Efficient Routing and Bandwidth Assignment for Inter-Data-Center Live Virtual-Machine Migrations

Omran Ayoub, Francesco Musumeci, Massimo Tornatore and Achille Pattavina
Politecnico di Milano, Department of Electronics, Information and Bioengineering, Milan, Italy
Email: {*firstname.lastname*}@polimi.it

***Abstract*—Using virtualization, service providers can create an abstraction of the physical servers hosted in their data centers, and run their services directly on these abstract entities, called Virtual Machines (VMs). As a direct results of service virtualization, services hosted on VMs can be migrated from a data center to another. VM migration can be performed without or with a minimal service interruption and in this case we talk about live VM migration. The main issue of live VM migration is that memory pages are modified during the process. Due to this fact, same memory pages might be transferred several times, thus increasing both the migration duration and the amount of network resources occupied by the VM migration process. Considering the increasing amount of VM migrations carried on in inter data center networks, efficient strategies for live-VM-migrations bandwidth provisioning are needed. In this paper, we propose and compare various novel Routing and Bandwidth Assignment (RBA) algorithm for the live migration of VMs in a distributed DC infrastructure, under dynamic traffic conditions. We find that assigning a bandwidth to the VM migration request in proportion to the path length with the objective of minimizing the amount of resources occupied by the migration request could improve the network performance. In addition, we present a comparative analysis of the serial and the parallel migration of multiple VMs. We quantify the impact of the migration strategy (serial versus parallel) on blocking probability, migration duration, downtime and resource occupation.**

***Index Terms*—Virtualization, virtual machines, live migration, routing and bandwidth assignment.**

I. Introduction

Cloud computing has been undergoing an exponential growth as the cloud infrastructures are increasingly used to host enterprise and public-Internet services. This growth has been made possible also by the introduction of virtualization, which enables the sharing of computing resources across diverse locations, and it allows Data Center (DC) operators to efficiently exploit their infrastructure. The cloud infrastructure is typically composed of a number of geographically-distributed DCs connected through a high-capacity optical transport Wide Area Network (WAN). To ease operation and management, these cloud infrastructures are built over physical servers (based on general-purpose hardware), which are virtualized using Virtual Machines (VMs) such that each VM supports a given cloud service. For several purposes, such as load balancing [1], [2], fault tolerance and resiliency [3], redundancy, improved quality of experience, energy conservation [4], these interconnected DCs often perform migration of VMs [5]. As a result, the underlying optical network needs to support a rapidly-increasing traffic demand, in terms of VM migrations.

Although VM migration has been widely investigated for intra-DC scenarios (i.e., where the endpoint physical servers are hosted within the same DC), in this study we consider an inter-DC scenario as VMs migration across a distributed DC infrastructure enables further benefits.

VM migration requires transferring all VM data, i.e., disk, memory and processors states, from a source to a destination server. A possible baseline approach for VM migration consists in halting the VM at the original server and, after the data transfer associated with the VM migration is completed, the VM can be re-activated at the destination server location. This approach is known as off-line VM migration and it is characterized by service interruption as the VM can not be accessed while the migration process is taking place. For this reason, DC operators and service providers are resorting to the new concept of *live VM migration* [6], where VMs are not halted during the data transfer process, thus significantly reducing service downtime [1] [7], [8]. As live VM migration increases the network resource occupation, additional amount of data needs to be transferred to keep the VM memory synchronized between the source and destination DC. This extra amount of data transferred is due to the data modification that occurs while the migration is taking place, as end-users can still access the service and the VM's memory. In addition to the migration of a single VM, in some scenarios, multiple cooperating VMs need to be migrated together to provide certain services. The main two approaches proposed for multiple VMs migrations are the parallel and the serial migration techniques (more details are provided in Sec. II). In parallel VM migration, cooperating VMs are migrated simultaneously in an on-line manner, whereas, in the serial VM migration technique, the VMs are migrated sequentially following a mixed on-line and off-line VM migration techniques [9].

Since live VM migration requires a significant amount of bandwidth reserved between DCs, we should study new bandwidth assignment techniques avoiding waste of resource [10] [11]. Moreover, since memory pages are modified during the VM migration, a non-linear relationship between the bandwidth assigned to migrate a VM and the migration duration arises, which might significantly penalize the migration duration specially when low bandwidths are assigned to migrations. However, there are situations in which a lower bandwidth assignment is preferred if it saves on network resources. Hence, the bandwidth assignment of live VM migration is not

[1]The time period during which the VM is paused.

trivial.

The focus of this paper is to propose efficient routing and bandwidth assignment (RBA) algorithms for single and multiple inter-DC VM migration in a dynamic scenario. In addition, in this paper we perform a comparative study between the performance of parallel and the serial multiple VM migration techniques.

A. Related Works

Several studies have addressed the RBA problem in optical networks with the aim to improve the network utilization but without considering VM migration (see, e.g. [12] and [13]). Other studies have formulated the RBA problem for VMs migration with duration and downtime constraints in photonic cloud networks such as Ref. [10] but without taking the network state into consideration. Moreover, some studies have focused on VM migration but mainly to improve the performance of the migration process by minimizing the downtime or the VM migration duration. Refs. [14], [15], and [16] proposed quantitative models for the migration duration and downtime of VM migration over a WAN whereas Ref. [17] presented a theoretical analysis of the necessary bandwidth to satisfy the total migration time and the downtime constraints of a single VM migration. In addition, the use of compression techniques was also proposed to reduce the amount of data transferred during the VM migration [18] [19]. As for multiple VM migration techniques, Ref. [20] proposed improved mixed serial and parallel migration strategies as to optimized the migration process but not yet considering the state of the network. Nevertheless, some studies proposed strategies for an appropriate assignment of live VM migration bandwidth showing that changing the provisioned bandwidth during the VM migration could reduce the resource consumption [11]. Live VM migration has been also investigated with regards to energy expenditures [21] and bandwidth utilization [22], as well as for security issues as in [23] but no specific work has focused on the routing and bandwidth assignment of live VMs with respect to minimizing the overall network resource consumption.

B. Paper Contribution

In our previous work [24], we investigated the impact of a proper Routing and Bandwidth Assignment (RBA) for live VM migration on the underlying transport network performance by proposing three algorithms RBA algorithms for live VM migration. The results of one of the algorithms showed how a proper VM migration process could be performed while taking the network resources into consideration. In this paper, we extend the work in [24] by proposing a new enhanced algorithm that utilizes the network resources in a more efficient way. Moreover, through the use of the new enhanced algorithm, we perform multiple parallel and serial VMs migration and present a comparative analysis between the two approaches.

The rest of the paper is organized as follows: In Sec. II we present background information on the techniques adopted for single and multiple VMs migration. In Sec. III we introduce the proposed RBA algorithms for single VM migration whereas in Sec. IV for multiple VMs migration. Sec. V shows illustrative numerical results used to evaluate the performance of proposed algorithms. Finally, Sec. VI concludes the paper.

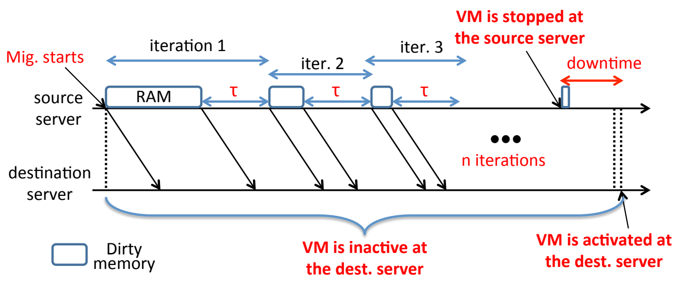


Fig. 1. Example of the iterative-copy phase during a VM migration.

II. Background on VMs Migration

In this section we present the background concepts for off-line and on-line single VM migration and we explain how the parallel and the serial VM migration work.

A. Single VM Migration

As previously mentioned, a single VM migration could be performed either off-line or on-line. Off-line migration is performed by transferring a VM from the source to the destination physical server while the VM is paused (halted) at the source server. In this case, the VM, and thus the service provided by it, could only be accessed after migration is completed. In this case, a high service outage is introduced which might be intolerable in some cases [25]. More precisely, given a VM with V^M bits to be transferred (including all disk, memory and processors states), if migration is performed using B bit/s of bandwidth, the VM migration duration will be approximately $T_{mig} = V^M/B$, and this will roughly coincide with the service downtime in the off-line migration strategy. As an example, considering a VM size of 10 Gbit [8] and a migration bandwidth of 100 Mbit/s, service downtime $=$ approximately 2 minutes, which may be intolerable from a user-perspective.

On the contrary, live migration approach performs data transfer while the VM is still running [9]. This approach is characterized by an iterative data transfer process [8], that stops only after a given threshold is reached, in order to achieve low service downtime [26]. In this approach the VM memory is "*dirtied*" (i.e., modified) during the migration due to the users activity, therefore additional information (the *dirtied* data) needs to be transferred together with the original VM state to make sure the VM at the destination server is synchronized with the one at the source server.

The parameter capturing this aspect is called *memory dirtying rate* D, the rate at which the VM memory is modified during the migration process. D is usually expressed in the number of dirtied pages per second (pages/s.), but in our paper we refer to it in bit/s. It might vary based on the type of VM, its hosted applications, as well as the number of served users and their activity.

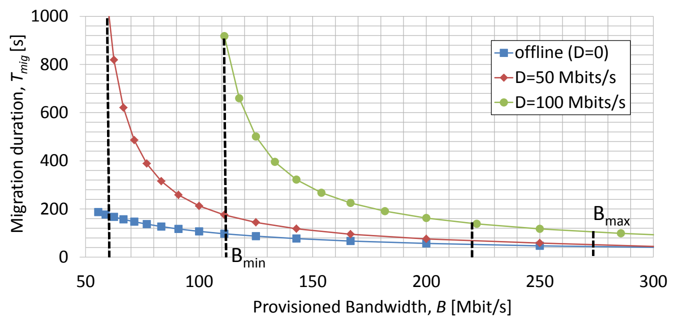


Fig. 2. Function points curves for a VM of size $V^M = 10$ Gbit and variable dirtying rates: $D = 0$ (offline), $D = 50$ Mbit/s and $D = 100$ Mbit/s.

Figure 1 shows how live migration works. The first iteration is used to transfer the original VM memory to the destination, while following iterations are used to transfer only the "dirtied" memory. Thus, the duration of each iteration depends on the amount of memory dirtied during the previous iteration. An inter-iteration delay τ is also shown in the figure, which is due to end-to-end network delay, processing delay at either end, or a combination of both. The iterative copy phase stops when a specific stop condition is met. The stop condition could be a predefined number of iterations or the amount of dirtied memory low enough to meet with a targeted maximum downtime. During the final stop-and-copy phase, the VM is stopped at its source, remaining dirtied memory is copied, and the network is re-configured before bringing the VM up at the destination location. The duration of the stop-and-copy phase is the period when the VM is inactive [27]. Thus, to achieve low downtime, the duration of iteration i must be lower than that of iteration $i-1$, therefore the bandwidth used to perform migration must be greater than the VM dirtying rate D[2]. Note that the total migration duration T_{mig} not only depends on the provisioned bandwidth, B, but also on the dirtying rate D.

The dependency between T_{mig} and B is represented in the function-points curves shown in Fig. 2, the 3 curves correspond for the migration of a VM with size $V^M = 10$ Gbit and for different dirtying rate values, i.e., $D = 0$ (off-line migration), $D = 50$ Mbit/s and $D = 100$ Mbit/s. We used the model in [27] to obtain these curves, which in the following will be referenced to as function points curve. First of all, we highlight the effect of the dirtying rate on the migration process. As an example, if provisioning bandwidth B is set to 110 Mbit/s, the off-line migration would require around 100 seconds, whereas for $D = 50$ Mbit/s T_{mig} increases to about 200 seconds, and for $D = 100$ Mbit/s it becomes 900 seconds. This means that the overall amount of transferred data is doubled for a dirtying rate of 50 Mbit/s and could be as high as 10x for a high dirtying rate (e.g. $D = 100$ Mbit/s or higher). Moreover, we highlight the effect of the bandwidth set for the migration process. Considering the migration of the VM of Fig. 2 at a $D = 50$ Mbit/s, if the provisioning bandwidth B is set to 60 Mbit/s, T_{mig} = 800 seconds while if B is doubled (set to 120 Mbit/s), T_{mig} decreases to about 120 seconds. If B is doubled again (set to 240 Mbit/s), T_{mig} decreases to around 90 seconds. On one hand we note that, as the provisioned bandwidth increases, the advantage in terms of total migration duration is reduced, so higher resources occupation (intended as the product $T_{mig} \cdot B$) can be experienced if increasing the bandwidth too much. On the other hand, reducing B produces a drastic increase in the total migration duration, due to the high number of iterations needed for the migration. For all these reasons, the choice of the bandwidth to be associated to the VM migration is not trivial.

[2]We assume that D is constant for the entire duration of a VM migration. This can be interpreted as a worst-case scenario, where D is the maximum possible value for that specific VM.

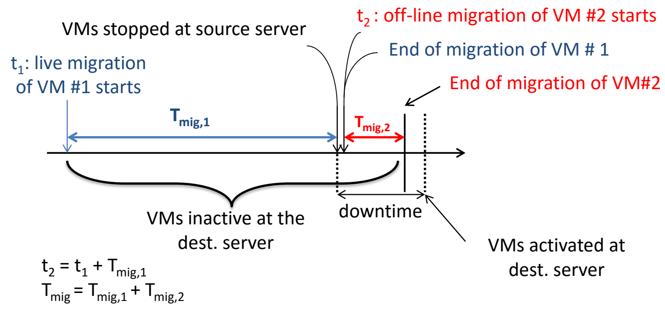
TABLE I
B_{max} VALUES FOR DIFFERENT VMS TYPES

Dirtying rate	V^M	B_{max}
100 Mbit/s	100 MB	286 Mbit/s
100 Mbit/s	1 GB	400 Mbit/s
100 Mbit/s	10 GB	667 Mbit/s
500 Mbit/s	100 MB	700 Mbit/s
500 Mbit/s	1 GB	2000 Mbit/s
500 Mbit/s	10 GB	2500 Mbit/s

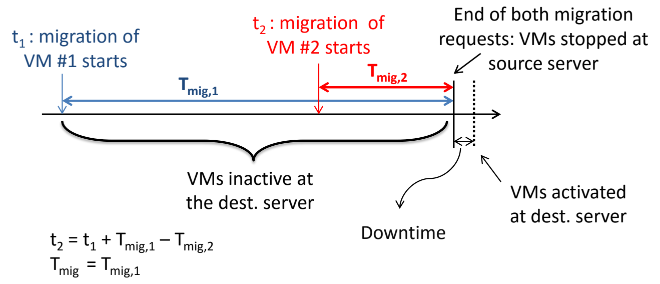
As stated before, the lower bound for the provisioned bandwidth is the VM dirtying rate, D. In other words, the VM migration process could not converge in case the provisioned bandwidth is less than the dirtying rate D. As for the upper bound, we set a limit on the maximum bandwidth we can allocate namely when $\frac{\Delta T_{mig}}{\Delta B} \leq 0.1\%$, i.e., when the advantage in terms of migration duration is very low. These lower and upper bounds are qualitatively shown in Fig. 2 for the case of $D = 100$ Mbit/s, and are denoted as B_{min} and B_{max}, respectively. In the following, we use a fixed setting for B_{min}, according to the VM dirtying rate, i.e., $B_{min} = 1.2D$, whereas the values of B_{max} are shown in Tab. I for the considered VM types. The allocated bandwidth will fall between the upper and lower bounds and such allocation is preformed according to the specific RBA algorithm considered.

B. Cooperating VMs

In the case of a group of cooperating VMs, all VMs in the group must be active simultaneously to provide the service. The two main approaches for the migrations of a group of cooperating VMs are the serial VMs and the parallel VMs migration [9]. In the serial VM migration, VMs are migrated one at a time. In this case, both on-line and off-line VM migration is performed as shown in Fig. 3(a). First, the VM with the largest memory size (VM#1) is migrated on-line at t_1. After the end of VM#1 migration, the service is stopped and an off-line migration of VM#2 starts at t_2. At the end of the migration of the last VM, the VMs are reconfigured at the destination server and the service becomes available again. A main disadvantage of this approach is the high downtime introduced by the off-line VM migration. As for parallel VM migration, all VMs are migrated simultaneously as shown in Fig. 3(b). First, the VM with the largest memory size (VM#1) is migrated at t_1 whereas VM#2 is migrated at t_2 in such a way that both migrations are scheduled to be completed at the same instant. Parallel VM migration reduces downtime but has a longer migration duration as different migrations have to share the capacity available. In our paper, we further investigate how to reduce the network resource occupation


t$_2$: off-line migration of VM #2 starts
VMs stopped at source server
t$_1$: live migration of VM #1 starts
End of migration of VM # 1
End of migration of VM#2
T$_{mig,1}$
T$_{mig,2}$
VMs inactive at the dest. server
downtime
VMs activated at dest. server
t$_2$ = t$_1$ + T$_{mig,1}$
T$_{mig}$ = T$_{mig,1}$ + T$_{mig,2}$

(a) Serial Migration


t$_1$: migration of VM #1 starts
t$_2$: migration of VM #2 starts
End of both migration requests: VMs stopped at source server
T$_{mig,1}$
T$_{mig,2}$
VMs inactive at the dest. server
Downtime
VMs activated at dest. server
t$_2$ = t$_1$ + T$_{mig,1}$ – T$_{mig,2}$
T$_{mig}$ = T$_{mig,1}$

(b) Parallel Migration

Fig. 3. An example of the serial and the parallel multiple VMs migration process in case of 2 VMs.

of serial and parallel migration by using the proposed RBA algorithms.

III. Routing and Bandwidth Assignment for Single VM Migration

In this section, we describe the four proposed dynamic RBA algorithms proposed for live VM migrations in optical transport networks, namely *MIN*, *MAX*, *Distance Adaptive* (*DA*), *Enhanced Distance Adaptive* (*E-DA*).

Given a physical network topology, where optical fiber links interconnect DC locations via generic nodes equipped with WAN IP routers, the dynamically arriving VMs migration requests are provisioned by performing RBA according to one of the four proposed strategies. One VM migration request r is modeled by a t-uple (s_r, d_r, V_r^M, D_r), where s_r and d_r are the source and destination DC for the VM migration request, V_r^M is the VM size (in bytes), whereas D_r is the VM dirtying rate, in Mbit/s[3]. For each link we guarantee that the overall bandwidth provisioned for the migration requests does not exceed the link capacity. No wavelength continuity is considered as we assume traffic undergoes opto-electrical conversion at each node.

In Fig. 4 we show the flow-chart of the VM migration request provisioning/deprovisioning process. Upon the arrival of a VM migration request $r : (s_r, d_r, V_r^M, D_r)$ at time instant t, the K-shortest paths between s_r and d_r are calculated. The cost metric for the K-shortest paths algorithm is the number of hops. The K paths are then sorted in increasing order of number of hops and inserted in a list. Starting from the first

[3]Here we do not refer to specific types of VMs, but we identify them only with their size and dirtying rate, so their corresponding function points curves.

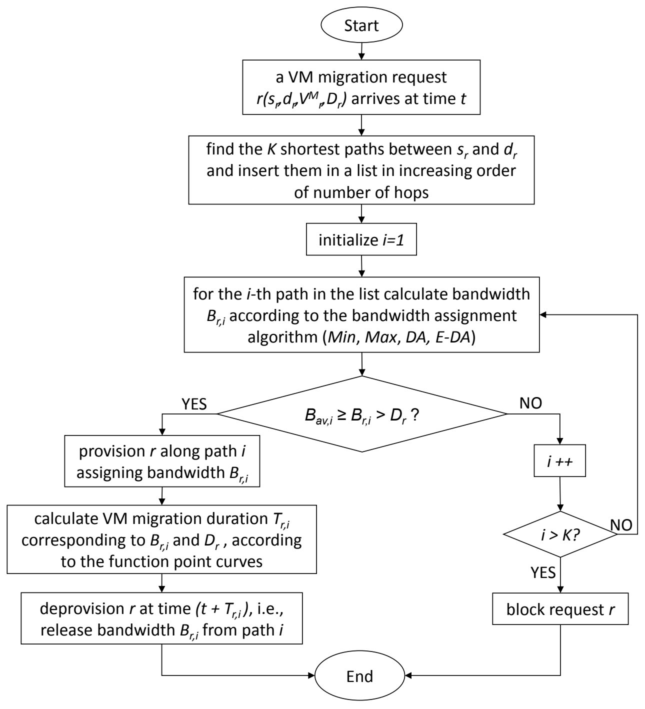

Start
a VM migration request r(s$_r$,d$_r$,V^M_r,D$_r$) arrives at time t
find the K shortest paths between s$_r$ and d$_r$ and insert them in a list in increasing order of number of hops
initialize i=1
for the i-th path in the list calculate bandwidth B$_{r,i}$ according to the bandwidth assignment algorithm (Min, Max, DA, E-DA)
B$_{av,i}$ ≥ B$_{r,i}$ > D$_r$?
YES
NO
provision r along path i assigning bandwidth B$_{r,i}$
calculate VM migration duration T$_{r,i}$ corresponding to B$_{r,i}$ and D$_r$, according to the function point curves
deprovision r at time (t + T$_{r,i}$), i.e., release bandwidth B$_{r,i}$ from path i
i ++
i > K?
NO
YES
block request r
End

Fig. 4. Flow chart for the provisioning/deprovisioning of a migration request.

(i.e., shortest) path in the list, say path i, the corresponding bandwidth to be allocated for request r, $B_{r,i}$, is calculated, according to the chosen RBA algorithm (*MIN*, *MAX* or *DA*). If such bandwidth is available along path i and it is greater than the dirtying rate (i.e., if $B_{av,i} \geq B_{r,i} > D_r$, where $B_{av,i}$ is the available bandwidth along path i), then the VM migration request is provisioned along path i using bandwidth $B_{r,i}$ and its duration, $T_{r,i}$, is calculated according to $B_{r,i}$ and D_r using the function points curves as described in Sec. II. Finally, the migration request is deprovisioned at time $t+T_{r,i}$, deallocating bandwidth $B_{r,i}$ from path i. If no path is found with enough available bandwidth, the VM migration request is blocked.

A. Minimum Bandwidth Assignment Algorithm (MIN)

Algorithm MIN assigns the minimum amount of bandwidth to the migration request, with the intent of avoiding links congestion. As observed in Sec. II, for the iterative on-line migration approach to converge, the provisioned bandwidth must be greater than the dirtying rate D. Therefore, in the MIN approach, for a generic path i the bandwidth assigned to a VM migration r is calculated as a fixed (small) increase with respect to the D, i.e., $B_{r,i} = D_r(1 + \alpha)$. The choice of proper α value is not trivial and has been obtained through a sensitivity study where we evaluated the performance of *MIN* algorithm for different values of α. The best performance has been obtained for $\alpha = 0.2$.

B. Maximum Bandwidth Assignment Algorithm (MAX)

Contrary to MIN, the *MAX* RBA algorithm aims at assigning the maximum bandwidth possible to VM migrations requests. By doing so, a higher capacity is occupied for a shorter period (i.e., migrations duration is low) with the intent of performing the VM migration in a short duration.

TABLE II
AN EXAMPLE OF THE BANDWIDTH ASSIGNMENT BY THE DA ALGORITHM ALONG WITH THE AMOUNT OF RO OF THE POSSIBLE BANDWIDTH ASSIGNMENT VALUES FOR A VM OF SIZE = 100 MB AT A $D = 100$ $Mbit/s$.

Path #	H	B (Mbit/s)	T_{mig} (sec)	RO $[B \cdot T_{mig} \cdot H]$
1	4	233	11.03	10279.96
2	5	208	12.21	12698.4
3	6	184	14.11	15577.44

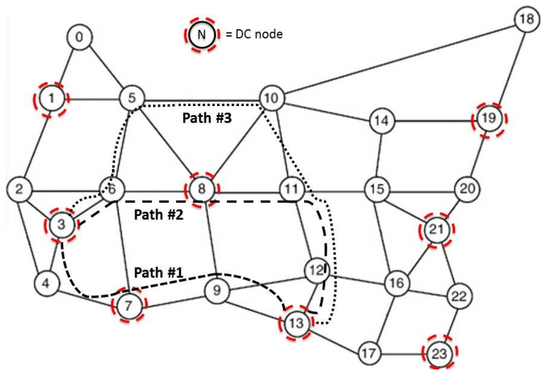


Fig. 5. USA24 Network Topology showing the 3 paths from node 3 to node 13. DCs nodes are highlighted.

For a certain path i selected to accommodate VM migration request r, the provisioned bandwidth is calculated as $B_{r,i} = min(B_{av,i}, B_{max,r})$, where $B_{max,r}$ is the maximum possible bandwidth provided by its function points curve, as shown in Tab. I.

C. Distance Adaptive: DA

The intuition behind this algorithm was based on the results revealed by the previously described algorithms, MIN and MAX. On one hand, the results showed that under low traffic conditions (i.e., when short paths are available), provisioning a VM migration request with a high bandwidth is more suitable for both the performance of the migration process, as the migration could be completed in a small duration and for the network resources as high network resources would be occupied but for a short amount of time. On the other hand, under high traffic conditions (i.e., when only long paths are possible due to the fact that short paths are already occupied) it is better to provision VM migration requests with a smaller bandwidth as to avoid network capacity shortage and thus avoid more blocking.

Thus, the idea behind the DA algorithm is to assign bandwidth to a VM migration request according to the length of the selected path. Specifically, for path i, having h_i hops, we assign to request r, a bandwidth $B_{r,i}$ given by:

$$B_{r,i} = B_{max,r} - (h_i - H_{min}) \frac{B_{max,r} - B_{min,r}}{H_{max} - H_{min}} \quad (1)$$

H_{min} is the number of hops of the shortest path between two any DC locations whereas H_{max} is the network diameter. $B_{min,r}$ and $B_{max,r}$ are the minimum and maximum bandwidth values corresponding to request r (i.e., to its dirtying rate D_r), as provided by function points curves. Thus, according to Eqn. 1, a VM migration request of a path length H_{min} will be provisioned by the maximum bandwidth value whereas the minimum bandwidth value will be assigned to a path of length H_{max}. An example of 3 possible paths that could be assigned to a VM migration request from node 3 to node 13 by DA is shown in Fig. 5 and the values of the bandwidth assigned and respective RO are tabulated in Tab. II. Note that the RO is calculated as the product $B \cdot T_{mig} \cdot H$. Note that the bandwidth assigned to shorter paths is always greater than the bandwidth assigned to longer paths. In this case, the DA algorithm chooses the shortest path with its respective bandwidth. A drawback of this algorithm is that lower or higher amounts of bandwidths than that suggested by DA could save on network resources.

Algorithm 1 *E-DA* Algorithm

```
- Find 4 shortest paths;
- provision = $false$;
- Initialize $i = 0$;
while ($i < 4$ && $provision == false$) do
    - Calculate bandwidth $B$ according to path length $h$ using Eqn. 1;
    - Calculate corresponding $T_{mig,}$ using function points curves;
    - Evaluate resource occupation, $RO = B \cdot T_{mig} \cdot h$;
    - Find the bandwidth values which precedes ($B^-$) and follows ($B^+$) $B$ in the function points curve
    - For each of these values of bandwidth, calculate corresponding migration duration and then resource occupation, $RO^- = B^- \cdot T^-_{mig} \cdot h$ and $RO^+ = B^+ \cdot T^+_{mig} \cdot h$;
    - Select provisioning bandwidth $B$, $B^-$ or $B^+$ which minimizes resource occupation.
    if (provisioning bandwidth<bandwidth available) then
        - provision = $true$;
    else
        - $Increment$ $i$;
End;
```

TABLE III
AN EXAMPLE OF THE BANDWIDTH ASSIGNMENT BY THE *E-DA* ALGORITHM ALONG WITH THE AMOUNT OF RO FOR EACH POSSIBLE PATH.

Path #	H	B (Mbit/s)	T_{mig} (sec)	RO $[B \cdot T_{mig} \cdot H]$
1a	4	250	10.73	10730
1b	4	233	11.03	10279.96
1c	4	200	12.39	9912
2a	5	233	11.03	12849.95
2b	5	208	12.21	12698.4
2c	5	200	12.39	12390
3a	6	200	12.21	14868
3b	6	184	14.29	15776.16
3c	6	167	16.79	16823.58

D. *Enhanced Distance Adaptive* E-DA

The main idea behind of *E-DA* is to capture the advantages of DA such as setting high migration bandwidth values for short paths and assigning lower bandwidth values for long paths, and yet reduce the overall resource occupation. When performing RBA, *E-DA* starts considering the path

Algorithm 2 Serial and Parallel Migration Techniques

```
- Initialize i=1;
- Sort VMs in descending order of memory size;
- Provision VM #1 with E-DA;
- Calculate $T_{mig_1}$, using function points curves;
- Set $t$ = current time;
if Migration technique is parallel then
  while i < number of VMs do
    - Provision VM #i with E-DA;
    - Calculate $T_{mig_i}$;
    - Set migration at time $t_i = t + T_{mig_1} - T_{mig_i}$;
    - Increment i;
if Migration technique is serial then
  while i < number of VMs do
    - Assign $B_{max}$ to migrate VM #i off-line;
    - Set migration at time $t_i = t_{i-1} + T_{mig_{i-1}}$;
    - Increment i;
End;
```

and bandwidth value which would be provisioned with the DA algorithm and then computes the resulting RO. Then, considering the function points curve corresponding to the migration dirtying rate, other values of provisioned bandwidth are investigated, namely the previous and following bandwidth values in the function points curve, and for each value the corresponding T_{mig} and RO are computed. The actual provisioned bandwidth will then be the one providing the lowest RO. Algorithm 1 shows the technique followed by the *E-DA*. As an example, for each of the paths listed in Tab. II, *E-DA* creates two new possible solutions with different bandwidth values. For each value of bandwidth proposed by DA, *E-DA* checks if the nearest lower and the nearest higher values according to the function points curve could yield to a lower RO. The possible solutions found by *E-DA* are tabulated in Tab. III. We can see that assigning a bandwidth value of 200 $Mbit/s$ for the migration request would save on RO.

IV. RBA Algorithms for Serial and Parallel Migrations for Multiple VMs

In this section we show how the serial and parallel approaches for multiple VM migration are performed.

A. Parallel VMs migrations

In the parallel VM migration, all the VMs in a group are migrated simultaneously. The service downtime experienced in this approach is the interval of time that separates the end of the first VM migration and the last one. For this reason, we schedule the migration of all VMs in such a way that they are completed at the same instant. Note that in the case when the VMs have the same memory size the best solution would be to assign the VMs migrations the same bandwidth and have them start at the same time, thus they will be completed at the same instant and the service downtime will be minimal[4]. Instead, if the VMs have different memory sizes, the VMs are sorted by their size and the first VM (largest memory size) is migrated first and other VMs are then scheduled in such a way that the instant of the completion of the migration request coincides with that of the first VM such as to minimize the downtime. Note that the on-line migration of the requests is performed following the *E-DA* algorithm such as to minimize the overall network RO.

[4]Note that we consider the dirtying rate to be at a fixed rate during the same simulation.

As an example, assume there are two VMs to be migrated of sizes VM_1 = 10 GB and VM_2 = 1 GB, respectively, at a dirtying rate of $DR = 100$ Mbit/s. If the bandwidth set for the migration of VM_1 is 667 Mbit/s, its migration duration $T_{mig1} = 28.64$ seconds, whereas for VM_2 the bandwidth is 286 Mbit/s and its migration duration $T_{mig2} = 16.87$ seconds. The best solution to minimize the downtime of the service offered by these two correlated VMs would be to migrate the second virtual machine after the start of the first VM migration by 11.77 seconds *(=28.64-16.87)*. It is also important to point out that in rare cases the bandwidth required to have a perfect alignment of the migration requests might not be available on the shortest path, and thus smaller capacity will be assigned to the request which increases the migration duration and thus the service downtime increases. Algorithm 2 depicts the technique followed for both the parallel and the serial VM migrations.

B. Serial VMs migrations

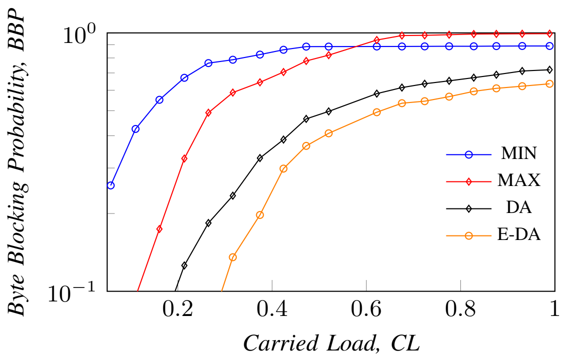
The migrations of VMs in the same cooperating group are performed in a serial manner. A common way of performing the serial VMs migrations process is to migrate the VM with the largest size memory using live VM migration but then migrate the other cooperating VMs using off-line approach. Note that the service could not be provided after the first VM has fully migrated, and thus, an off-line VM migration is performed for the remaining VMs. The VMs that are migrated off-line are assigned the maximum bandwidth possible from the function-points curve (Fig. 2). This technique introduces larger service downtime with respect to the parallel technique, but it may significantly decreases the overall network RO.

Referring to the same example used for the parallel case, the two VMs VM_1 and VM_2 will be now migrated according to the serial approach. VM_1 will be migrated in the same manner. After the migration of VM_1 is completed, VM_2 will be migrated off-line and assigned a maximum migration bandwidth $B = 800$ Mbit/s yielding a migration duration $T_{mig2} = 10$ seconds.

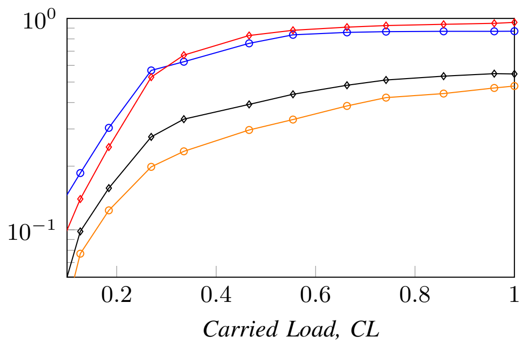
V. Simulative Numerical Results

A. Simulation Settings and Performance Metrics

To compare the performance of the 4 RBA algorithms, we developed a discrete event-driven C++ simulator. The simulations were run on a workstation equipped with 4 x 3.1 GHz processors and with 8 GB of RAM. The topology considered in this study is the USA-24 network Fig. 5, constituted by $N = 24$ nodes (8 of them serve as DC locations) and $E = 43$ bidirectional links, each with $C = 100$ Gbit/s capacity in both directions. We simulate the arrival of 300.000 VM migration requests, with Poisson-distributed arrival rates λ. For each VM migration, source and destination DCs are uniformly selected



(a) $BBP\ vs\ CL\ (D = 100Mbit/s)$



(b) $BBP\ vs\ CL\ (D = 500Mbit/s)$

Fig. 6. BBP values for increasing carried load, (CL), for the four RBA algorithms MIN, MAX, DA and *E-DA*. (a) BBP at $D = 100$ Mbit/s; (b) BBP at $D = 500$ Mbit/s.

TABLE IV
PERFORMANCE COMPARISON BETWEEN *MIN*, *MAX*, *DA* AND *E-DA* ALGORITHMS.

$D\ (Mbit/s)$	*Algorithm*	$B_{avg}\ (Mbit/s)$	T_{mig} *(sec.)*	$ADT\ (Mbit)$	H	$RO\ (Mbit \cdot H)$
100	MIN	120	766.5	79980.9	3.30701	308297
	MAX	450	17.7008	9199	3.45046	31216.4
	DA	408	21.7974	10030	2.89701	30192
	E-DA	415	20.0771	8339.5	2.80017	26849.2
500	MIN	600	486.6	232010	3.30701	521119
	MAX	2054	10.9427	23497.6	3.9609	79732
	DA	1861	13.705	25895.9	2.9892	74315.7
	E-DA	1905	11.4561	22238.87	2.91092	67490

among the 8 locations. Simulations are performed for different values of λ, ranging from 1 to 200 arrivals per second. We considered three different types of VMs, differing in their memory size and dirtying rate, as specified in Tab. I. The function points curves used to calculate the VM migration duration are based on formulas in [27], and their behavior is similar to that described in Fig. 2. For all the RBA algorithms, we set the number of shortest paths to be considered in the provisioning step to $K = 4$.

The performance of the algorithms are evaluated considering the following metrics:

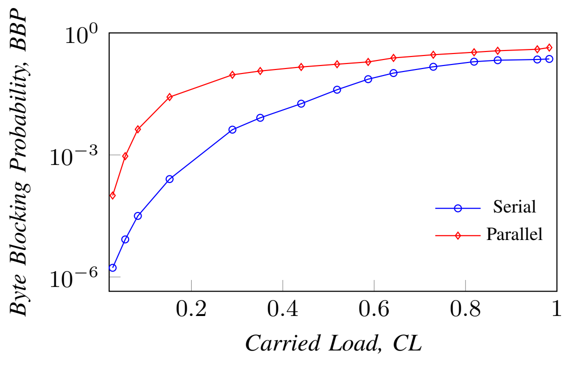
- *Byte Blocking Probability* (BBP): Calculated as the ratio between unprovisioned bytes (i.e., amount of bytes for VM migration requests which are not provisioned) and total offered bytes (i.e., the total amount of data of arriving VM migration requests)[5].
- *Average Migration Time* (T_{avg}): The average migration duration of all request.
- *Average Bandwidth Assigned* (B_{avg}): It is the average bandwidth assigned for all requests.
- *Average Data Transferred* (ADT): It is the average amount of data transferred for a VM migration request.
- *Average downtime*: It is the average duration the service provided by the VM is down.
- *Resource Occupation (RO)*: it measures the amount of network resources utilized to accommodate all the provisioned VM migration requests expressed in $bit \cdot hop$ (or, similarly, $byte \cdot hop$).

[5]Note that for each VM migration request, the offered data is different from the data actually transferred, as it depends on the associated bandwidth.

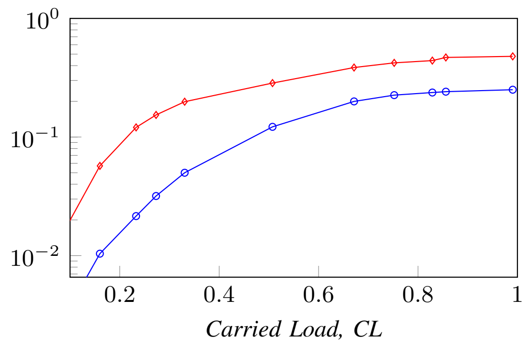
B. Discussion

1) Single VM Migrations: In this analysis we first compare the performance of the four proposed algorithms, MIN, MAX, DA and *E-DA* in terms of the BBP for increasing carried load, *CL*. This represents the average level of network resource occupancy and is defined as the ratio between allocated resources for provisioned requests and the total resources of the network, i.e., $CL = \frac{RO \cdot \lambda}{E \cdot C}$. Moreover, we compare the two algorithms in terms of B_{avg}, T_{mig} and ADT.

Figure 6 shows the numerical results comparing the proposed algorithms for different values of the dirtying rate $D = 100$ Mbit/s and $D = 500$ Mbit/s. As shown in Fig. 6(a) and 6(b), MAX performs better than MIN at lower *CL* where MIN performs better than MAX at a higher *CL*. This is due to the fact that, when the network is low-loaded, shorter paths can be usually exploited to accommodate VM migration requests, especially in the MAX case, where network bandwidth provisioned to migrate VMs can be released soon, due to shorter migration duration values. On the other hand, for increasing CL, longer routes are often needed, so the MIN algorithm shows its benefits since lower network bandwidth are allocated throughout the network for each VM migration request. The DA algorithm, provisions requests with bandwidth proportional to the path length (Eqn. 1), thus catching the potentials of both MIN and MAX algorithms and has a lower BBP than both of MIN and MAX. This is



(a) $BBP\ vs\ CL\ (D = 100 Mbit/s)$



(b) $BBP\ vs\ CL\ (D = 500 Mbit/s)$

Fig. 7. BBP values for increasing carried load, (CL), for the serial and the parallel VM migration techniques. (a) BBP at $D = 100$ Mbit/s; (b) BBP at $D = 500$ Mbit/s using the *E-DA* algorithm

TABLE V
PERFORMANCE COMPARISON BETWEEN THE SERIAL AND PARALLEL VM MIGRATION TECHNIQUES.

$D\ (Mbit/s)$	$Technique$	$T_{mig}\ (sec.)$	$Downtime\ (sec.)$	$ADT\ (Mbit)$	$RO\ (Mbit \cdot H)$
100	Serial	27.13	4.79	14289	30152
	Parallel	17.47	0.052	17830	44575
500	Serial	19.67	4.79	27938	44803
	Parallel	14.89	0.031	32713	58869

due to the fact that DA exhibits a behavior similar to MAX for shorter paths while a behavior similar to MIN for longer paths, thus taking advantage of MIN and MAX. This way, DA avoids the main drawbacks of both algorithms which are the long T_{mig} in case of MIN and inefficient use of resources in case of MAX. Moreover, the *E-DA* performs better than DA for all values of the *CL*. This is due to the fact that it routes requests with the goal of savings on network resources, which in turn allows the network to provision requests on shorter routes.

In order to compare other metrics of the proposed algorithms, new simulations were performed using a CL that allowed to maintain a ***BBP*** lower than 1%. Tab. IV compares the 4 algorithms in terms of B_{avg}, T_{mig}, ADT, H and RO at fixed BBP for two values of D. For both values of D, MIN and MAX have the lowest and the highest values of average provisioned bandwidth B_{avg}, respectively. As for ***E-DA*** and DA, the value of B_{avg} when applying ***E-DA*** is slightly higher than that of DA, as ***E-DA*** saves on network resources which allows it to use shorter paths than those used by DA and thus it uses higher values of bandwidth. In turn, we observe that ***E-DA*** has lower T_{mig} than DA while MIN and MAX exhibit the longest and the shortest T_{mig}, respectively. As for ADT, at both values of the considered dirtying rate, the ADT for *E-DA* is lower than that of the other algorithms. This shows that the total data transferred for the VM migrations could be minimized by balancing between provisioned bandwidth and needed migration duration. As previously mentioned, as a result of saving on network resources, it is confirmed that *E-DA* was capable of finding shorter routes than the other algorithms as H for *E-DA* is lower than that of MIN, MAX and DA. Moreover, the average RO of *E-DA* is the lowest among the algorithms proposed.

2) Multiple VMs Migrations: In Fig. 7 we show the comparison between the serial and parallel VM migration techniques in terms of the BBP for increasing carried load, *CL* and for different values of the dirtying rate, D. For both values of D, the serial VM migration technique outperforms the parallel technique at a lower CL, whereas the gap between the two strategies reduces for increasing CL. This is mainly due to the fact that with the serial approach live migration is only adopted to migrate the first VM, while other VMs are migrated in an off-line manner. Therefore, less network resources are occupied in comparison to the parallel approach.

In order to compare more in detail both techniques, we show numerical results obtained for CL values providing ***BBP*** lower than 1%. Tab. V shows the comparison of parallel and serial VM migration techniques in terms of T_{mig}, $downtime$, ADT and RO for values of D equal to 100 and 500 $Mbit/s$. We see that T_{mig} in the *serial* case is greater than that for the *parallel* one. Referring to the table, we also show the service $downtime$. For both values of D, the downtime is negligible in the *parallel* case, as we force the migration of cooperating VMs to be completed at the same instant, while it is relevant (i.e., it is equal to $4.79s$) in the *serial* scenario. We can point out that, due to the fact that an off-line VM migration is adopted and the same amount of bandwidth is set to perform the off-line migration for both values of D, the downtime in the *serial* case is the same, independently of the value of D. Due to the same reason, ADT for the *serial* technique is notably lower than that of the *parallel* technique. Similarly, the RO of the *serial* technique is lower than that of the *parallel* technique, thus confirming that it might be preferred to apply a serial migration technique to save on network resources especially if the tolerated downtime is in the order of few seconds.

From Tabs. IV and V, we note that T_{mig} for $D = 500$ is lower than that for $D = 100$ for both cases of multiple VM migration and single VM migration. This is because T_{mig} is dominated by the first iteration which transfers the whole memory of the VM, and since the migration bandwidth values at $D = 500$ are higher than those at $D = 100$, T_{mig} at $D = 500$ is lower.

VI. Conclusion

In this paper we focused on the RBA problem for live VMs migrations in an inter-data center networks. We first provided four Routing and Bandwidth Assignment (RBA) algorithms for live VM migration under dynamic traffic. We investigated the dependency between the bandwidth for VMs migration requests, VM dirtying rate and migration duration. We showed that an efficient network utilization (and, in turn, lower BBP) could be obtained when bandwidth provisioning is performed taking into account path length, bandwidth assigned and migration duration. Counter intuitively, the results showed that there are situations where a lower bandwidth assignment could be preferred. Moreover, we provided a comparative analysis between serial and parallel multiple VMs migration techniques. We showed that the serial VMs migration technique could save on network resources since an off-line VM migration technique is adopted but introduces a high service downtime.

References

[1] P. Padala, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, and K. Salem, "Adaptive Control of Virtualized Resources in Utility Computing Environments," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 3, pp. 289–302, 2007.

[2] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic Placement of Virtual Machines for Managing SLA Violations," in *2007 10th IFIP/IEEE International Symposium on Integrated Network Management*, pp. 119–128.

[3] B. Cully, G. Lefebvre, D. Meyer, M. Feeley, N. Hutchinson, and A. Warfield, "Remus: High Availability via Asynchronous Virtual Machine Replication," in *5th USENIX Symposium on Networked Systems Design and Implementation*. San Francisco, 2008, pp. 161–174.

[4] U. Mandal, M. Habib, S. Zhang, B. Mukherjee, and M. Tornatore, "Greening the Cloud Using Renewable-Energy-Aware Service Migration," *IEEE Network*, vol. 27, no. 6, pp. 36–43, 2013.

[5] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding Data Center Traffic Characteristics," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 92–99, 2010.

[6] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, 2005, pp. 273–286.

[7] A. Murphy, "Enabling long distance live migration with F5 and VMware vMotion," *F5 Technical Report, https://f5.com/resources/white-papers/enabling-long-distance-live-migration-with-f5-and-vmware-vmotion*, 2011.

[8] T. Wood, K. Ramakrishnan, J. Van Der Merwe, and P. Shenoy, "Cloudnet: A Platform for Optimized WAN Migration of Virtual Machines," *University of Massachusetts Technical Report TR-2010-002*, 2010.

[9] K. Ye, X. Jiang, D. Huang, J. Chen, and B. Wang, "Live Migration of Multiple Virtual Machines with Resource Reservation in Cloud Computing Environments," in *2011 IEEE International Conference on Cloud Computing (CLOUD)*. Washington DC, 2011, pp. 267–274.

[10] U. Mandal, M. Habib, S. Zhang, M. Tornatore, and B. Mukherjee, "Bandwidth and Routing Assignment for Virtual Machine Migration in Photonic Cloud Networks," in *39th European Conference and Exhibition on Optical Communication (ECOC)*, 2013.

[11] U. Mandal, M. F. Habib, S. Zhang, P. Chowdhury, M. Tornatore, and B. Mukherjee, "Heterogeneous Bandwidth Provisioning for Virtual Machine Migration over SDN-Enabled Optical Networks," in *Optical Fiber Communication Conference (OFC)*, 2014.

[12] J. Chou and B. Lin, "Optimal Multi-Path Routing and Bandwidth Allocation Under Utility Max-Min Fairness," in *17th International Workshop on Quality of Service*, 2009.

[13] D. Andrei, M. Tornatore, M. Batayneh, C. U. Martel, and B. Mukherjee, "Provisioning of Deadline-Driven Requests with Flexible Transmission Rates in WDM Mesh Networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 353–366, 2010.

[14] S. Akoush, R. Sohan, A. Rice, A. W. Moore, and A. Hopper, "Predicting the Performance of Virtual Machine Migration," in *IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2010.

[15] M. Bari, M. Zhani, Q. Zhang, R. Ahmed, and R. Boutaba, "CQNCR: Optimal VM Migration Planning in Cloud Data Centers," in *IFIP Networking Conference, 2014*, pp. 1–9.

[16] W. Cerroni, "Multiple Virtual Machine Live Migration in Federated Cloud Systems," in *IEEE Conference on Computer Communications Workshops (INFOCOM Workshops)*, 2014.

[17] J. Zhang, F. Ren, and C. Lin, "Delay Guaranteed Live Migration of Virtual Machines," in *IEEE Conference on Computer Communications (INFOCOM)*, 2014, pp. 574–582.

[18] H. Jin, L. Deng, S. Wu, X. Shi, and X. Pan, "Live Virtual Machine Migration With Adaptive Memory Compression," in *IEEE International Conference on Cluster Computing and Workshops (CLUSTER)*, 2009, pp. 1–10.

[19] S. Al-Kiswany, D. Subhraveti, P. Sarkar, and M. Ripeanu, "VMFlock: Virtual Machine Co-Migration for the Cloud," in *Proceedings of the 20th international symposium on High performance distributed computing*. ACM, 2011, pp. 159–170.

[20] G. Sun, D. Liao, V. Anand, D. Zhao, and H. Yu, "A New Technique for Efficient Live Migration of Multiple Virtual Machines," *Future Generation Computer Systems*, vol. 55, pp. 74–86, 2016.

[21] A. Gupta, U. Mandal, P. Chowdhury, M. Tornatore, and B. Mukherjee, "Cost-Efficient Live VM Migration Based on Varying Electricity Cost in Optical Cloud Networks," *Photonic Network Communications*, vol. 30, no. 3, pp. 376–386, 2015.

[22] H. Maziku and S. Shetty, "Towards a Network Aware VM Migration: Evaluating the Cost of VM migration in Cloud Data Centers," in *IEEE International Conference on Cloud Networking (CloudNet)*, 2014.

[23] T. Mofolo, R. Suchithra, and N. Rajkumar, "Resource Allocation Using Virtual Machine Migration: A Survey," in *ACEEE Int. Conf. on Advances in Information Technology and Mobile Communication*, 2013.

[24] O. Ayoub, L. Pace, F. Musumeci, and A. Pattavina, "Dynamic Routing and Bandwidth Assignment for Live Virtual Machines Migrations," in *20th International Conference on Optical Network Design and Modeling (ONDM)*. Cartagena, 2016.

[25] P. D. Patel, M. Karamta, M. Bhavsar, and M. Potdar, "Live Virtual Machine Migration Techniques in Cloud Computing: A Survey," *International Journal of Computer Applications*, vol. 86, no. 16, 2014.

[26] F. Travostino, P. Daspit, L. Gommans, C. Jog, C. de Laat, J. Mambretti, I. Monga, B. van Oudenaarde, S. Raghunath, and P. Y. Wang, "Seamless Live Migration of Virtual Machines over the MAN/WAN," *Future Generation Computer System*, vol. 22, no. 8, pp. 901–907, 2006.

[27] U. Mandal, P. Chowdhury, M. Tornatore, C. U. Martel, and B. Mukherjee, "Bandwidth Provisioning for Virtual Machine Migration in Cloud: Strategy and Application," *in IEEE Transactions on Cloud Computing (Accepted for Publication)*, 2016.

Omran Ayoub (S'15) received the B.Sc. degree in Computer Communications Engineering from the American University of Science and Technology in Beirut, Lebanon in 2011 and the M.Sc. degree in Telecommunication Engineering from the Politecnico di Milano, Milano, Italy in 2015. He is currently a Ph.D. student of Information Technology at the Department of Electronics, Information and Bioengineering, Politecnico di Milano. His current research interests are in the field of 5G networks, cloud/data-center networks, information-centric and content-delivery networks.

Francesco Musumeci (S'11-M'12) received a M.Sc. degree (Laurea) in Telecommunication Engineering and a Ph.D. degree in Information Engineering from the Politecnico di Milano, Italy, in 2009 and 2013, respectively. He is currently an Assistant Professor at the Department of Electronics, Information and Bioengineering, Politecnico di Milano. His current research interests are in the field of 5G networks, fixed/mobile convergence, cloud/data-center and content-delivery networks, energy-efficiency and network resilience. Dr. Musumeci is an author of more than 40 papers in the area of communication networks, published in international journals and conference proceedings, and is co-winner of two best paper awards from IEEE sponsored conferences.

Massimo Tornatore (S'03-M'06-SM'13) received the Ph.D. degree in Information Engineering from Politecnico di Milano, Milano, Italy, in 2006. He is currently an Associate Professor with the Department of Electronics, Information, and Bioengineering, Politecnico di Milano. He also holds an appointment as Adjunct Associate Professor with the Department of Computer Science, University of California, Davis, Davis, CA, USA. He is the author of more than 200 peer-reviewed conference and journal papers. His research interests include performance evaluation, optimization and design of communication networks (with an emphasis on the application of optical networking technologies), cloud computing, and energy-efficient networking. He is a member of the Editorial Board of Photonic Network Communications, Optical Switching and Networking and IEEE Communication Surveys and Tutorials. He is an active member of the technical program committee of various networking conferences such as INFOCOM, OFC, ICC, and GLOBECOM. He is a co-recipient of ten best-paper awards.

Achille Pattavina (SM'93) received the Dr.Eng. degree in Electronic Engineering from the University of Rome La Sapienza, Rome, Italy, in 1977. He was with the same university until 1991, when he moved to Politecnico di Milano, Milan, Italy, where he has been a Full Professor since 1995. He has been the author or coauthor of more than 200 papers in the area of communications networks published in leading international journals/conferences and of two books: Switching Theory, Architectures and Performance in Broadband ATM Networks (New York: Wiley, 1998) and Communication Network: Networking and Internets (McGraw-Hill, 2nd ed., 2007, in Italian). He has been the coordinator of national and international research activities, including European Union funded projects. His current research interests are in the areas of green ICT and cloud computing, optical networks, switching theory, traffic modeling, and broadband convergent access/metro networks. He has been the Editor for Switching Architecture Performance of the IEEE TRANSACTIONS ON COMMUNICATIONS from 1994 to 2011 and the Editor-in-Chief of the European Transactions on Telecommunications from 2001 to 2010.