Analytical Model for Anycast Service Provisioning in Data Center Interconnections

Abstract—This paper presents an analytical model to evaluate the performance of an optical wavelength-routed network for inter- and intra-data center interconnection, designed to work according to the cloud computing paradigm. Such a scenario calls for anycast routing of service requests to find a network path that best suits both connectivity and IT resource requirements. The analytical model is based on conventional queuing theory for loss systems combined with an ad-hoc combinatorial analysis of the anycast service alternatives. The model presented here is meant to provide engineering guidelines for data center interconnection networks supporting the cloud paradigm, in particular in terms of performance and cost of resource distribution.

I. INTRODUCTION

The cloud paradigm is gaining popularity as a platform to offer a large set of computing services (e.g., network storage, computation power, server time) to the general public in an on-demand fashion according to the utility delivery model, i.e., “pay as you use”. Cloud computing customers may be offered a subscription-based access to either infrastructures, platforms or applications, referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) or Software as a Service (SaaS), respectively [1]. The cloud service delivery model promises significant Capex and Opex reductions to enterprises, and can generate new market opportunities for network operators and service providers [2]. On the other hand, it is driving new architectural directions for data center infrastructures, i.e., from a static set of equipments vertically dedicated to different applications, towards geographically distributed pools of shared resources that can be dynamically combined and orchestrated as business policy dictates to deliver many different applications (e.g., content storage, high performance computing) [3], [4].

In this context, fast reconfigurable communication services capabilities within or among data centers are required to address new issues and requirements for data center network infrastructures. In fact, the scaling up of computational capabilities located in data centers or the emergence of new masses of users, may require the dynamic reconfigurability of machine-to-machine data paths for addressing stricter bandwidth requirements or different levels of QoS assurance according to IT requirements. Definitely, dynamic allocation of network resources is required among data centers that is adaptive with respect to the IT resource dynamics [5].

Specifically, the IaaS delivery model requires the capability to access the IT resources which best match the user requests, without a-priori specification of the “target” destination, e.g., the hosting server etc., according to the anycast principle. A further benefit is offered by the choice of the best combination of IT and connectivity resources, which must be selected in accordance with the request’s QoS requirements. The anycast approach aims at minimizing the effects of possible architecture bottlenecks, e.g., due to bandwidth unavailability or network device reconfiguration time, thanks to its native capability to balance the load among the network and IT resources available. In particular, we have already shown that the router reconfiguration time plays an important role in determining the performance of an agile network capable of self-adapting to the application requests [6], and that its negative effect may be limited when IT resources are available and may be accessed at different network locations [7].

The anycast service provisioning drives new architectural directions for optical networking and poses new network design challenges. Therefore, new performance evaluation tools are needed, with the goal to provide engineering guidelines and means of comparison among design alternatives. In this paper we propose an analytical model to study the performance of an optical wavelength-routed data center interconnection network serving customer requests according to the aforementioned principle. The model is based on conventional queuing theory for loss systems combined with an ad-hoc combinatorial analysis of the anycast service alternatives.

The paper is organized as follows. Section II defines the network model assumed in our study. Section III presents the combinatorial model of the anycast service, which is a prerequisite for the markovian model of service blocking described in section IV. Then a simple cost-performance trade-off is discussed in section V, with the help of some numerical results. Finally, section VI concludes the work.

II. NETWORK MODEL

The general network model assumed here consists of an optical interconnection network within the context of either inter-data center or intra-data center communication between users and resources. In this work a “user” is any entity that requests a given IT “resource”, assuming that equivalent instances of the resource are present within one data center...
or across more data centers, according to the anycast service model. For instance, a user could be an IaaS customer who needs to access some specific computing or storage resources operated by Virtual Machines (VMs) in cloud providers’ data centers; or it could represent a service manager that needs to migrate one or more VMs to different racks in the same data center, or from one data center to another.

Users and data center resources are assumed to be located at the network edges. Specifically, in the inter-data center scenario the IT resource instances are located across a set of data centers served by edge nodes generally different from those serving users, e.g., customer premises, service management system. In this case, the interconnection network between IT resources and users spread out along either metropolitan or geographical area. In the intra-data center scenario the interconnection network is the local inter-rack network, and the user and the resource instances are located in different racks. In both scenarios, the interconnection network must provide the user with a suitable edge-to-edge network path towards the requested resource.

The optical interconnection network is made by a mesh of lightpaths established between edge nodes. Lightpaths are established according to a long-term network planning strategy (e.g., by means of well-known routing and wavelength assignment techniques [8]). However, the provisioning of the network, i.e. a higher network infrastructure cost is required. This also means more lightpaths need to be established on the network, i.e. a higher network infrastructure cost is required. Such a cost/performance trade-off should be evaluated with the help of our model together with the definition of a suitable cost function, as will be discussed in section V.

In order to make our model more tractable, let us consider the following assumptions:

A.1 each user request needs the whole lightpath bandwidth $B^u$ (bandwidth segmentation approaches will be considered for an extended model in future works);

A.2 each location has the capacity of hosting the same number $k$ of resource instances (not necessarily instances of the same specific resource);

A.3 the $m$ instances of the resource requested by a given user are uniformly distributed over the $n$ locations (considering the general case when multiple instances of the same resource could be available at the same location).

The service request blocking model is obtained in two steps. First, we derive a combinatorial model of the anycast service that is able to capture the random distribution of resource instances over the potential locations (section III). Then, we define a Markov chain that describes the current state of the lightpaths and allows to compute the service request blocking probability (section IV).

III. COMBINATORIAL MODEL OF THE ANYCAST SERVICE

The anycast service model assumed in this paper raises a combinatorial problem that we must solve in order to evaluate the service request blocking probability. As a matter of fact, a given request from user $u$ is blocked whenever the locations of resource instances $C^u_1, C^u_2, \ldots, C^u_n$ are all unreachable due to the lack of lightpath bandwidth. Let $h$ be the number of lightpaths currently not available, i.e. with the whole bandwidth $B^u$ occupied, according to assumption A.1. The occurrence of the request blocking situation can be evaluated by counting how many placements of the $m$ resource instances on the $n$ locations are possible such that all of them end up to be located at the endpoints of some or all the $h$ lightpaths currently not available.

Let us start with counting how many placements of the $m$ resource instances are possible in general, recalling that, according to assumption A.2, each of the $n$ locations is capable of hosting $k$ resource instances. This is given by the number of combinations of $m$ indistinguishable objects chosen out of $nk$, i.e. by the binomial coefficient [9]

$$\binom{nk}{m} = \frac{(nk)!}{m! (nk - m)!}$$

(1)

According to assumption A.3, the $m$ instances are uniformly distributed over the $nk$ places, so each of the possible placements in (1) has the same probability of occurrence, given
be the set of all partitions of an integer \( m \). For instance, the five existing circles indicate the places where the resource instances are located.

Individually, since they represent three distinct placements of the boxes, given that:
1. each box is made by \( k \) placements of the compartments;
2. each compartment can hold up to one object;
3. places and locations are ordered.

Figure 1 are indistinguishable because we assumed that all the resource instances can be placed in \( m \) instances in exactly \( m \) ways.

To proceed with the count, we must find all the possible placements of the \( m \) requested resource instances: \{I\_1, I\_2, I\_3\}, \{I\_1, I\_2, I\_3\}, \{I\_1, I\_2, I\_3\}, \{I\_1, I\_2, I\_3\}, \{I\_1, I\_2, I\_3\}.

To proceed with the count, we must find all the possible placements of the number \( m \) into \( s \) positive terms. Let \( P_{m,s} \) be the set of all partitions of an integer \( m \) into \( s \) positive terms.

The generic partition \( \varphi \in P_{m,s} \) can be represented in two ways:
1. as a sum of \( s \) terms \( c^\varphi_i > 0 \), \( i = 1, 2, \ldots, s \), such that \( c^\varphi_1 + c^\varphi_2 + \ldots + c^\varphi_s = m \);
2. as a sequence of \( m \) coefficients \( a^\varphi_j \geq 0 \), \( j = 1, 2, \ldots, m \), such that \( a^\varphi_1 + 2a^\varphi_2 + 3a^\varphi_3 + \ldots + ma^\varphi_m = m \) and \( a^\varphi_1 + a^\varphi_2 + \ldots + a^\varphi_m = s \), where \( a^\varphi_m \) counts the number of terms \( c^\varphi_i \) in partition \( \varphi \) that are equal to \( j \).

Each partition \( \varphi \in P_{m,s} \) represents a subset of possible placements of the \( m \) instances in the \( s \) locations. For example, if \( m = 4 \) and \( s = k = 3 \), partition \( 4 = 2 + 1 + 1 \) represents any placement with two instances in one location and one instance in each of the other two locations, including (among others) the three cases displayed in Fig. 1. To count how many placements are represented by a given partition \( \varphi \), let us first consider the case where the box compartments are indistinguishable. In this case, we need to count how many arrangements of the \( c^\varphi_i \) terms are there such that \( a^\varphi_1 \) terms are equal to 1, \( a^\varphi_2 \) terms are equal to 2 and so on. In other words, how many ways \( s \) objects can be divided in \( a^\varphi_1 \) objects of class 1, \( a^\varphi_2 \) of class 2 and so on. This number is given by the multinomial coefficient [9]

\[
\binom{s}{a^\varphi_1, a^\varphi_2, \ldots, a^\varphi_m} = \frac{s!}{a^\varphi_1!a^\varphi_2! \ldots a^\varphi_m!}
\]  

Considering now the compartments distinguishable, any arrangement counted by (3) is such that there are \( \binom{k}{c^\varphi_i} \) ways of placing the resource instances in the first location, \( \binom{k}{c^\varphi_i} \) in the second and so on. Therefore, the number of possible placements represented by a given partition \( \varphi \in P_{m,s} \) is

\[
N_\varphi = \binom{a^\varphi_1, a^\varphi_2, \ldots, a^\varphi_m}{s} \prod_{i=1}^{s} \binom{k}{c^\varphi_i}
\]  

In our example with \( m = 4 \) and \( s = k = 3 \), partition \( \varphi : 4 = 2 + 1 + 1 \) is such that \( c^\varphi_1 = 2, c^\varphi_2 = c^\varphi_3 = 1, a^\varphi_1 = 2, a^\varphi_2 = 1 \) and \( a^\varphi_3 = 0 \). Then the possible placements related to this partition are

\[
N_\varphi = \binom{3}{2, 1, 0} \binom{3}{1} \binom{3}{1} = 3^4 = 81
\]

In fact, the multinomial coefficient counts the three ways in which the “2” and “1” terms can be arranged, i.e. \( \{2, 1, 1\} \) and \( \{1, 1, 2\} \), whereas each binomial coefficient counts the three ways in which one or two objects can be placed in the three compartments of each box.

By summing \( N_\varphi \), for all the partitions \( \varphi \in P_{m,s} \), we obtain the total number of possible placements of \( m \) requested resource instances in exactly \( s \) locations, given that each location hosts \( k \) places. This number must be multiplied by the \( \binom{m}{h} \) ways it is possible to choose \( h \) locations out of the \( m \) currently unreachable due to bandwidth unavailability.

Finally, this count must be repeated for each possible value of \( s \) that causes a blocking situation. These cases are limited by the following upper and lower bounds: \( s \leq \min\{m,h\} \), because the block occurs only when the \( m \) instances are placed in no more than the \( h \) unreachable locations; \( s \geq \lceil m/k \rceil \), because this is the minimum number of different locations in which \( m \) resource instances can be placed if each location can hold at most \( k \) of them. The result is the total number of possible placements \( M_{m,h} \) such that the \( m \) instances of resource \( C^m \) are located in some or all the \( h \) locations currently unreachable:

\[
M_{m,h} = \sum_{s=\lceil m/k \rceil}^{\min\{m,h\}} \binom{h}{s} \sum_{\varphi \in P_{m,s}} N_\varphi
\]
Since each placement of \( m \) instances has probability \( q_c \), the probability of a placement resulting in a blocking situation is given by \( q_c M_{m,h} \).

**IV. MARKOVIAN MODEL OF SERVICE BLOCKING**

According to our network model, user service requests are blocked when other users consume all the bandwidth on lightpaths reaching all the possible resource locations. The service time is therefore defined as the period an admitted user keeps both a resource instance and the lightpath to the corresponding location busy. In order to compute the service request blocking probability, we need to determine which states of the network are such that all the resource instances requested by a generic user are placed in unreachable locations. To this purpose, we can represent the network state as the number of locations currently unavailable due to lack of bandwidth, i.e., \( h = 0, 1, \ldots, n \). Assuming that the statistical behavior of user requests follows a Poisson process and that the service time is exponentially distributed, a Markov chain describing the network state evolution can be defined using the following parameters:

- \( \tau \): average inter-arrival time between consecutive requests in the same user location (i.e., such that the arrivals see the same set of lightpaths);
- \( \lambda = 1/\tau \): request arrival rate;
- \( \vartheta \): average service time;
- \( \mu = 1/\vartheta \): service rate;
- \( A_0 = \lambda/\mu \): load offered to the system.

The network is therefore modeled as a loss system where the number of servers is equal to the number of potential resource locations \( n \) and service request blocking events may occur in different system states depending on the current resource location. In particular, the number of possible placements of the \( m \) requested instances such that these are all located in the \( h \) currently unreachable locations is given by (5). This leads to the following blocking probability when the system is in state \( h \):

\[
P_{m,h} = \sum_{s=[m/k]}^{\min\{m,h\}} \left( \frac{h}{s} \right) \sum_{\phi \in \mathcal{P}_{m,s}} \left( a_1^\phi a_2^\phi \cdots a_m^\phi \right) \prod_{i=1}^{\lfloor k/s \rfloor} \left( c_i^\phi \right)^{nk/m} \tag{6}
\]

The Markov chain state diagram is shown in Fig. 2. According to assumption A.1, on request arrival the transition from generic state \( h \) to state \( h+1 \) occurs when at least one of the \( m \) requested resource instances is placed in a location reachable through one of the \( n-h \) available lightpaths. In state \( h = 0 \), all locations are reachable and any incoming request can be accepted, making the system move to state \( h = 1 \) with rate \( \lambda_0 = \lambda \). In state \( h = 1 \), the request is blocked when all the \( m \) instances are located in the only location currently unreachable.

\[
P_{m,1} = (1 - p_{m,1}) \lambda \tag{7}
\]

The blocking rate for the \( h \)-th state: the transition to state \( h+1 \) occurs with rate \( \lambda_h = (1 - p_{m,h}) \lambda \). This time the blocking in state \( h \) can only happen when \( m \leq k \), otherwise it is impossible to place \( m \) instances in no more than the \( h \) locations not reachable. In state \( h = n \) all the locations are unreachable and \( p_{m,n} = 1 \). Transitions from state \( h \) to state \( h-1 \) occur with rate \( h \mu \), since \( h \) locations are currently unreachable and each of them becomes reachable again with rate \( \mu \).

By solving the Markov chain in Fig. 2 we obtain the following steady-state probabilities:

\[
P_0 = \left( 1 + A_0 + \sum_{h=2}^{n-1} \prod_{r=1}^{h-1} \left( 1 - p_{m,r} \right) \frac{A_h}{h!} \right)^{-1}
\]

\[
P_1 = P_0 A_0 \tag{7}
\]

\[
P_h = P_0 \prod_{r=1}^{h-1} \left( 1 - p_{m,r} \right) \frac{A_h}{h!} \quad 2 \leq h \leq n
\]

where \( p_{m,h} = 0 \) when \( 0 \leq h < \lfloor m/k \rfloor \).

Finally, the total blocking probability can be obtained by adding the contributions from each state, resulting in

\[
P_L = \sum_{h=1}^{n} p_{m,h} P_h \tag{8}
\]

**V. NUMERICAL RESULTS**

This section provides some numerical results obtained by applying the Markov model defined above. The main performance metric considered here is the service request blocking rate \( P_L \) as computed in (8). First, we investigate the role of the model parameters and discuss how different network design choices affect the performance. Then, we define a simple cost function and evaluate the cost/performance trade-off offered by the network model described in section II.

Figure 3 shows the service request blocking rate as a function of the offered load for different numbers of resource locations and, consequently, different numbers of lightpaths. The number of locations/lightpaths is clearly a very influential parameter, as it measures the degree of distribution of the requested resource instances. Obviously, the blocking rate
increases with the offered load, but distributing the $m = 3$ resource instances among a higher number of locations can significantly reduce the blocking rate.

Figure 3 includes results obtained with the model (lines) and compares them with the corresponding values obtained through simulation (points). An ad-hoc, discrete-event simulator of the network described in section II has been developed to validate the proposed Markov model. All simulations were performed by generating Poisson arrivals with variable input traffic load. The service times were generated according to two distributions, namely deterministic and exponential, with the same mean value, leading to the same blocking rate. This confirms that the loss system we are studying is affected only by the average value of the service time. Simulations were run long enough ($10^7$ request arrivals were generated for each simulation point) to measure the results with a very good confidence. The comparison shows a perfect match between model and simulation results, which successfully validates the Markov model.

Figure 4 shows again the service request blocking rate as a function of the offered load for different numbers of locations/lightpaths. This time different parameters have been chosen and $P_L$ is plotted on a logarithmic scale, to better show the differences under light load. The advantage of distributing the resource instances among multiple locations is confirmed, with differences of a few orders of magnitude between the case of $n = 2$ and $n = 16$ locations. This figure shows how the proposed model can be used as a design tool to determine the amount of potential resource locations that must be enforced to obtain a given performance level under a given traffic load.

Figure 5 shows the service request blocking rate as a function of the offered load for different numbers of requested resource instances. The curves span over a large range of values under light traffic conditions, whereas they tend to converge when $A_0$ increases. This behavior implies that increasing the $m$ resource instances available in the network, without increasing the total number $n$ of potential locations, has a significant impact only below a given traffic value. Therefore, the model can be used to determine the traffic threshold above which simply increasing $m$ is not sufficient anymore, but additional lightpaths towards new potential locations must be established.

Figure 6 shows the service request blocking rate as a function of the offered load for different numbers of resource instances hosted by each location. The curves are very close to each other, demonstrating that increasing only the number $k$ of places available in each location to host the resources does not help in reducing the blocking situations, which are mainly determined by how the $m$ resource instances are distributed among the $n$ locations.

Figure 7 shows the service request blocking rate as a function of the offered load assuming different arrangements of the $nk$ potential resource places over $n$ locations. The goal here is to obtain a design tool to determine, for a given number of potential resource places and a given traffic load, how much these places should be distributed over multiple locations in order to obtain a given performance. Of course, the more distributed the places, the lower the blocking rate, with the best
choice represented by the extreme case of a single resource placed in each location \((n = 32, k = 1)\) in Fig. 7.

However, a price must be paid to improve the blocking performance by increasing the number of locations \(n\), since distributing resources over more locations means additional lightpaths must be established, causing an increase in the network infrastructure cost. Also, adding more resource places to an existing location has its cost, since additional storage space and processing power is needed, although this cost is reasonably less relevant due to the economy of scale.

To have a general idea of the cost/performance trade-off offered by our network model, we define the following simple normalized infrastructure cost function:

\[
I(n, k, z) = nz + k
\]  

where \(z\) is the ratio between the cost of adding a new location/lightpath and the cost of adding a new resource place:

\[
z = \frac{\text{cost of increasing } n \text{ by one unit}}{\text{cost of increasing } k \text{ by one unit}}
\]

So, \(I(n, k, z)\) gives the cost of interconnecting the user location to the resources available in the data center, normalized to the cost of adding a single resource place. Two examples of the infrastructure costs resulting from the choices considered in Fig. 7 are given in Table I, namely for \(z = 10\) and \(z = 100\). The table offers a simple, generic but useful tool to find the most suitable choice of \(n\) and \(k\) to achieve a reasonable trade-off between infrastructure cost and performance.

**VI. Conclusion**

In this paper we presented an analytical model for service provisioning over optical data center interconnection networks, according to the anycast principle. The model is based on conventional queuing theory for loss systems with an additional combinatorial analysis specific to the anycast service. Results show the possible trade-off between the advantages of a wider distribution of the requested resource instances, in terms of reduced service blocking rate, and the consequent infrastructure cost. The model can be considered as a useful design tool for data centers supporting the cloud computing paradigm, which provides a quantitative methodology to compute how many resource instances must be distributed and how in order to satisfy some given performance and cost requirements.

**References**