An Application-Aware Multi-Layer Service Provisioning Algorithm based on Auxiliary Graphs

Marco Savi, Federico Pederzolli, Domenico Siracusa
CREATE-NET, Via alla Cascata 56/D, 38123, Povo, Trento, Italy
{*m.savi, fpederzolli, dsiracusa*}*@create-net.org*

Abstract: A novel application-aware multi-layer resource allocation algorithm is proposed. We demonstrate that it prevents the violation of application requirements (bandwidth, latency, availability, encryption), while keeping blocking probability lower than an existing algorithm.
OCIS codes: (060.4251) Networks, assignment and routing algorithms; (060.4256) Networks, network optimization

1. Introduction

Modern Internet traffic is generated by a multitude of different applications (e.g., video streaming, online gaming, financial transactions, etc.), each with its own requirements in terms of bandwidth, latency, reliability etc. Many works have tried to improve traffic engineering techniques at the IP/Multi Protocol Label Switching (MPLS) packet layer, however such efforts are severely limited by the fact that application traffic with diverse requirements is eventually groomed into the same few large optical connections. For this reason, a more comprehensive approach, able to achieve full service differentiation for applications all the way down to the optical transport layer, is needed.

Such deep differentiation can only be achieved if *i)* the network can learn information about application requirements and *ii)* it is able to steer traffic according to the needs of its originating application (i.e., in an *application-aware* fashion) at both IP/MPLS and optical layers (i.e., in a *multi-layer* fashion). The former information exchange can be enabled by the novel *intent-based networking* paradigm [1], according to which an application specifies, through an intent-based interface, *what* it wants from the network in terms of requirements (i.e., the *intent*) without specifying *how* to achieve it. The network is then responsible for configuring itself by allocating IP/MPLS and optical resources in a convenient way that guarantees the requested service.

In this paper we focus on dynamically allocating resources to *service requests* (SRs, i.e., intents), which arrive and leave over time and are generated by applications with heterogeneous requirements, in a joint multi-layer and application-aware fashion. To do so, we define a novel resource allocation algorithm, which we call *Application-Aware* (AA), able to jointly allocate resources on IP/MPLS and optical layers, while meeting multiple optional application requirements (bandwidth, latency, availability, encryption). The algorithm exploits the concept of *Auxiliary Graphs* (AGs), firstly formalized in [2], and extends the multi-layer AG construction methodology proposed in [3] to provide application-awareness for each SR. Moreover, the algorithm has a modular design, i.e., it can be easily extended to consider further application requirements (ARs). We ran extensive simulations to evaluate our AA Algorithm: results show that it outperforms a *Baseline* (application-unaware) algorithm, based on concepts taken by [3], in terms of both Blocking Probability (BP) and of AR Violation Probability (VP) for the SRs, since it is able to better distribute the application traffic flows across the network according to application needs.

2. Multi-Layer Application-Aware Service Provisioning

We consider a 2-layer physical network with a transparent Dense Wavelength-Division Multiplexing (DWDM) optical layer and an IP/MPLS packet layer. At the optical layer nodes (i.e., reconfigurable optical add-drop multiplexers, ROADMs) are interconnected by fiber links supporting multiple wavelengths, while at the IP/MPLS layer nodes (i.e., IP/MPLS routers) are interconnected by IP adjacencies. Each IP adjacency is realized through a *lightpath* (LP, i.e., a transparent optical connection) in the optical layer. Multiple SRs are generated over time by applications, each requesting connectivity between a pair of IP/MPLS nodes. Each SR r is represented by a $r(s,d,b,l,a,e)$ tuple, where s and d are the IDs of the source and destination IP/MPLS nodes, respectively, b is the requested bandwidth, l the maximum tolerable end-to-end latency, a the minimum tolerable path availability and e specifies whether the application traffic associated to the SR must be encrypted or not. The network must then provide connectivity to the application by computing an *AA* end-to-end *path* on the IP/MPLS layer that satisfies all of b,l,a,e (i.e., the ARs).

2.1. Auxiliary Graph Model

We propose an AG model for computing AA paths, based on the IP/MPLS layer network topology. We start by adding the IP/MPLS nodes to the AG. Then, two types of edges can be added: *existing* and *potential* lightpath edges.

An existing LP has been previously established in the optical layer and already carries some network traffic, while a potential LP represents whether a transparent LP could be established between a certain pair of nodes on the AG given the current state of network resources. The set of potential LP edges that can be added to the AG is computed by running the K_{wdm}-Shortest Path (SP) First-Fit (FF) routing and wavelength assignment algorithm between any AG node pair, considering only the optical resources (i.e., wavelengths) and topology. This way, we define a simple single-layer graph that takes into consideration *i)* the multi-layer nature of the physical network and *ii)* its state in terms of available resources, and can be manipulated to provide application-awareness to SRs, as shown in the next Section.

2.2. Application-Aware Algorithm Description

The flow diagram of our AA Algorithm is shown in Fig. 1(a). It is executed every time a SR $r(s,d,b,l,a,e)$ arrives. Firstly, the algorithm attempts to find an AA path by reusing already-existing LPs. To do so, an AG including only existing LP edges is created. Among these edges, the ones not meeting b, i.e., with available bandwidth $B_{lp} < b$, are pruned. Then, in case of $e = \texttt{true}$, all the *non-encrypted LPs* are pruned as well.

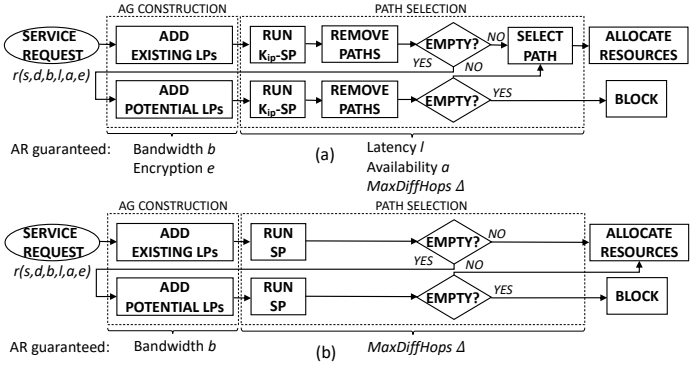

Fig. 1. *AA* (a) and *Baseline* [3] (b) Algorithms flow diagrams.

After constructing the AG, the K_{ip}-SP algorithm between s and d is run on it. The weight chosen for each existing LP edge is its physical length. K_{ip}-SP gives as output up to K_{ip} *candidate paths*, all meeting the b and e ARs. Among such paths, the algorithm prunes the ones with latency $L_{path} > l$ and with availability $A_{path} < a$. Note that $L_{path} = \sum (L_{nodes} + L_{links})$, where L_{nodes} (L_{links}) is the latency added by the traversed packet/optical nodes (links), while $A_{path} = \prod (A_{nodes} \cdot A_{links})$, where A_{nodes} (A_{links}) is the availability of the traversed packet/optical nodes (links). At this point, all remaining candidate paths meet all ARs (b,l,a,e). We also introduce a control parameter called *MaxDiffHops* (Δ). Δ indicates the maximum allowed difference in the overall number of optical hops for a candidate path δ_{path} between s and d with respect to the number of hops of the optical SP δ_{SP}. All the candidate paths with a number of optical hops $\delta_{path} > \delta_{SP} + \Delta$ are pruned. This tunable parameter can help to control the occupation of resources over the optical links, as we will show in the next Section. Among the remaining candidate paths, if any, the first one in the list (i.e., the shortest) is chosen as AA path for the SR r, and the requested bandwidth b is allocated over the traversed existing LPs.

Alternatively, if the set of candidate paths is empty, the algorithm performs a second phase and augments the AG constructed earlier by adding potential LP edges. A potential LP edge between a node pair is added only if no usable existing LP edge already exists between those nodes in the current AG, and there are enough resources to establish it. This way, in the *path selection* phase, we increase the chance of finding an AA path at the expense of having to set up one or more new LPs in the network. If at least one AA path is selected (i.e., it satisfies all ARs), the chosen wavelengths are allocated for the traversed potential LP edges, their state is changed from *potential* to *existing*, they are flagged as *encrypted* if $e = \texttt{true}$, and lastly the requested bandwidth b is allocated over the traversed LPs. Otherwise, if the set of candidate paths is still empty, the SR is blocked.

In the next Section we will compare such *AA Algorithm* with the *Baseline* (Base) algorithm shown in Fig. 1(b). The Baseline algorithm is built upon the algorithm proposed in [3]: it constructs the AG by only considering the bandwidth AR b and it selects the SP (in terms of physical length) between s and d by running the well-known Dijkstra algorithm.

3. Performance evaluation

We implemented and simulated our algorithm in Net2Plan [4]. We used the Telefónica Spain national network topology, with 30 ROADMs/OXCs, 56 bi-directional fiber links (each one carrying up to 80 wavelengths, with a capacity of 100 Gb/s each) and 14 IP/MPLS routers [5]. We considered a realistic (non-uniform) traffic matrix provided by the national provider, where most of the traffic is routed from/to IP/MPLS nodes around the capital city. We also doubled the propagation delay for each fiber to model a larger network. We assumed that each link/node has an availability of 99.9%. SRs are generated according to a Poisson process, with an exponential holding time: once a SR expires, the resources allocated to it are released. We simulated the processing of 10^6 SRs, starting to collect statistics after the first $2 \cdot 10^4$ requests (so as to reach a desired network utilization). For each SR, the ARs b,l,a are uniformly chosen from the following sets: $b = \{1$ Gb/s, 10 Gb/s$\}$, $l = \{15$ ms, $\infty\}$, $a = \{0\%, 99\%\}$. $l = \infty$ ($a = 0\%$) means that the SR is not constrained by latency (availability). Concerning e, we assume that $\Pr(e = \texttt{true}) = 0.01$. An example of SR is $r(s = 1, d = 10; b = 1$ Gb/s$, l = \infty, a = 99\%, e = \texttt{false})$ where $s = 1$ ($d = 10$) is the ID of the source (destination) IP/MPLS router. Concerning the AA algorithm parameters, we chose $K_{wdm} = 5$ and $K_{ip} = 50$.

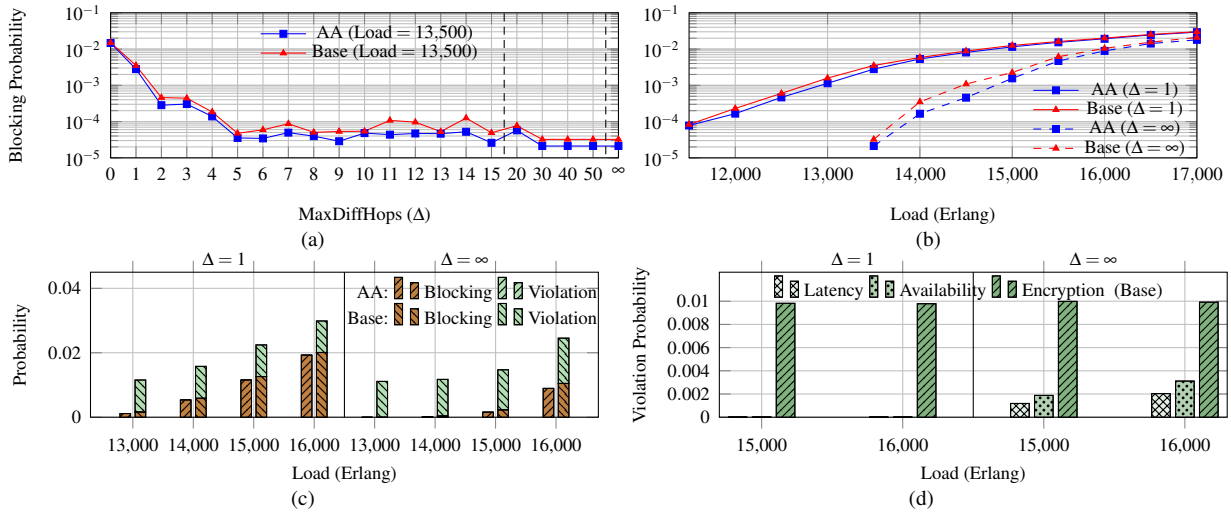

Fig. 2. Performance comparison between the Application Aware (AA) and the Baseline (Base) Algorithm.

Figure 2(a) shows the BP of SRs as a function of MaxDiffHops Δ. Low values of Δ force all the chosen paths to closely match optical SPs. On the one hand, this assures that resources are allocated on a few number of optical links (avoiding an excessive consumption of optical resources), but on the other it limits the number of paths that can be selected. Results show how larger values of Δ, in the long term, reduce BP for both AA and Base. After a certain threshold ($\Delta \geq 30$) the exhibited BP stabilizes to a minimum value. We then focus our evaluation on $\Delta = 1$ and $\Delta = \infty$. Figure 2(b) shows the BP as a function of the network load, expressed in terms of average number of SRs in the network (Erlang). It shows how AA performs slightly better than Base for any load condition and how the BP difference between AA and Base is lower for $\Delta = 1$ than for $\Delta = \infty$, since the overall number of paths that can be selected is lower in the former case and thus AA and Base behave similarly in terms of chosen paths.

Figures 2(c) and 2(d) focus on another evaluation metric, i.e., VP. This metric evaluates the probability that the path provided to a SR violates one or more ARs. Figure 2(c) shows how VP, for AA, is zero for any traffic load and for any value of Δ, since the algorithm has been designed to always meet *all* the ARs (and to block the SR if this is not possible). Conversely, Base always leads to high VPs, since it allocates resources in an application-unaware manner. It is also possible to notice how $\Delta = 1$, for any load condition, leads to a lower VP than $\Delta = \infty$, while BP behaves in the opposite manner, as explained above. To understand why, we look at Fig. 2(d), that shows the breakdown of VP for the different ARs. We can see that the encryption VP is around 0.01 in all cases. This means that almost all the SRs requiring encryption have such AR violated, since Base does not distinguish among encrypted and non-encrypted LPs, and the probability of having a path crossing one (or more) non-encrypted LPs is around 1. Likewise, we can see how no SR violates latency and availability constraints in case of $\Delta = 1$, while some violation occurs in case of $\Delta = \infty$. This can be explained by the fact that in the latter case longer paths can be chosen (especially at higher loads, where the shorter ones might all be busy) and thus a higher number of nodes/links can be traversed, leading to higher latencies and lower availabilities. This does not happen for AA, since longer paths are still allowed by Δ, but those not meeting latency and availability ARs are pruned a priori before path selection (in the case of latency and/or availability-constrained SRs).

4. Conclusion

We proposed a novel multi-layer dynamic algorithm to allocate resources for service requests in an application-aware manner, i.e., while taking multiple application requirements into account, based on auxiliary graphs. We proved, using simulations, that it achieves better performance than a state-of-the-art application-unaware algorithm, preventing all application requirement violations while exhibiting no worse blocking probability (slightly better, in fact).

Acknowledgment: this study has received funding through the EU ACINO project, grant agreement n. 645127.

References

[1] M. Pham *et al.*, "SDN applications - the intent-based northbound interface realisation for extended applications," Proc. IEEE NetSoft (2016).
[2] H. Zhu *et al.*, "Dynamic traffic grooming in WDM mesh networks using a novel graph model," Proc. IEEE GLOBECOM (2002).
[3] J. Zhang *et al.*, "Dynamic virtual network embedding over multilayer optical networks," IEEE/OSA JOCN (2015).
[4] P. Pavon-Marino *et al.*, "Net2Plan: an open source network planning tool for bridging the gap between academia and industry," IEEE Network (2015).
[5] F. Rambach *et al.*, "A multilayer cost model for metro/core networks," IEEE/OSA JOCN (2013).