

Carrier SDN for Flexgrid-Based Inter-Datacenter Connectivity

A. Asensio*, L. Gifre, M. Ruiz, and L. Velasco

Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

*e-mail: adrian.asensio@tsc.upc.edu

ABSTRACT

Managing inter-datacenter (DC) connectivity to transport increasing amounts of data and optimize DC performance and costs becomes a challenge not only for DC operators but also for network operators. Current inter-DC connections are configured as static big fat pipes, resulting in large bitrate over-provisioning thus increasing operational costs for DC operators. Furthermore, because of the nature of inter-DC traffic, which highly varies during the time, network operators cannot share such connections with other customers, and network resources may be underutilized. Therefore, an increasing number of studies are being carried out with the aim to address different challenges arising from inter-DC connectivity requirements. This paper summarizes connectivity models and results showing remarkable advantages of considering carrier software defined network (SDN) and elastic operations for managing inter-DC connections.

Keywords: datacenter federation, carrier software defined networks, Flexgrid optical networks.

1. INTRODUCTION

Datacenter (DC) operators deal not only with inherent DC issues such as huge energy consumption but also with extremely dynamic workload mixes and intensities. Thus, when designing DCs, overprovisioning resources for peak workloads might result highly inefficient whereas under-provisioning them might result in poor quality of service (QoS) and service level agreement (SLA) non-fulfillment. In order to optimize resources usage, elastic resources management is performed according to demand. For example, virtualization allows encapsulating jobs (e.g., web applications) in virtual machines (VM) mixed with other workloads and consolidate them in the most proper server according to their performance requirements. Moreover, the local resource manager can migrate VMs among servers for reducing energy consumption while ensuring the committed QoS [1]; in practice, physical machines in DCs are turned off when they are not used and turned on to satisfy increments in demand.

In addition, as proposed in [2], cloud providers can benefit from interconnecting their DCs to create DC federations, not only improving load balancing but also increasing their revenue from using IT resources that in other case would be underutilized and also they can expand their geographical coverage without building new DCs. Under this scenario, DC operations are commonly automated using certain cloud middleware running scheduled-based algorithms for migrating VMs and synchronizing databases (DB) among federated DCs with the aim to optimize a given utility function.

Network operators can provide the necessary network infrastructure for inter-DC connections in DC federations. Connections' set up with the desired bitrate and tear-down can be requested on-demand. In the last years standardization work has been done defining control plane architectures and protocols to automate connection provisioning. Furthermore, the application-based network operations (ABNO) architecture [3] is being defined by the IETF. Based on the ABNO architecture, in [4] we propose a cross-stratum orchestrator and a dynamic elastic connectivity model for managing connections within federated DCs. Additionally, carrier software defined network (SDN) as an abstraction layer to the underlying network on top of ABNO has been studied in [5]. Carrier SDN implements a northbound interface to allow cloud middleware to request transfers using its native semantic, i.e. amount of data to be transferred, DC destination, completion time, etc. Applications' operations are then transformed into network connection requests. The northbound interface uses application-oriented semantic, liberating application developers from understanding and dealing with network specifics and complexity. Finally, from the data plane perspective, the maturity of the Flexgrid technology has enabled elastic optical networks development [6], allowing bandwidth allocation to accommodate spikes [7], [8], so network operators can optimize network resources usage.

This paper summarizes connectivity models for inter-DC connections studied in our previous work in [4] and [5], from current static connections to transferences managed by carrier SDN in an ABNO-based architecture, and present performance results showing the advantages of using carrier SDN and elastic operations in Flexgrid-based optical networks. Detailed descriptions of architectures supporting connectivity models in DC federations can be found in [4] and [5].

2. CONNECTIVITY MODELS

Transport networks are currently configured as big static *fat pipes* overprovisioning the required average capacity to guarantee QoS and traffic demand. The capacity of each inter-DC optical connection is dimensioned in advance according to a foreseen amount of data to transfer. Once in operation, scheduling algorithms inside cloud management run periodically trying to optimize some cost function, such as energy costs, and organize VM migration and DB synchronization according to the bitrate available. If VM migration or DB

synchronization cannot be performed in the desired time, it may eventually lead to performance degradation, thus to ensure transfers finish within the desired time, some over-dimensioning is needed. Clearly, said static connectivity configuration entails high costs since large connectivity capacity remains unused in time periods where low capacity is required. In addition, static connections result inefficient also from the network operator point of view, since they cannot share network resources and optimize their usage. Figure 1a represents a 40 Gb/s optical connection being used only in certain time fractions but occupying network resources during all the time.

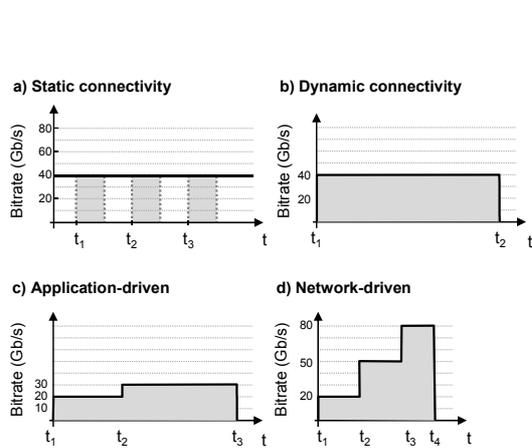


Figure 1. Bitrate vs. time for different connectivity models.

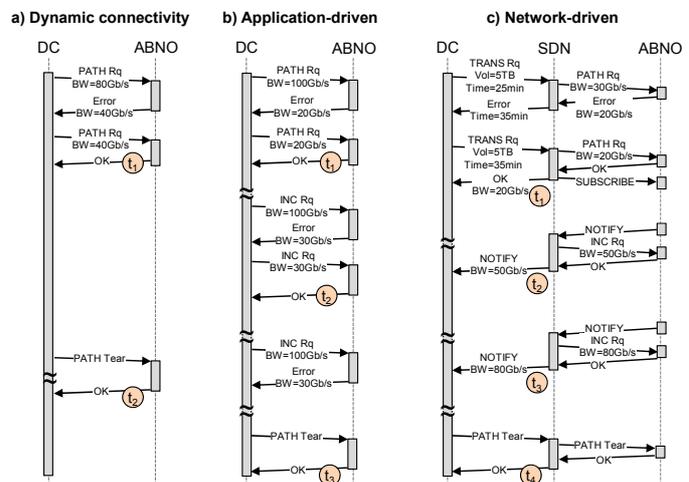


Figure 2. Connectivity models.

In view of the inefficient usage of resources when considering static connectivity, cloud services require new mechanisms to provide reconfiguration and adaptability of the transport network to reduce the amount of overprovisioned bandwidth; the efficient integration of cloud-based services among distributed DCs, including the interconnecting network, then becomes a challenge. Therefore, the cloud-ready transport network was introduced as an architecture to handle this dynamic cloud and network interaction, allowing on-demand connectivity provisioning. Dynamic connectivity allows DC operators to manage optical connections to remote DCs, requesting connections as they are really needed to perform data transfers and releasing them when all data has been transferred. Furthermore, the fine spectral granularity and wide range of bitrates in elastic optical networks make the actual bitrate of the optical connection closely fit connectivity needs. After requesting a connection and negotiating its capacity as a function of the current network resources availability, the resulting bitrate can be used by scheduling algorithms to organize transferences. Figure 1b represents a 40 Gb/s connection established at t_1 and torn down at t_2 ; its capacity is used during the whole time interval.

The sequence in Figure 2a illustrates a simplified version of the corresponding messages exchanged between the proper DC interface and ABNO. Once the algorithms in DC have computed a transfer to be performed, an 80 Gb/s optical connection request to a remote DC is sent to ABNO and operations as described in [4] are performed. Assuming that not enough resources are available for the bitrate requested, an algorithm inside ABNO finds the maximum bitrate and sends a response back to the originating DC with said information. Upon the reception of the maximum available bitrate, 40 Gb/s in the example, the transfer is recomputed, the amount of data to transfer reduced, and finally a new connection with the available bitrate is requested. Transfer is then established (t_1). When the transfer ends (t_2), a message to tear down the connection is sent to ABNO from DC, and the used network resources are released so they can be assigned to any other connection.

Nonetheless, the availability of resources is not guaranteed using dynamic connectivity, and lack of network resources at request time may result in long transference times and even in transference period overlapping. Note that a connection's bitrate cannot be renegotiated and remains constant along the connection's holding time. To reduce the impact of the unavailability of required connectivity resources, in [4] we propose to take advantage of the elasticity that the Flexgrid technology provides, allowing the amount of spectral resources assigned to each connection to be increased/decreased, and thus its bitrate. This adaptation is done if there are not enough resources at request time so that more bandwidth may be requested at any time after the connection has been set up. We refer to this type of connectivity as dynamic elastic or application-driven. It is worth noting that, although applications have full control over the connectivity process, physical network resources are shared with a number of clients and connections' set up and elastic spectrum increments could be blocked as a result of lack of resources in the network. Hence, applications need to implement some sort of periodical retries to increase the allocated bandwidth until reaching the required level.

Figure 1c represents the connection's bitrate during the time according to set up, tear-down and elastic operations shown in Figure 2b. Similarly to dynamic connectivity, the DC proper interface negotiates for a 100 Gb/s connection but ABNO responds to it with maximum connection's capacity available, 20 Gb/s in the example, and after organizing the data to transfer the corresponding connection set up is requested to ABNO and the connection is established (t_1). In the example, some resources are released after the connection has been established, and after a request is received (t_2), they can be assigned to increment the bitrate of the already established connection; the assigned bitrate is then increased to 80 Gb/s, reducing thus the total transfer time. Note that this is beneficial for both DC federation, since better performance could be achieved, and the network operator, since unused resources are immediately occupied. Once the transference finishes, the DC requests to ABNO to tear down the connection (t_3).

Finally, we propose a network-driven model (Figure 1d and Figure 2c) using carrier SDN as described in [5], where applications request transferences instead of connectivity. In this case, the source DC sends a transfer request to carrier SDN specifying the destination DC, the amount of data to be transferred, and the maximum completion time. Upon its reception, the carrier SDN requests ABNO to find the greatest spectrum width available, taking into account local policies and current service level agreements and sends a response back to the source DC with the best completion time. The proper manager in the source DC organizes data transference and sends a new transfer request with the suggested completion time. A new connection is established and its capacity is sent in the response message; in addition, carrier SDN requests ABNO to keep it informed upon more resources are left available in the route of the corresponding label switched path (LSP) supporting that connection. Algorithms deployed in ABNO controller monitor spectrum availability in those physical links. When resource availability allows increasing the allocated bitrate of some LSP, the carrier SDN performs elastic spectrum operations so as to ensure committed transfer completion times. Each time the carrier SDN modifies bitrate by performing elastic spectrum operations, a notification containing the new throughput is sent to the source DC. In the source DC, VM migration is then optimized as a function of the actual throughput while delegating ensuring completion transfer time to the carrier SDN.

3. ILLUSTRATIVE RESULTS

For evaluation purposes, we developed scheduling algorithms in an OpenNebula-based cloud middleware emulator. A worldwide DC federation is considered and its DCs are connected to an ad-hoc event-driven simulator developed in OMNeT++ [9]. The simulator implements the carrier SDN and the Flexgrid network with an ABNO controller on the top. Traffic among DCs (DC2DC) and between users and DCs (U2DC) compete for resources in the physical network. A follow-the-work strategy for VM migration is considered, where VMs are moved closer to the users, reducing thus latency. The DB synchronization policy tries to update the differential images between services running in all the DCs; in the case in which an image cannot be synchronized in time, the next update will attempt to synchronize the whole DB image, increasing the DC2DC traffic overhead. The scheduler runs hourly and the desired transferring time is 30 minutes. Connections of 200 Gb/s and 150 Gb/s are needed for DB synchronization and VM migration, respectively. Further configuration and scenario details can be found in [4] and [5].

Figure 3 shows the used bitrate during time for the dynamic and the application-driven connectivity models under low U2DC traffic; *i.e.* all transferences finish before the next interval starts and thus no overlapping is appreciated. Time to transfer is kept under 1 hour for DBs and VMs in both models. Note that connection's bitrate varies with the amount of data to transfer in contrast to the constant 200 Gb/s bitrate used in the static model and resulting in bitrate savings as high as 60% [4].

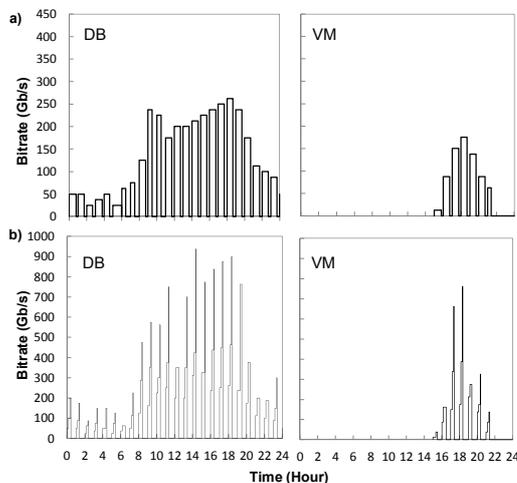


Figure 3. Dynamic (a) and application-driven (b) connectivity.

Table 1. Time-to-transfer (minutes).

	Static	Dynamic	App-driven
DB (Max/Avg)	54.0/28.5	58.0/39.9	49.0/24.7
VM (Max/Avg)	50.0/28.7	48.0/39.9	40.0/24.4

Table 2. Application and network -driven comparison.

	App-driven	Net-driven
DB (Max/Avg)	58.0/29.0	28.0/22.4
VM (Max/Avg)	54.0/28.2	27.0/22.2
Requests (#/h, %success)	43.1, 53.5%	65.3, 100%

Table 1 shows maximum and average values of transference times. Values obtained with the application-driven model are clearly lower than those obtained with the dynamic and even the static connectivity models. Furthermore, dynamic connectivity exceeds the desired average transference time of 30 minutes while it is kept near 25 minutes when considering the application-driven model. It is worth noting that, as shown in [4], in scenarios with higher U2DC traffic, the amount of needed bitrate for the dynamic connections might not be available at the time of the request, so this model is not able to perform data transfers within 1 hour, leading to period overlapping and poor performance as some DBs become degraded and users perceive an increased latency. The dynamic elastic model is able to keep time to transfer under 1 hour for both low and high U2DC traffic thanks to elastic bitrate operations performed on the established optical connections.

Notwithstanding, the cost of elasticity in the application-driven model is in the control plane, since the amount of messages that need to be processed is slightly increased; 40.8 request messages/h to increase connections' capacity in addition to 36.8 messages/h for connections' setup and teardown. More than half of requests, 53.8%, were successful, and the optical connection was expanded.

Finally, in order to compare the application-driven and the network-driven models we use carrier SDN as described in [5]. Table 2 shows that when using the network-driven model both the maximum and average required time-to-transfer are significantly lower than when the software-driven model is used. The longest transfers could be done in only 28 minutes when the network-driven model was used compared to just under 60 minutes using the software-driven model. Note that the amount of requested bitrate is the same for both models. Table 2 also shows the number of required requests messages per hour needed to increase bitrate of connections for the whole scenario. Only 53% of those requests succeeded to increase connections' bitrate under the application-driven model, in contrast to 100% reached under the network-driven model.

4. CONCLUSIONS

Inter-DC connectivity models have been presented based on our previous work in [4] and [5]. Aiming to improve the performance of dynamic connectivity we propose application-driven and network-driven connectivity models, both taking advantage of elastic connections supported by Flexgrid-based interconnection networks. The application-driven model needs periodical retries requesting to increase connection's bitrate, which do not translate into immediate bitrate increments and could have a negative impact on the performance of the inter-DC control plane. In contrast, the network-driven model takes advantage of using carrier SDN allowing cloud middleware to request transfers using its native semantic, *i.e.* amount of data to be transferred, DC destination, completion time, *etc.* Applications' operations are transformed into network connection requests.

Illustrative results show that application-driven connectivity improves the performance of the dynamic one in scenarios where the physical optical network is shared by several services, so competence to use network resources arises. Both dynamic and application-driven models show bitrate savings near 60% with respect to static connectivity. In addition, when using the network-driven model transferring time is remarkably reduced in comparison with the application-driven model.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement n° 317999 IDEALIST project. Moreover, it was supported by the Spanish science ministry through the TEC2011-27310 ELASTIC project.

REFERENCES

- [1] M. Mishra *et al.*, "Dynamic resource management using virtual machine migrations," *IEEE Commun. Mag.*, vol. 50, 2012, pp. 34-40.
- [2] L. Velasco *et al.*, "Elastic operations in federated datacenters for performance and cost optimization," *Elsevier Computer Communications*, 2014.
- [3] D. King, A. Farrel, "A PCE-based Architecture for Application-based Network Operations," IETF draft, Oct. 2013.
- [4] L. Velasco *et al.*, "Cross-stratum orchestration and Flexgrid optical networks for datacenter federations," *IEEE Network Magazine*, vol. 27, pp. 23-30, 2013.
- [5] L. Velasco *et al.*, "Towards a carrier SDN: An example for elastic inter-datacenter connectivity," (Invited Paper) *OSA Optics Express*, vol. 22, pp. 55-61, 2014.
- [6] M. Jinno *et al.*, "Spectrum-efficient and scalable elastic optical path network: Architecture, benefits, and enabling technologies," *IEEE Communications Magazine*, vol. 47, pp. 66-73.
- [7] M. Klinkowski *et al.*, "Elastic spectrum allocation for time-varying traffic in FlexGrid optical networks," *IEEE JSAC*, vol. 31, 2013, pp. 26-38.
- [8] A. Asensio *et al.*, "Impact of aggregation level on the performance of dynamic lightpath adaptation under time-varying traffic," in *Proc. IEEE International Conference on Optical Network Design and Modeling (ONDM)*, 2013.
- [9] A. Asensio *et al.*, "An elastic networks OMNeT++-based simulator," in *Proc. International Conference on Transparent Optical Networks (ICTON)*, 2013.