

Application-Centric Dynamic Multi-layer Resource Allocation in Availability-aware SDN-Orchestrated Networks

Ćiril Rožić⁽¹⁾, Marco Savi⁽²⁾, Chris Matrakidis⁽¹⁾,
Dimitrios Klonidis⁽¹⁾, Domenico Siracusa⁽²⁾, Ioannis Tomkos⁽¹⁾

⁽¹⁾ Athens Information Technology (AIT), Marousi 15125, Greece, croz@ait.gr

⁽²⁾ Fondazione Bruno Kessler (FBK), CREATE-NET Research Center, Povo, Trento, Italy

Abstract We evaluate our method for dynamically accommodating applications that require more than just bandwidth. Our simulations show the advantages of two availability-aware approaches on a network equipped with an SDN orchestrator.

Introduction

Many network applications today want more than just connectivity and a bandwidth pipe. Examples include autonomous car driving, remote surgery, etc. These require guarantees such as maximum end-to-end latency, component availability, and failure resilience. Accommodating these *service requirements* can be critical, and network providers ignoring them may miss an opportunity for extra revenue¹.

In EU project ACINO², service requirements are dealt with by exploiting the paradigms of intent-based networking and SDN to build a network orchestrator. The orchestrator has a very complete view of the network infrastructure and running services. It receives application intents translated to service requirements, computes a path through the network, and requests actions from the underlying IP and optical controllers. The service requirements are handled not only at the packet (IP/MPLS) layer, as is conventional, but also by the optical layer. The applications thus drive the multi-layer operation of the network in an automated fashion.

To that end, we have devised and evaluated a network resource allocation framework^{3,4} (Fig. 1). Briefly, upon receiving the service requirements, the framework first attempts to establish a connection in the IP/MPLS layer, and if that fails, moves the running IP/MPLS connections around to make space for the new request. If that fails, new lightpaths (IP links) are established. The framework thus prefers to set up the connection quickly, in IP/MPLS, and the slower optical operations are avoided unless necessary. In all cases, accommodating a new connection request must not affect any of the running services.

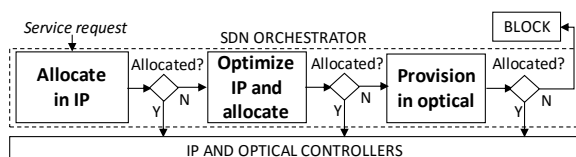


Fig. 1: ACINO resource allocation network.

In this paper, we use the framework to show how the application-centric approach can be used to the benefit of both the applications and the service provider. The difference to study³ is that the orchestrator now interacts more intensively with the application, and that we use network path availability (with respect to component mean times to failure and repair) as the constraining service requirement.

Network-application communication

The application specifies, via its intent, a certain bandwidth and possibly availability. If the orchestrator can find a path through the network, it sends a request to the controllers. Otherwise, if the availability is too constraining, we introduce the option of attempting to find a backup path to be set up together with the primary. The combination of the availabilities* of the primary and backup paths, implemented with mechanisms such as IP/MPLS Fast Reroute⁵, can give the required availability. The backup uses 1+1 reserved bandwidth and thus needs extra network resources (such as IP ports). The orchestrator now notifies the application that the request can be met only with an additional effort. If the application agrees, it may be charged a premium. We refer to this procedure as *aware-retry*.

As a comparison, the *aware* approach is the same as in study³, i.e. it does not attempt to set up a backup path. As a least intelligent approach, *unaware* uses the multi-layer framework, but the network completely ignores the availability requirement. Since the primary connection paths are less constrained and there are no backup paths, the network is expected to utilise less resources. However, unaware cannot charge a premium for the applications running on network paths with an unsatisfactory availability, and the operator may be liable for contract violations. We thus have two closely related benchmarks to compare to the most application-centric approach that is aware-retry.

*For two paths with availabilities A_1 and A_2 , their combined availability is $1-(1-A_1)(1-A_2)$.

Scenario setup

Our test network is the same as in study³, i.e. the Spanish national core network. The optical fibre network consists of 30 optical switches and 112 fibre-links. We limit our use of the network to one fibre per fibre-link, and assume 80 wavelengths per fibre. We assume no wavelength conversion in the network, so the same wavelength must be used on all fibre-links in each lightpath. Kilometre distances for each fibre-link were obtained. We assume no regeneration is needed, which is reasonable given the lightpath lengths produced by the simulation described below. The IP network has 14 routers co-located with 14 of the optical nodes. We start with no IP links present in the network initially, and let the framework algorithm create and remove them.

To obtain the availability figures for fibres, we started from a target value of 2.5 failures per 1000 kilometers per year⁶, adjusted for the fibre length of each link to get the average MTBF, and set MTTR to 8 hours⁶. For the routers and optical nodes we used the typical availability of 99.999% or *five nines*.

The input traffic matrix was obtained from the network operator. It contains the average traffic volumes in Gb/s for each pair of routers. Our dynamic traffic generator maintains these volumes while scheduling the traffic for each *traffic class* separately. Each class is characterized by the percentage of the total traffic, as well as its service requirements (in this paper bandwidth and availability). The arrival-departure of connection requests is a Poisson process, where each connection has a random uniformly distributed bandwidth between 10 and 100 Gb/s. In this paper the traffic classes are *sensitive* and *best effort*, where the former requires an availability, and the latter does not.

To find a path for a request, for the framework parameters we use 100Gb/s transceivers over WDM with 2000 km reach, $K=50$ and $K=5$ for the well-known *K shortest path* subroutines in the IP and optical, respectively. IP links carrying no traffic are removed every 10 IP optimizations (i.e. middle boxes in Fig. 1).

Results

To evaluate the three approaches, we conduct network simulations using Net2Plan⁷, and measure *blocking* and *violations*. An application is blocked, and no path is set up, when the requirements cannot be met. The requirements are violated when a path is set up with availability lower than required. In the case of aware and aware-retry, an application will be

blocked if either the required bandwidth or the availability requirements cannot be met, but they may never violate. Unaware may violate only availability, but not bandwidth; if it cannot meet the bandwidth requirement, it will block. Consequently, only the sensitive class will cause violations, and only in unaware. In our tests, no blocking was observed with unaware.

We vary the required availability between two and five nines. Such figures are well known in the context of network component availability. The amount of availability-sensitive traffic is varied as well.

Fig. 2 shows that on our test network all three approaches have little blocking or violations for up to three nines service availability. At three nines, the aware approach shows some savings compared to the unaware one. At four and five nines, there is little difference between the two.

The aware-retry approach can carry all traffic with no blocking up to and including four nines. Five nines result in about equal blocking for all three approaches, and the amount of blocked is about equal to the amount of sensitive traffic. This is expected, because the five nines cannot be achieved if the paths in the test network consist of links with no more than three nines, as is the case here.

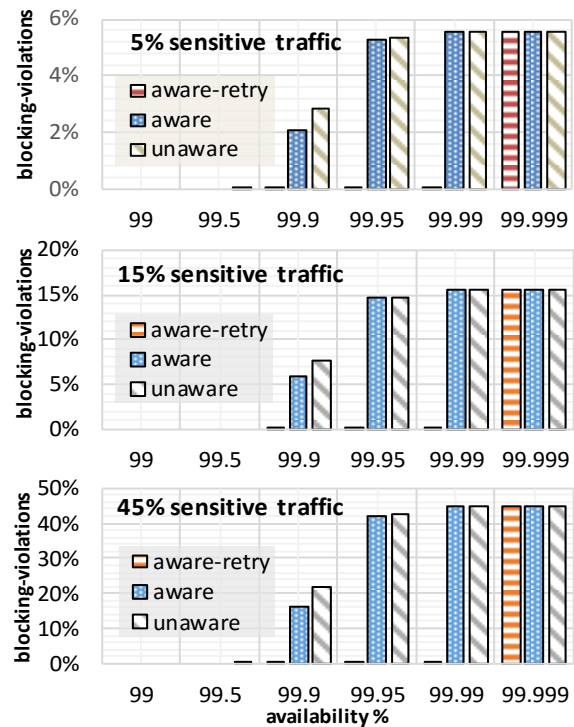


Fig. 2: Fractions (out of the total traffic) of blocked and violated traffic bandwidth for different service availabilities.

Fig. 3 shows the advantages of the two aware approaches. Aware-retry does not block, and aware blocking is lower than unaware service requirement violations.

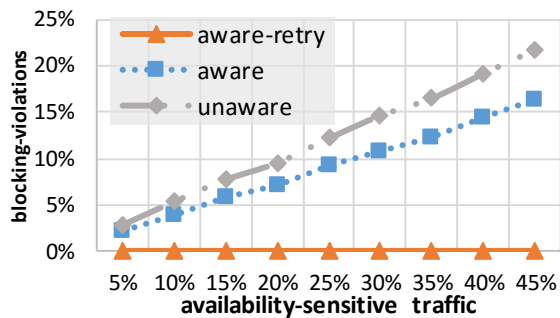


Fig. 3: For three nines availability, fractions (out of the total traffic) of blocked and violated traffic bandwidth for different fractions of sensitive traffic bandwidth.

Fig. 4 (top) shows, for a fixed percentage of sensitive traffic (15%) and availability (99.9%) the network resource usage measured as the number of IP links (lightpaths) over time. After the ramp-up phase, aware-retry uses significantly more links, while aware is reasonably close to unaware. The network can thus support sensitive traffic for little extra resources. However, the near-zero blocking shown in Figs. 2 and 3 comes at the cost of extra IP links for aware-retry, which is due to the extra bandwidth needed for the backup paths.

In Fig. 4 (bottom) we measure how often optical connections, i.e. links in IP, are added to the network, using unaware as the baseline. As expected, both aware and aware-retry add more links than the baseline throughout the simulation run. However, in this case, aware adds lightpaths more intensively than aware-retry. This is because aware-retry uses more IP links, as shown in Fig. 4(top). The IP link bandwidth can be shared by several applications**, so there is no need to set up new links as often. Because optical set ups are relatively slow, the extra resources for aware-retry are traded off for faster application admission. This behaviour is only seen for average percentages of availability sensitive traffic, e.g. around 15% as shown here. For low percentages, say 5%, the added paths are too few to make a difference. In contrast, for percentages of more than 30%, the backup paths dominate so links have to be added more aggressively.

Conclusions

We used the ACINO resource allocation and optimization framework algorithms, the Net2Plan simulation engine, and real network data to investigate how two application-centric approaches compare to a similar baseline in a dynamic multi-layer setting. Aware and retry-aware show less blocking than unaware violations. The trade-off is that aware-retry requires more network resources and activity.

**Recall that our transceivers are 100Gb/s and the application bandwidths are between 10 and 100Gb/s.

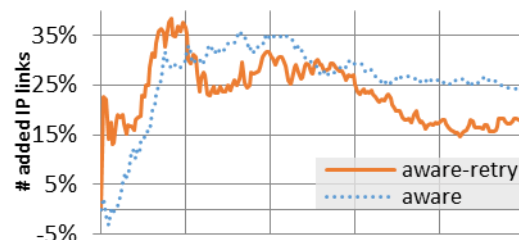
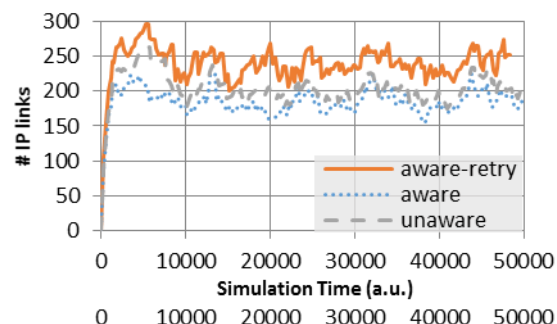


Fig. 4: For three nines availability and 15% sensitive traffic (top) total IP links (lightpaths) in the network; (bottom) extra lightpaths added by the availability-aware network orchestrator, with the unaware approach as the baseline.

However, this in turn allows for almost zero blocking, as well as a more responsive optical layer. Instead of such an all-or-nothing choice, the operator may favour the aware approach as a balance between network resources, responsiveness, and blocking.

Acknowledgements

The research leading to these results has received funding from the European Commission within the H2020 Research and Innovation programme, ACINO project, Grant Number 645127, www.acino.eu. We thank Dr. Victor Lopez of Telefonica I+D for providing the network data.

References

- [1] R. Farahbakhsh et al., "How far is Facebook from me? Facebook network infrastructure analysis," *IEEE Communications Magazine*, vol. 53, no. 9, 2015.
- [2] ACINO website. [Online]. Available: <http://www.acino.eu/>
- [3] Č. Rožić et al., "A framework for dynamic multi-layer resource allocation and optimization in application-centric networking," *OFC 2017*.
- [4] V. Lopez et al., "On the benefits of multi-layer optimization and application awareness," *J. Lightwave Technol.*, vol. 35, no.6, pp. 1274–1279, 2017.
- [5] P. Pan et al., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.
- [6] Alcoa Fujikura, "Reliability of Fiber Optic Cable Systems: Buried Fiber Optic Cable Optical Groundwire Cable all Dielectric, Self Supporting Cable", May 2001.
- [7] P. Pavon-Marino, J. Izquierdo-Zaragoza, "Net2plan: an open source network planning tool for bridging the gap between academia and industry", *IEEE Network*, vol. 29, no. 5, 2015.