance of finding paths to fulfill QoS requirements/bandwidth demand of a request. To decrease the resource consumption, instead of finding secondary paths which fulfill the whole bandwidth demand of a request, we reserve bandwidth equal to the maximum bandwidth assigned on the links of the primary paths.

Table I lists all the notations used to define the model.

TABLE I
SYMBOLS AND NOTATIONS IN THE FORMAL MODEL

Variable	Description
V	Set of physical vertices.
E	Set of physical edges.
B_e	Bandwidth capacity of link e.
$cost_e$	The cost of link e.
A_e	The availability of link e.
E_v^{out}	Set of all outgoing edges of node v.
E_v^{in}	Set of all incoming edges of node v.
R	Set of concurrent requests.
s^r	Source node of request r.
d^r	Destination node of request r.
b^r	Bandwidth demand of primary paths for request r.
b_s^r	Decision variable. Bandwidth demand of secondary paths for request r.
$x_p^{r,e}$	Binary decision variable. 1 if link e is used in the primary paths of request r, 0 otherwise.
$x_s^{r,e}$	Binary decision variable. 1 if link e is used in the secondary paths of request r, 0 otherwise.
$\beta_p^{r,e}$	Decision variable. Dedicated bandwidth for primary paths between link e and request r.
$\beta_s^{r,e}$	Decision variable. Dedicated bandwidth for secondary paths between link e and request r.

VI. ILP-based resilient multipath routing

In this section, we detail an ILP formulation for the explained problem.

A. Decision variables

In our ILP model two sets of decision variables corresponding to primary and secondary paths are required. $x_p^{r,e}$ is a binary decision variable which indicates whether an edge e is used in the primary paths of request r. $x_s^{r,e}$ represents the usage of the edge e in the secondary paths of request r.

$$x_p^{r,e}, x_s^{r,e} \in [0, 1] \; \forall r \in R, \forall e \in E$$

$\beta_p^{r,e}$ is a decision variable which indicates the amount of bandwidth used on edge e for the primary paths of request r and $\beta_s^{r,e}$ is the corresponding decision variable for the secondary paths.

$$\beta_p^{r,e}, \beta_s^{r,e} \in \Re^+ \; \forall r \in R, \forall e \in E$$

b_s^r is a decision variable which determines the amount of bandwidth to be reserved for the secondary paths.

$$b_s^r \in \Re^+ \; \forall r \in R$$

B. Objective function

The objective is to find two sets of optimal primary and secondary paths, where optimal refers to the minimum total cost of the paths. The cost of the network edges are defined based on their availability and residual bandwidth (Formula (5)) and minimizing the total cost leads to paths with fewer resources and higher availability while the unpopular edges are preferred. The objective function is defined as:

Minimize

$$\sum_{r \in R} \sum_{e \in E} cost_e \times x_p^{r,e} + \sum_{r \in R} \sum_{e \in E} cost_e \times x_s^{r,e} \qquad (6)$$

C. Flow constraints

Since requests are routed over a network, they are subject to capacity and network flow constraints. The capacity constraint in Formula (7) ensures that the cumulative bandwidth reservation over each edge does not exceed its capacity.

$$\sum_{r \in R} \beta_p^{r,e} + \sum_{r \in R} \beta_s^{r,e} \leq B_e \; \forall e \in E \qquad (7)$$

The next two constraints set the flow conservation for both primary and secondary paths. These constraints ensure that the incoming flow of the intermediate nodes along a path equals the outgoing flow.

$$\sum_{e \in E_v^{out}} \beta_p^{r,e} = \sum_{e \in E_v^{in}} \beta_p^{r,e} \; \forall r \in R, \forall v \in V | v \notin s^r, d^r \qquad (8)$$

$$\sum_{e \in E_v^{out}} \beta_s^{r,e} = \sum_{e \in E_v^{in}} \beta_s^{r,e} \; \forall r \in R, \forall v \in V | v \notin s^r, d^r \qquad (9)$$

The Formulas (10)-(13) ensure that for both primary and secondary paths, the outgoing flow of the source and destination are equal to the request's demand and 0 respectively.

$$\sum_{e \in E_{s^r}^{out}} \beta_p^{r,e} = b^r, \sum_{e \in E_{s^r}^{out}} \beta_s^{r,e} = b_s^r \; \forall r \in R \qquad (10)$$

$$\sum_{e \in E_{d^r}^{out}} \beta_p^{r,e} = 0, \sum_{e \in E_{d^r}^{out}} \beta_s^{r,e} = 0 \; \forall r \in R \qquad (11)$$

Also the incoming flow of the source and destination should be equal to 0 and the request's demand respectively.

$$\sum_{e \in E_{s^r}^{in}} \beta_p^{r,e} = 0, \sum_{e \in E_{s^r}^{in}} \beta_s^{r,e} = 0 \; \forall r \in R \qquad (12)$$

$$\sum_{e \in E_{d^r}^{in}} \beta_p^{r,e} = b^r, \sum_{e \in E_{d^r}^{in}} \beta_s^{r,e} = b_s^r \; \forall r \in R \qquad (13)$$

To ensure that the reserved resource for the secondary paths is equal to the maximum bandwidth assigned on the links of the primary paths, we add the following constraint.

$$b_s^r \geq \beta_p^{r,e} \; \forall e \in E, \forall r \in R \qquad (14)$$

Finally, we use the next two constraints to connect the two binary decision variables $x_p^{r,e}$ and $x_s^{r,e}$ to $\beta_p^{r,e}$ and $\beta_s^{r,e}$ respectively. These constraints ensure that if $\beta_p^{r,e}$ ($\beta_s^{r,e}$) is more than 0, then $x_p^{r,e}$ ($x_s^{r,e}$) is 1 as well.

$$\beta_p^{r,e} \leq x_p^{r,e} \times B_e \; \forall r \in R, \; \forall e \in E \qquad (15)$$

$$\beta_s^{r,e} \leq x_s^{r,e} \times B_e \; \forall r \in R, \; \forall e \in E \qquad (16)$$

D. Disjointness constraints

To ensure that the set of primary and secondary paths do not share any link, we add the following disjointness constraint. This constraint guarantees that an edge is either used in the primary or in the secondary paths of a request.

$$x_p^{r,e} + x_s^{r,e} \leq 1 \; \forall r \in R, \; \forall e \in E \qquad (17)$$

VII. Availability and Bandwidth-aware Multipath Routing algorithm (ABMR)

Due to the large number of constraints and variables in the ILP formulation, the optimal solution may not always be found in a reasonable time in large-scale settings. To have a more scalable solution, we propose a heuristic-based approach (ABMR).

In this work, the disjointness is only required between primary and secondary paths. The paths used as primary (or secondary) can share links among themselves. The proposed scheme guarantees resiliency against any single link failure.

We assume that requests arrive over time and thus are addressed sequentially. If multiple requests arrive at the same time, they are sorted first and then addressed. The ABMR algorithm is shown in Algorithm 1. In this algorithm, first the requests are sorted based on their bandwidth demand in a descending order. For each request, the network resource usage is maintained as depicted in Algorithm 1. The requests are then given to the BWallocationResilient algorithm which finds the close to optimal paths for each request. If a feasible set of paths for both primary and secondary paths is found, the network is updated based on the new allocation, otherwise the algorithm backtracks to the previous state of the network.

The BWallocationResilient is the main module of the ABMR algorithm which is responsible for finding multipaths as primary and secondary. This algorithm is shown in Algorithm 2. In this algorithm first a cost is assigned to each network edge using a cost allocation module. This cost is calculated based on Formula (5). Then the request is given to the BWallocation module together with the network topology

Algorithm 1: ABMR

```
Data: requests, network infrastructure
sortedList← sortBandwidth(all concurrent requests);
for request ∈ sortedList do
    currentState← save the current network state;
    feasible ← BWallocationResilient(request);
    if feasible then
        update the network;
    else
        set current network state to currentState;
    end
end
```

(PIT) with residual capacities. BWallocation finds a first set of paths fulfilling the request's demand. We refer to the paths found by this algorithm as 'primary paths'. Then the links used in the primary paths are removed from the network and BWallocation is used for the second time to find the secondary paths for that request. Therefore, the secondary paths are edge-disjoint from the primary paths but they can share links among themselves. As mentioned earlier, for the secondary paths, it is not required to find flows with exact amount of bandwidth as in the primary paths. Since, we aim at resiliency against single link failure, it is sufficient to find secondary paths which fulfill the maximum bandwidth (maxBW) allocated on the edges of the primary paths. With such secondary paths, if any link in the primary paths fails, there is sufficient capacity for the recovery of the affected paths. In practice, it is possible that a request does not ask for 100% recovery of the bandwidth demand upon a failure occurrence. It might be enough that a portion of the demand is transferred to the destination. Therefore, we compare the maxBW and the requested backup capacity and select the minimum value as the amount of backup capacity to be allocated to that request. This is indicated in Algorithm 2. If a feasible set of primary and secondary paths are found the request is admitted and the network is updated according to the new allocations otherwise the request is rejected.

Algorithm 2: BWallocationResilient

```
Data: a request
costAllocation(edges);
primaryReservation ← BWallocation(req,
primaryDemand, graph);
maxBW ← max Bandwidth(primaryReservation);
graphReduced ← remove the links in primaryReservation
from the network graph;
backupDemand ← min(maxBW,backupDemand(req));
backupReservation ← BWallocation(req, backupDemand,
graphReduced);
if primaryReservation && backupReservation then
    return primaryReservation, backupReservation;
else
    return false;
end
```

Algorithm 3: BWallocation

```
Data: a request, demand, graph
path ← LeastCostPath(graph);
while path do
    minBW ← min Bandwidth(path);
    flow ← flow + minBW;
    if flow ≥ demand(req) then
        minBW ← minBW - (flow - demand(req));
        reservation(req, flow);
        return reservation;
    else
        augment flow along path with minBW
        update the residual graph;
    end
    path ← LeastCostPath(graph);
end
return false;
```

It is worth mentioning that the algorithm does not prefer multipaths over single path. By preferring the paths with more bandwidth, the algorithm tries to find a single path for each set first and only if there is not enough capacity in a single path to fulfill the request's demand, multipaths are searched for. The reason for this preference is to reduce traffic splitting of media transfer, since dealing with out-of-order packets caused by mutlipath solutions is a challenge on its own.

The BWallocation algorithm is shown in Algorithm 3. This algorithm successively finds shortest augmenting paths in the residual network, and augments flow along such paths until the total flow fulfills the demand. The shortest paths are found based on a modified version of the Bellman-Ford algorithm. In this modified algorithm the calculated costs (based on availability and bandwidth) are used as weights.

Note that the proposed algorithms (both ILP and heuristic solutions) guarantee a single link failure recovery. However multiple simultaneous link failures in the primary paths can also be recovered as long as the sum of the bandwidth of the affected paths does not exceed the backup capacity reserved on the secondary paths.

VIII. Performance evaluation

In this section, we first describe the simulation environment and introduce the performance metrics and then the evaluation results are reported. The focus of the experiments is on quantifying the added value of considering available bandwidth and availability in the calculation of the paths in terms of network reliability, load balancing and acceptance rate. In the simulations, we compare the heuristic-based approach (ABMR) with the ILP-based algorithm in network scenarios where ILP can be executed in a reasonable time. Additionally we compare the proposed schemes with multipath routing based on shortest path algorithm.

A. Simulation environment

Our simulation is based on Python code. We used Networkx library for graph-based implementations. The PuLP LP mod-

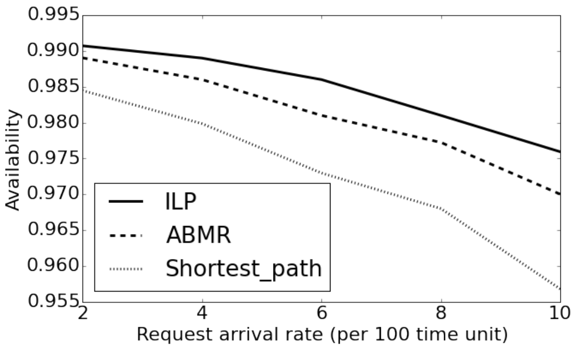

Fig. 1. Comparison of connection availability. Network size = 20.

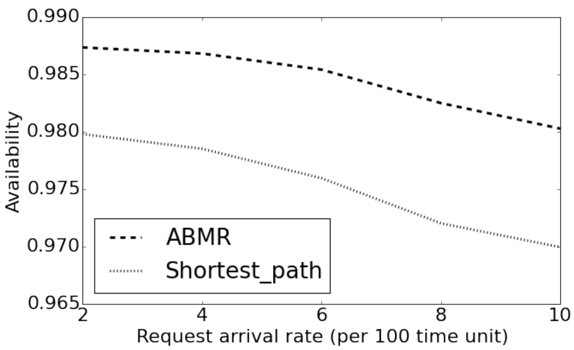

Fig. 2. Comparison of connection availability. Network size = 100.

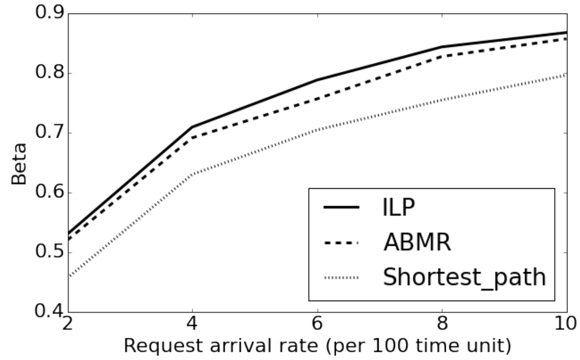

Fig. 3. Comparison of β-ratio. Network size = 20.

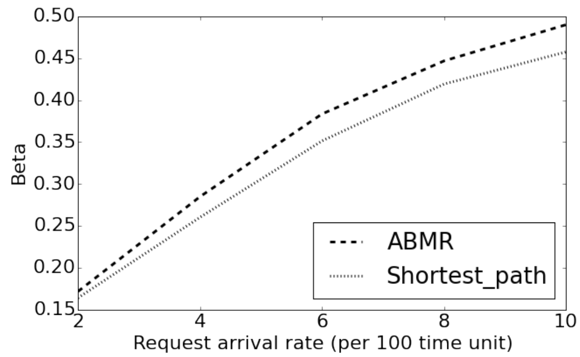

Fig. 4. Comparison of β-ratio. Network size = 100.

eler in Python is used to generate the LP files. The ILP model is solved using the Cbc solver included in PuLP[1].

For the network topology we considered two scenarios: i) small network and ii) large network. We generated a set of network topologies defined by the Barabasi-Albert (B-A) model [17] which generates random scale-free networks. For the small network, we considered a topology composed of 20 nodes and a topology with 100 nodes was generated for the large network scenario. The bandwidth capacity of the links are numbers uniformly distributed between 100 and 200 and links availability are numbers between 0.9 and 0.99999.

The service requests arrive over time in a Poisson process with average rates between 2 and 10 requests per 100 time units. Each request has a lifetime exponentially distributed with an average of $\mu = 1000$ time units. For these requests, the source-destination pair is selected randomly and the bandwidth demand of requests are numbers uniformly distributed between 10 and 50. For the links' cost calculation $\alpha = \beta = 1$.

The hardware which is used for running the simulations is Intel Xeon quad-core CPU at 2.40 GHz with 12 GB RAM. Each simulation is iterated 10 times and the average result over all the iterations is reported.

B. Performance metrics

The following performance metrics are measured to evaluate and compare the proposed schemes.

[1]The solver is from COIN-OR, https://projects.coin-or.org/Cbc.

Availability. This metric indicates the probability that a component is functional at any time (see Section IV).

Load balancing. This metric is defined to objectively measure the load balancing properties of a routing mechanism.

Acceptance ratio. It measures the ratio of the accepted service requests.

C. Availability analysis

We have evaluated the performance of the proposed schemes in terms of availability and compared them with the multipath routing based on shortest-path algorithm. The availability of the paths are calculated based on Formula (4).

Figures 1 and 2 depict the results of the availability evaluation in the two network scenarios. In these figures, the average connection availability achieved for requests in different schemes for different request arrival rates is compared. In both network scenarios, considering the link availability in the path calculation leads to higher connection availability which results into a more reliable network compared to the shortest path-based approach. As we see, for higher arrival rates lower connection availability is achieved. This can be explained by the fact that higher arrival rates lead to existence of more flows at the same time in the network. Accordingly, less options are available for selecting the paths and thus many requests are assigned paths with low availability.

D. Load balancing analysis

In this section, we evaluate the load balancing properties of the proposed schemes. To this end, load balancing metrics have

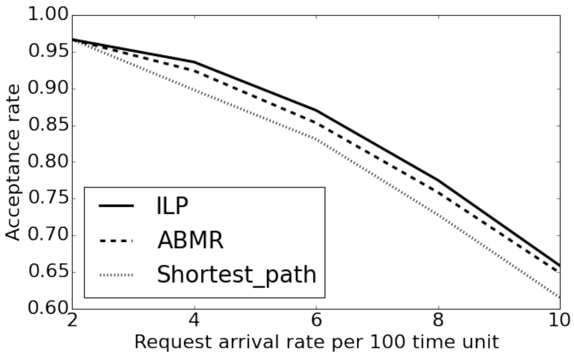

Fig. 5. Comparison of acceptance ratio. Network size = 20.

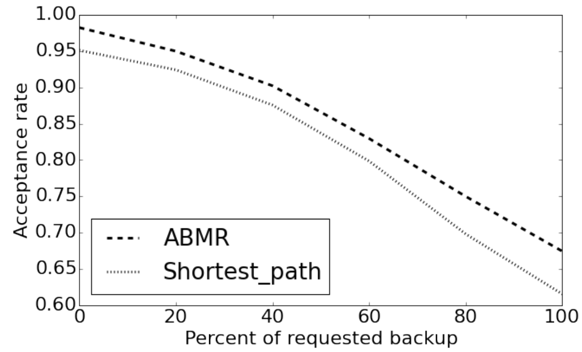

Fig. 7. Comparison of acceptance ratio. Network size = 20.

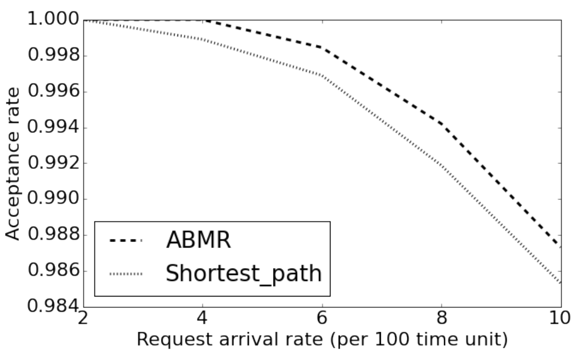

Fig. 6. Comparison of acceptance ratio. Network size = 100.

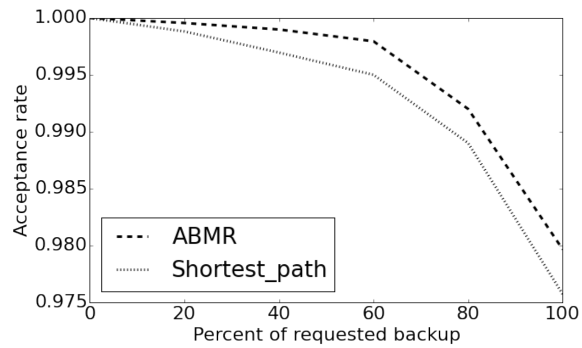

Fig. 8. Comparison of acceptance ratio. Network size = 100.

to be defined. Metrics found in the literature were originally designed to be used for measuring node load balancing, but they can simply be extended to the link load balancing. A metric defined in [18] is called β-ratio. This ratio calculates the traffic distribution over each node/link as:

$$\beta = \frac{(\sum_{x \in X} f_x)^2}{|X| \sum_{x \in X} f_x^2}$$

with X the set of vertices V or edges E of the network $G = (V, E)$ and f_x the number of paths going through vertex or edge x. The β-ratio ranges between $\frac{1}{X} \approx 0$ (unbalanced) to 1 (balanced). Since this ratio has a standard range [0,1] it is easy to interpret.

As we take the available bandwidth of the links into account at the time of finding multipaths, we apply a load balancing mechanism implicitly by giving higher priority to the links which have more bandwidth available. This is visible in the evaluation of the β-ratio as depicted in Figures 3 and 4. As we see, more balanced loads are achieved on the links by the proposed schemes compared to the shortest path algorithm.

E. Impact of request arrival rate and backup requirement

In this section, we evaluate the acceptance ratio of the proposed schemes and compare them with the shortest path algorithm. Figures 5 and 6 depict the results for the two network scenarios. The acceptance rate of the schemes for different request arrival rates are evaluated. Indeed, the increase of the requests arrival rate leads to lower acceptance rate as less resources are available.

Since requests may ask for different backup capacity, in the following experiment, we evaluate the impact of different backup requirements on the acceptance ratio of the proposed schemes. In this evaluation fixed arrival rate of 10 requests per 100 time units is considered and the acceptance ratio for different backup requirement ranging from 0 to 100% is illustrated in Figures 7 and 8. As we see, when large backup capacity is requested the acceptance ratio decreases significantly.

IX. Conclusion

In this paper, we studied the problem of finding optimal multipaths with guaranteed bandwidth and resiliency against a single failure for media transfer over the Internet. We first proposed an ILP-based algorithm taking into account links' availability and bandwidth to increase the reliability of the connections and enable congestion control. To solve the scalability limitation of the ILP, we proposed a heuristic-based approach which finds least-cost paths. The proposed schemes reduce the amount of bandwidth that is reserved as backup while guaranteeing resiliency against any single link failure. The simulation results indicated that the proposed schemes have better performance in terms of acceptance rate, connection availability and load balancing compared to the schemes based on the shortest path algorithm. In order to use the proposed algorithms for multipath provisioning over the Internet, we relied on a PCE-based architecture to create a MPLS tunnel corresponding to each path. This architecture does not scale to the whole Internet and as part of the future work we will focus on more scalable solutions. Another interesting future direction includes considering other factors such as maximum possible delay difference between the paths.

ACKNOWLEDGMENT

The research described in this paper is partially funded by the iMinds EMD project and by the UGent BOF/GOA project 'Autonomic Networked Multimedia Systems'.

REFERENCES

[1] C. V. N. Index, "Forecast and methodology, 2014-2019 white paper," Technical Report, Cisco, Tech. Rep., 2015.

[2] R. Zhang, "Mpls inter-autonomous system (as) traffic engineering (te) requirements," 2005.

[3] A. Farrel, A. Ayyangar, and J. Vasseur, "Inter-domain mpls and gmpls traffic engineering–resource reservation protocol-traffic engineering (rsvp-te) extensions," *Internet Engineering Task Force, RFC*, vol. 5151, 2008.

[4] S. Secci, J.-L. Rougier, and A. Pattavina, "On the selection of optimal diverse as-paths for inter-domain ip/(g) mpls tunnel provisioning," in *Telecommunication Networking Workshop on QoS in Multiservice IP Networks, 2008. IT-NEWS 2008. 4th International*. IEEE, 2008, pp. 235–241.

[5] S. K. Singh, T. Das, and A. Jukan, "A survey on internet multipath routing and provisioning," *Communications Surveys & Tutorials, IEEE*, vol. 17, no. 4, pp. 2157–2175, 2015.

[6] S. Qazi and T. Moors, "Finding alternate paths in the internet: A survey of techniques for end-to-end path discovery," *arXiv preprint arXiv:1310.8125*, 2013.

[7] F. Paolucci, F. Cugini, A. Giorgetti, N. Sambo, and P. Castoldi, "A survey on the path computation element (pce) architecture," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 4, pp. 1819–1841, 2013.

[8] T. Lehman, J. Sobieski, and B. Jabbari, "Dragon: a framework for service provisioning in heterogeneous grid networks," *Communications Magazine, IEEE*, vol. 44, no. 3, pp. 84–90, 2006.

[9] G. Stea and E. Mingozzi, "The euqos system: A solution for qos routing in heterogeneous networks," *IEEE Communications Magazine*, p. 3, 2007.

[10] D. Griffin, J. Spencer, J. Griem, M. Boucadair, P. Morand, M. Howarth, N. Wang, G. Pavlou, A. H. Asgari, and P. Georgatsos, "Interdomain routing through qos-class planes [quality-of-service-based routing algorithms for heterogeneous networks]," *Communications Magazine, IEEE*, vol. 45, no. 2, pp. 88–95, 2007.

[11] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 1, pp. 66–66, 2002.

[12] W. Xu and J. Rexford, *MIRO: multi-path interdomain routing*. ACM, 2006, vol. 36, no. 4.

[13] H. Xie, A. Krishnamurthy, Y. R. Yang, and A. Silberschatz, "P4p: Proactive provider participation for p2p," *Yale Computer Science YALE/DCS/TR1377*, 2007.

[14] A. Malatras, "State-of-the-art survey on p2p overlay networks in pervasive computing environments," *Journal of Network and Computer Applications*, vol. 55, pp. 1–23, 2015.

[15] X. Wang, G. Shen, Z. Zhu, and X. Fu, "Benefits of sub-band virtual concatenation for enhancing availability of elastic optical networks," *Journal of Lightwave Technology*, vol. 34, no. 4, pp. 1098–1110, 2016.

[16] J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Elsevier, 2004.

[17] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[18] H. Velayos, V. Aleo, and G. Karlsson, "Load balancing in overlapping wireless lan cells," in *Communications, 2004 IEEE International Conference on*, vol. 7. IEEE, 2004, pp. 3833–3836.