An Automated Service-downgrade Negotiation Scheme for Application-centric Networks

Antonio Marsico, Marco Savi, Domenico Siracusa, Elio Salvadori
FBK CREATE-NET, Via alla Cascata 56D, 38123, Povo, Trento, Italy
{*amarsico, m.savi, dsiracusa, esalvadori*}*@fbk.eu*

Abstract: We propose a novel negotiation scheme for an application-driven relaxation of different requirements in multi-layer networks. Simulative results show that it improves service acceptance while keeping requirements' degradation much lower than applications' worst-case acceptable values.

OCIS codes: (060.4251) Networks, assignment and routing algorithms; (060.4256) Networks, network optimization

1. Introduction

Application-centric (or application-aware) *networking* is an emerging paradigm that aims at catering to multiple requirements (e.g. bandwidth, latency, availability, security, etc.) when serving the traffic generated by business applications (e.g. database synchronization, financial transactions, remote drones control, etc.). Recent studies (e.g. [1,2]) show how application-centric networking can be pursued in multi-layer transport networks by means of joint configuration/optimization of IP/MPLS and optical layers, in order to provide a tailored service throughout the network stack. Specifically, in [2] we showed the advantages of considering a number of application requirements (ARs) in addition to simple bandwidth when applications' service requests (SRs) must be provisioned. We proved that it is possible to achieve service blocking probabilities similar to an application-unaware scheme, while also ensuring that application needs are met. This way, the network can ultimately deliver added-value services to customers at roughly the same cost. However, when facing high network utilizations, the service acceptance ratio experiences a reduction that negatively impacts on the revenues of both network operators, which cannot accommodate new SRs while meeting all the ARs, and customers, which have their service blocked.

A possible solution to overcome this issue is to gracefully downgrade the service in a way that is acceptable for both network operators and applications. The general concept of *service downgrade* has already been tackled in literature: for example, works [3–5] propose some service downgrade schemes as a solution to guarantee reasonable service acceptance ratios even in the face of high network utilization. These works, however, miss two important aspects that are pivotal in application-centric networking: first, they only consider the bandwidth requirement, without any relaxation of other relevant ARs (e.g. latency); second, the service downgrade is unilaterally enforced from the network side with no input or feedback from applications, which may experience unacceptably degraded service.

In this work, we exploit the application-centric vision to extend the notion of service downgrade. We introduce a novel bi-lateral *negotiation scheme* in which, for each SR that cannot be satisfied, the applications are requested to select among a set of alternative solutions proposed by the network. By means of simulations, we show that (i) our negotiation scheme is able to significantly reduce network service blocking probability while leading to predictable (application-endorsed) downgrades in service quality and (ii) jointly relaxing multiple ARs (i.e. the bandwidth and the latency) can lead to counterintuitive effects on how each of these ARs is impacted by the relaxation of the other.

2. Description of the Negotiation Scheme

Our negotiation scheme consists of two distinct algorithmic blocks. An extended version of the Application-Aware Service Provisioning Algorithm [2], including negotiation features, is implemented on the network side as part of a generic Control and Management plane that controls the multi-layer network, e.g. a hierarchical SDN controller. On the application side, an algorithm is designed to autonomously decide among multiple solutions with downgraded service as offered by the network (Alternative Solution Selection Algorithm). Fig. 1 shows the interaction between the application and the Control and Management plane to support negotiation, which can be eased by adopting a well-designed intent-based northbound interface [6]. Next, we describe the algorithms' features.

The Service Provisioning Algorithm proposed in [2] takes as input, for each SR that must be provisioned, (i) the source/destination IP/MPLS nodes to be provided with network connectivity and (ii) the ARs that must be guaranteed at the same time (i.e., minimum bandwidth, maximum latency, minimum availability and optical encryption yes/no). The algorithm's objective is finding an *application-aware path*, i.e., a path meeting all the ARs. To do so, it first tries to find a suitable path at the IP/MPLS layer. If it cannot be found, the algorithm augments the IP topology by reserving optical resources for the establishment of new lightpaths (i.e., IP links).

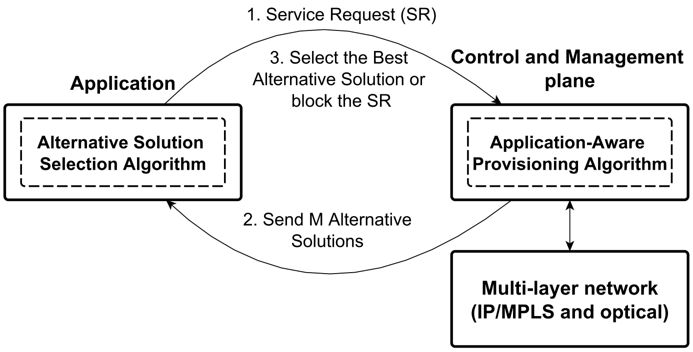


Fig. 1. Negotiation scheme and involved algorithms.

This increases the chance of finding a solution at the expense of resorting to new optical resources. If a path is then found, the SR is provisioned and the resources are allocated, otherwise it is blocked.

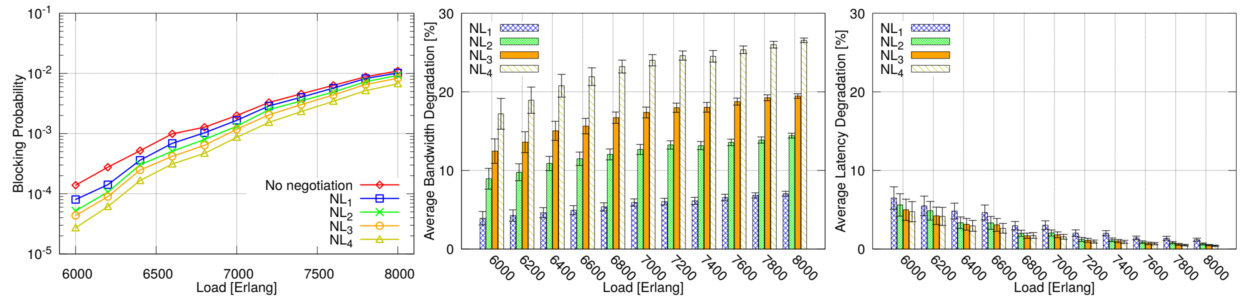
In this paper, we extend the features of such algorithm to support our negotiation scheme. When it is not possible to provision an application-aware path for a SR, a *negotiation phase* is started instead of blocking it. Without any loss of generality, we focus on minimum bandwidth b and maximum latency l ARs, leaving the evaluation of availability and encryption for future work. In our scheme, the application can decide whether to negotiate any AR or not. For this purpose, we represent each SR with a $\overline{SR} = \{s, d, b, l, n_b, n_l\}$ tuple. The tuple carries the information on (i) source s and destination d IP/MPLS node IDs, (ii) minimum bandwidth b and maximum latency l ARs, (iii) flags n_b and n_l, associated with b and l, respectively. Such flags can be set to `true` or `false` to inform whether the associated AR is *negotiable* or not. The Provisioning Algorithm thus computes a set of M Alternative Solutions (ASs), in which the negotiable ARs can have looser and network-achievable values than the ones specified in the SR. Operatively, the ASs are computed by applying a relaxation to all the combinations of negotiable ARs. The tuple $\overline{AS} = \{AS_1, AS_2, \cdots, AS_m\}$ represents all the possible ASs where, in our specific case, $AS_i = \{b_{n,i}, l_{n,i}\}$ includes the values for the considered ARs. For example, if a SR requires b and l with $n_b = n_l =$ `true`, the algorithm offers three different ASs: (i) AS_1 with $b_{n,1} < b$ and $l_{n,1} = l$ (i.e., relax b, not l), (ii) AS_2 with $b_{n,2} = b$ and $l_{n,2} > l$ (i.e., relax l, not b), (iii) AS_3 with $b_{n,3} < b$ and $l_{n,3} > l$ (i.e., relax both b and l).

The set of ASs is then sent to the application for evaluation. On the application side, the Alternative Solution Selection Algorithm analyzes the received ASs to find the best one according to the application needs. In our scheme, every application maintains (i) the preferable value for each AR (i.e., b and l) and (ii) the corresponding least-acceptable values (i.e., b_t and l_t), which are not disclosed to the network and represent the *threshold* values (i.e., minimum for bandwidth and maximum for latency) that the application is willing to accept when negotiation is triggered. For instance, consider a company (e.g. a bank) that requires a seamless virtual machine (VM) migration between two end-points. Standard VM techniques require a maximum latency of $l = 10$ ms. However, the VM can be migrated, with reduced performance, also in case of latencies up to $l_t = 150$ ms. Note that, since the Provisioning Algorithm does not have any knowledge on the threshold values for the ARs, it cannot bias its choice to provide the application with ASs meeting exactly such values. In this way, the application can potentially experience a lower downgrade than the least-acceptable one, if there are enough available resources in the network. Going back to the example, if the Provisioning Algorithm sends an AS with $l_{n,1} = 50$ ms $< l_t$, the VM migration experiences a lower downgrade. Clearly, similar considerations also hold for the bandwidth requirement. The Alternative Solution Selection Algorithm first excludes all the ASs that do not meet either b_t, or l_t, or both. Then, it calculates the normalized Euclidean distance between b and l and the corresponding $b_{n,i}$ and $l_{n,i}$ ARs for each AS_i. The AS associated to the minimum normalized Euclidean distance, defined as *Best AS*, is communicated to the Control and Management plane, the negotiation phase is terminated and resources are allocated. If no AS meets both b_t and l_t, the SR is blocked and the negotiation fails.

3. Performance Evaluation

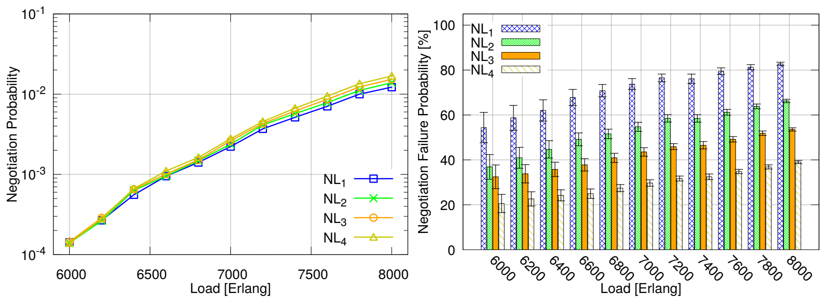
We implemented the negotiation scheme on Net2Plan [7]. We consider a real multi-layer network topology provided by Telefónica Spain (see [2]). The network traffic is distributed according to the non-uniform traffic matrix provided by the same operator. The SRs are generated based on a Poisson process (i.e., with exponential inter-arrival times) and also have exponential holding times. The ARs for each SR are uniformly chosen from the following sets: $b = \{1, 2, 5, 10\}$ Gbps and $l = \{10\}$ ms. We define multiple Negotiation Levels (NLs), in which the threshold values b_t and l_t are set as reported in Fig. 3c. According to the defined NLs, each SR can always tolerate relaxation of both b and l: the maximum bandwidth tolerated degradation is set in terms of bandwidth loss percentage, while the maximum latency tolerated degradation is set in terms of additional delay (in ms). Higher NL subscript is always associated to higher tolerance to service downgrade. For each simulation, we made all SRs belong to the same NL.

Fig. 2 offers an overview on the trade-off between the gain in SR acceptance (in terms of blocking probability reduction) and SR average bandwidth and latency degradation, for the negotiated SRs, as a function of network load and NL. For instance, in the case of a network load of 6000 Erlang, the blocking probability decreases of about an order of magnitude between the *No negotiation* and *NL_4* cases, while bandwidth (Fig. 2b) and latency (Fig. 2c), are only degraded in average by 12% and 7%, respectively (i.e., much less than the maximum tolerated degradation, which is 40% for bandwidth and 300% for latency). The bandwidth degradation increases both with respect to (i) network load and (ii) NL. In fact, with higher loads, the network is only able to offer ASs with in average more degraded band-



(a) (b) (c)

Fig. 2. Evaluation of the blocking probability (a) vs. bandwidth (b) and latency (c) degradation trade-off.



	$(b-b_t)/b\cdot 100$ (%)	l_t (ms)
NL_1	10%	$l+5$
NL_2	20%	$l+10$
NL_3	30%	$l+15$
NL_4	40%	$l+20$

(a) (b) (c)

Fig. 3. Negotiation (a) and negotiation failure (b) probability together with the defined Negotiation Levels (c).

width, since the average network utilization is higher. Moreover, a higher NL makes SRs more tolerant to bandwidth degradations, and the SRs bandwidth is thus in average degraded more. As opposed to bandwidth degradation, latency degradation decreases with respect to NL and network load. The reason is that, in our assumptions, each SR can have b and l degraded at the same time. The higher b degradation is, both as a function of load and NL, the easier finding spare resources on shortest paths is. Higher b degradations are thus always associated to lower l degradations. This behavior points out how multiple ARs experience different degradation trends when they can be relaxed at the same time, and how they mutually influence their trends.

Figs. 3a-3b show the SR negotiation probability (i.e., the probability that the network starts the negotiation phase for a SR) and the negotiation failure probability (i.e., the probability that the negotiation fails because no AS suits the least-acceptable values for the ARs) as a function of network load and NL. Fig. 3a shows that the SR negotiation probability is, as expected, similar to the blocking probability of *No negotiation*: it increases as the network load increases and it is only slightly dependent on NL. Moreover, the negotiation failure probability (Fig. 3b) is higher (i) when the NL is lower and (ii) as the load increases, i.e., in all the cases where network utilization is higher.

4. Conclusion

We proposed a novel bi-lateral service-downgrade negotiation scheme for application-centric networking, which relies on the interaction between the network and the applications to identify solutions that can gracefully relax some requirements. Simulations show that a service blocking probability/downgrade trade-off exists and that our scheme allows applications to experience a much lower service downgrade than their tolerance thresholds. We also showed that a joint relaxation of multiple requirements has a non-intuitive impact on the ultimate quality experienced by downgraded services.

Acknowledgement: 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.

References

[1] V. Lopez *et al.*, "On the benefits of multilayer optimization and application awareness", IEEE/OSA JLT, vol. 35, no. 6, pp. 1274-1279, 2017.
[2] M. Savi *et al.*,"An application-aware multi-layer service provisioning algorithm based on auxiliary graphs", Proc. of OFC, 2017.
[3] H. Y. Chang *et al.*, "A multipath routing algorithm for degraded-bandwidth services under availability constraint in WDM networks", Proc. of IEEE WAINA, 2012.
[4] S. S. Savas *et al.*, "Exploiting degraded-service tolerance to improve performance of telecom networks", Proc. of OFC, 2014.
[5] Z. Zhong *et al.*, "On QoS-assured degraded provisioning in service-differentiated multi-layer elastic optical networks", Proc. of IEEE GLOBE-COM, 2016.
[6] M. Pham *et al.*,"SDN applications - The intent-based northbound interface realisation for extended applications", Proc of IEEE Netsoft, 2016.
[7] P. Pavon-Marino *et al.*,"Net2plan: An open source network planning tool for bridging the gap between academia and industry", IEEE Network, vol. 29, no. 5, pp. 90-96, 2015.