SPA: Smart Placement Approach for Cloud-service Datacenter Networks

Ahmad Nahar Quttoum*, Mohannad Tomara, Bayan Khawaldeh, Rana Refai, Alaa Halawani, Ahmad Freeja

*Computer Engineering Department, Faculty of Engineering, The Hashemite University, Zarqa 13133, Jordan

Abstract

Business disciplines expand rapidly, so does the tendency to rely on computer applications and its varying services to support such expansion. Often, this is achieved through introducing physical network infrastructures that provide the appropriate environments to run such applications. The required services change rapidly, and accordingly its resource requirements. In most cases, this may require building new physical networks which could lead to low utilization rates and high service-costs. A promising approach that is becoming increasingly popular to overcome such a problem is known as Virtual Datacenter Networks (VDNs). These VDNs are usually hosted over physical networks; overlaying its resources to gain the dynamic required services. Cloud-service Datacenter Networks (CDNs) can provide such a service under a delivery model that is called Infrastructure as a Service (IaaS). In this paper, we present a Smart Placement Approach (SPA) that provides for smart placement maps of VDNs over CDNs. In this, we point out that choosing the placement maps for such VDNs should satisfy its requirements while: maintaining load-balancing over the hosting CDNs, guaranteeing its Quality of Service (QoS) levels, and assuring low placement costs.

Keywords: Cloud-Service DataCenter Networks; Virtual DataCenter Networks; Smart Placement Schemes; Load-balancing Networks.

1. Introduction

Cloud-service Datacenter Network (CDNs) use the infrastructure of physical networks architectures to provide reliable and cost-effective streams of services. Indeed, prior the rise of cloud computing, IT business owners and computing companies were obliged to buy their own physical resources in order to run their applications. Beside being costly, that approach does not provide for flexible scale and agility solutions. Administrators of such physical networks need to take the burden of planning ahead for any provisioning and expansion requirements in their networks. This imposes more costs and management overhead. The theme of CDNs delivers attractive solutions. Truly, with such scheme of virilization, CDNs can provide the resource requirements to the end-users (i.e. IT business holders) in the form of a service. In the aforementioned scenario, this comes to a whole network Infrastructure as a Service (IaaS). Right, end-users have to pay only for the services they receive without worrying about: (1) up-front investments, (2) future scalability challenges, (3) management and administrative burden, or even (4) being limited to certain physical location.

* Corresponding author. Tel.: +962-5-3903333; ext: 4813, 4859.
1 Students that studied this article as part of one of their courses.
E-mail address: Quttoum@hu.edu.jo

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Conference Program Chairs
This provides for considerable opportunities and attractive solutions. However, it also conveys a great challenge that needs to be carefully considered by the CDNs administrators. Indeed, they need to follow special techniques that facilitate such business and allow for cost-effective high performance services. The basic idea in IaaS is Network Virtualization (NV). This could be implemented through different virtualization environments (e.g. Xen and VMware) that hosts (places) Virtual Machines (VMs) on top of physical ones. Talking about hosting a Virtual Datacenter Network (VDN) which we denote as infrastructure, this involves many types of resources, starting from the CPU cycles of the servers to the memory space of the look-up tables in the switching devices. So, to provide such IaaS, CDNs administrators need to dynamically check the resource availability over their physical resources (e.g. server, links, switches), check the anticipated hosting costs, any possible affects on performance and accordingly take a decision.

In this paper, we propose a Smart Placement Approach (SPA) that provides the CDNs administrators with a proper tool to find the best placement maps to host the VDNs. In SPA, we developed a model that dynamically studies the status of whole nodes and links of a CDN, and according to: (1) the VDN resource requirements, (2) the network performance constraints, and (3) the incurred hosting costs, it finds the best placement map. Different from other proposals in the literature, our contribution is a model that: (1) Finds the best placement map for a full VDN. (2) Provides for load-balancing scheme. (3) Maintains Quality of Service (QoS) guarantees. (4) Minimizes the incurred hosting costs.

The rest of the paper is organized as follows: Section 2 presents the related work and the problem statement. Section 3 discusses our placement model and the SPA selection algorithm. Section 4 shows an illustrative example and simulation results, then finally Section 5 concludes the paper.

2. Related Work and Problem Statement

The authors in proposed a model for placing Virtual server Machines (VMs) over Physical server Machines (PMs). In their work, the placement decision is made according to the total completion times (for the jobs/services at the hosting PMs), where the proposal suggests following one of two scenarios (1) direct placement, and (2) migration-based placement. According to the resultant total completion times of the aforementioned two scenarios, a decision is made by choosing the one that provides the shorter completion time. In the first scenario (i.e. the direct placement), the model estimates the anticipated completion time for the service(s) done by the migrating VM over the candidate PMs and then chose the PM that provides the lowest total completion time. The chosen candidate PMs are those who have sufficient amount of resources enough to host the migrating VM. In the second scenario (i.e. the migration-based placement), the model chooses the PM that provides for the shortest total completion time, whether is has the required resources to accommodate the VM or not. In the case of not having the required resources, the model proposes migrating (re-placing) one of the already placed VMs over this chosen PM (i.e. the PM who provides for shortest completion time) to another PM in a way to free some resources to accommodate the new migrating VM. The aforementioned proposal has several limitations that could be summarized as:

- First, the candidate PMs are chosen in a way that does not consider that fact that reaching those PMs is done through a network that consists of several nodes (i.e. the connecting devices like switches and routers) interconnected via set of links (i.e. bandwidth resources). Checking the resource availability at the servers only is not enough to provide a proper candidate to hold the required applications or the needed services. Typically, this model will result in a non-optimal placement decision. Indeed, where being part of a network, it is some times more crucial to check the resource availability over the network components (e.g. switches and links) that interconnects such servers with the end-users. Without these resources, servers capacity is useless!

- Second, the migration-based placement scenario allows migrating the already placed VMs from one PM to another in a way to make space for a new VM to be place instead. This is not efficient, as those VMs that already been placed and run over a PM may have functional dependencies between each other or with other VMs from different PMs. Migrating (re-placing) such VMs to new PMs may not be a feasible process, as this may cause several problems (e.g. service failures, interruptions) for many VMs which may violate the Service Level Agreements (SLAs) and the contracted QoS guarantees.
Third, the proposed model does not consider the load-balancing issues. Choosing the candidate PMs according to the total completion time does not necessarily comply with the load-balancing objectives. Administrators of Cloud-service networks always aim to maximize their revenue objectives while not violating the SLAs and the QoS guarantees provided to their customers. To do so, they tend to keep the loads over their networks balanced in a way to avoid congestions or any other problems that may harm the network performance.

To overcome the aforementioned limitations, we propose a new placement approach that takes into account the whole network resources (i.e. connecting devices and link) in addition to the interconnected servers. Through which, we are providing an smart placement map that guarantees the SLAs and load-balancing objectives while not harming or causing any inconvenience for the network customers (i.e. those VMs that are already in place). To do so, we developed a model that studies the status of the whole components (i.e. switches, links, servers) of the hosting networks before considering any placement decision for any migrating device. Having this in hand, we chose to expand our approach to deliver the infrastructure as a service which is another model for the cloud-computing services. Accordingly, our proposed approach provide for mapping a whole virtual network (e.g. a virtual datacenter network) to a new physical one. Knowing the status of the physical network devices with the requirements of the virtual network to be placed, our smart placement model chooses those least-loaded physical devices (that constitute a network) to host the virtual ones while providing for lower placement costs and SLAs guarantees.

3. The SPA: The Smart Placement Approach

In this section, we present the SPA model that overcomes the limitations discussed in Section 2 by offering a smart placement approach. An approach that takes into account the availability of the whole network resource, while guaranteeing load-balancing and SLAs objectives. To achieve this, migrating Virtual DataCenter Networks (VDNs) should clearly specify its detailed resource requirements (i.e. the resource vector) to the hosting physical network. This can provide for optimal placements and satisfying services. In this context, requirements may vary from a virtual network to another, depending on the considered topologies and the provided services.

However, among all the network components, the challenge for the hosting CDNs (i.e. the physical ones) mainly lies in the switching capabilities of its network, more precisely, its path processing capacities. Indeed, where for a packet to get processed through a switching device, certain resources are required. In this context, let us define the physical switch as a set of virtual switches, where each virtual switch operates a set of virtual switching paths. Mainly, a virtual switching path to operate requires a set of: (1) packet processing resources (network processor cycles, search caches, memories); (2) ports; (3) bandwidth over the ports. Typically, for a packet processing task to operate, this requires: (1) processors (for parsing and analysis); (2) memories (for the lookup tables) that can be either internal or external (e.g. TCAMs, SRAMs); (3) queues (for packets’ scheduling and storage, and for the process of shaping priorities); (4) bandwidth over the busses that interconnect the aforementioned internal components.

Accordingly, such physical resources will be virtually partitioned among the different virtual data paths that are allocated (reserved) to satisfy the requirements of the VDNs topologies. Hence, for efficient allocations and optimal placement decisions, such resource vectors need to be clear in order to check for resource availability at the hosting physical network.

Therefore, to simplify the presentation, the resource requirements of the migrating networks (i.e. the VDNs) will be represented by the virtual data path capacity. At the hosting side (i.e. the hosting CDNs topology), this will be translated to path processing resources, being a parameter for the placement process.

Thus, having the resource vector of the migrating VDN, the hosting network administrator can break this down to: (1) ports; (2) processing engine capabilities; (3) memories; (4) internal bandwidth; all constrained by certain speed/delay limits for QoS assurance. Besides, the administrator need to specify the external bandwidth requirements (bandwidth over the links that interconnects the different switching devices), this could be defined by the Network Interface Cards (NICs) capacities.
Based on that, in the following, we are proposing a model that can help in providing optimal placement/mapping decisions for the VDNs over a physical CDN topology. To do so, we will start by:

3.1. The Physical DCN Topology:

A CDN topology can be modeled as a weighted graph denoted by \( G^p = (S^p, L^p) \), where \( S^p \) represents the set of switch and server devices that are interconnected by the set of links \( L^p \) (representing the line-cards bandwidth capacities).

For each physical device \( s^p \), we define: (1) its path processing capacities \( \mathcal{I}(s^p) \) (this includes the different data channels and their associated resources like processors, memories, busses, and queues); (2) its line-cards’ indices and their associated bandwidth capacities \( b^p_{li} \) (i refers to the line-card index); (3) and its placement location \( \text{loc}(s^p) \). Accordingly, a link that provides connectivity among two network devices is defined by their line-cards’ capacities \( b^p_{li} \), where the total bandwidth capacity over a physical link can be denoted by \( b(l^p) \).

Paths from a source location \( s \) to a destination \( d \) are denoted by \( H^p_{s,d} \), where a path \( H^p \) can consist of one or more network links.

3.2. The Virtual DCN Topologies:

Same as the physical topology, a VDN can be modeled as a weighted graph \( G^v = (S^v, L^v) \). Such modeling can help in expressing the VDN requirements based on the physical network attributes. Each topology (of the VDNs) may have its own constraints and special requirements that vary among servers’ characteristics and bandwidth capacities.

3.3. The Model:

The network placement process is modeled as a two-sided problem, the virtual side is presented in section 3.3.1 and the physical side is presented in section 3.3.2.

3.3.1. Measurements at the VDN side

When it comes to the VDN, the main interest would be in getting the required interconnection paths between certain couples of source-destination nodes. This implicitly means: (1) the bandwidth resource over the links that interconnect the in-path switches; and (2) the in-switch virtual data paths (through the switches). Hence, the objective for a virtual request \( r^v \) can be represented by attaining its required resources that is defined as:

\[
\mathcal{R}(r^v) = \sum_{l^v \in L^v} b(l^v) + \sum_{s^v \in S^v} \mathcal{I}(s^v)
\]  

where it equals the sum of bandwidth resources added to the sum of path processing units allocated over the physical network resources.

3.3.1.1. Hosting Costs: Notice that the aforementioned formula can directly give an insight to the costs incurred by the physical provider to host the received request. Accordingly, the cost function for hosting a full VDN can be given by:

\[
C(G^v) = (x) \sum_{l^v \in L^v} \sum_{l^p \in L^p} b^p_{li} + (y) \sum_{s^v \in S^v} \sum_{s^p \in S^p} \mathcal{I}(s^p)
\]

Here we are taking into consideration the resources of the whole VDN (nodes, links). This means the total cost of all resources reserved for the usage of a VDN. This involves checking all the physical network nodes \( S^p \) and links \( L^p \) for any reservations related to the considered VDN. The notations \( x \) and \( y \) represents the cost-units set by the hosting network for leasing its resources. Note that \( y \) may get changed to a set of several cost-units, each associated to a certain type of resources (i.e. cpu processing resources, memory, ports’ capacity, ...). The term \( b^p_{li} \) refers to the total amount of bandwidth reserved for the virtual link \( l^v \) over the physical link \( l^p \).
3.3.2. Measurements at the CDN side

At the physical network side, the processing and bandwidth resources over each network node, are measured dynamically as follows:

3.3.2.1. Path Processing:. The available capacity at a given physical node \( R_\psi(s^\psi) \) could be defined by:

\[
R_\psi(s^\psi) = \mathcal{I}(s^\psi) - \sum_{\forall s^\nu \in s^\psi} z(s^\nu)
\]

(3)

Which refers to the total capacities of the node \( \mathcal{I}(s^\psi) \), reduced by the sum of the resources \( z(s^\nu) \) allocated for each virtual node \( s^\nu \) being already hosted. The term \( \forall s^\nu \uparrow s^\psi \) refers to the virtual nodes that are currently hosted in the corresponding physical node.

The terms \( \mathcal{I}(s^\psi) \) and \( z(s^\nu) \) are measured as a function of their components. As an example, for the \( \mathcal{I}(s^\psi) \), this means a matrix of: (1) processing engine capacity \( \tau(s^\psi) \); (2) memory resources \( \omega(s^\psi) \); (3) queuing \( \nu(s^\psi) \); (4) busses’ bandwidth \( \upsilon(s^\psi) \). This could be illustrated as:

\[
\mathcal{I}(s^\psi) = f(\tau(s^\psi) + \omega(s^\psi) + \nu(s^\psi) + \upsilon(s^\psi))
\]

(4)

3.3.2.2. Bandwidth capacity:. The available bandwidth capacities at any of the physical links \( R_b(l^\rho) \) can be measured as:

\[
R_b(l^\rho) = b(l^\rho) - \sum_{\forall l^\rho' \uparrow l^\rho} b(l^\rho')
\]

(5)

Which refers to the total link capacity \( b(l^\rho) \), reduced by the sum of bandwidth amounts allocated for each virtual link \( l^\rho \) using the considered link resources. The notation \( l^\rho \) is used to identify the switch/server virtual line-card from which the virtual link is initiated. Similarly, the term \( \forall l^\rho' \uparrow l^\rho \) refers to the virtual links that are hosted in the corresponding physical link.

In this context, the available bandwidth capacity of a chosen path (say \( H^\rho \)) is defined by the lowest capacity available at any of its links as follows:

\[
R_b(H^\rho) = \min_{l^\rho \in H^\rho} R_b(l^\rho)
\]

(6)

The term \( l^\rho \in H^\rho \) is used to refer to the links that are used to constitute the path. Indeed, where for any path, the amount of bandwidth capacity that is available for any connection that wants to pass from server/switch \( x \) to server/switch \( y \) is limited by the lowest bandwidth capacity among all the links from node \( x \) to node \( y \).

3.3.3. The Smart Placement Algorithm

According to the required resource vector and the measurements done at the physical nodes of the hosting network, to find the set of candidate networks to host the VDN, the proposed model deploys the the Algorithm presented in Table 1.

Initially, as an input to the SPA model, the resource vector of the VDN is given to the administrator of the hosting physical network (i.e. the CDN). In step (1), the model reads the inputs of the VDN resource vector and accordingly in step (1.1) it gathers the entries of \( G^\nu \) which illustrates the topological requirements of the VDN. Similarly, in step 1.2, it reads the specifications and the detailed status vector of the CDN to prepare \( G^p \) that illustrates its status. Next, for each node \( s^\psi \in S^\psi \), it breaks down the resource vector to the node’s detailed requirements in step 1.3. As illustrated in Equation 4, this checks the requirements for each virtual node \( s^\nu \) (i.e. switch or server). Now, in step 2, among the physical nodes \( S^p \), find the set of nodes \( N(s^p) \) that satisfy the resource availability constraints presented in Equation 7. Among this set of candidate nodes \( N(s^p) \), in step 3, the model finds the set \( N(s^\psi) \) that lists those nodes that satisfy the bandwidth requirements over their Network Interface Cards (NICs). This is required to provide connectivity between the nodes of \( N(s^\psi) \) in a way to find the candidate placement networks \( N(G^\rho) \) in step 4. For those placement networks in \( N(G^\rho) \), in step 5, the model finds the best candidate according to the formulas presented in Equations 8 and 9. Finally, step 6 updates the resource availability metrics of the CDN to be considered when hosting new VDNs.

Ahmad Nahar Quttoum et al. / Procedia Computer Science 56 (2015) 341 – 348
through Gbit-ethernet NICs (1Gbps). Each edge switch is connected to two heavy-duty servers via fast-ethernet NICs that interconnect the nodes together in order to form the adequate network topology (Figure 1.b) is consisted from: (a) two core switches \((V_{c})\), interconnected through 10Gbit-ethernet NICs (10Gbps) to (b) three aggregate switches \((V_{a1}, V_{a2}, V_{a3})\), each is interconnected to a couple of edge switches through 2 fast-ethernet NICs (100Mbps), (c) six edge switches \((V_{e1}, V_{e2},..., V_{e6})\). Each of these edge switches connects one or two servers \((V_{s})\) through standard-ethernet NICs (10 Mbps). The physical network topology (Figure 1.b) is consisted from: (a) two core switches \((P_{c1} and P_{c2})\), interconnected through 10Gbit-ethernet NICs (10Gbps) to (b) six aggregate switches \((P_{a1}, ..., P_{a6})\) that connects (c) eight edge switches \((P_{e1}, ..., P_{e8})\) through Gbit-ethernet NICs (1Gbps). Each edge switch is connected to two heavy-duty servers via fast-ethernet NICs (100Mbps).

### 3.3.4. The Placement process

Having the status of the hosting physical nodes and links got calculated in sections 3.3.1 and 3.3.2, in the following we show how the candidate physical network is chosen. The formula in Equation 7 shown below defines how the model finds the nodes of \(S^{p}\) that satisfy the resource availability constraints.

\[
\tau(s^{p}) \geq \tau(s^{s}), \omega(s^{p}) \geq \omega(s^{s}), \nu(s^{p}) \geq \nu(s^{s}), \upsilon(s^{p}) \geq \upsilon(s^{s})
\] (7)

These chosen nodes from the set \(S^{p}\) that satisfy the aforementioned formula are gathered in a new set called \(N(s^{p})\). The nodes of this new set are then checked to find which of them can provide the required bandwidth capacities over its Network Interface Cards (NICs). These NICs that interconnect the nodes together in order to form the adequate placement topologies of the VDN (i.e. the candidate networks, \(N(G^{p'})\)).

Each node of these candidate networks in \(N(G^{p'})\) is examined under the formula presented in Equation 8, and accordingly, the total amount of resource availability over each node in the candidate networks is calculated.

\[
\mathcal{I}(s^{p'})_{G^{p'}} = f[\eta \tau(s^{p}) + \gamma \omega(s^{p}) + \sigma \nu(s^{p}) + \varphi \upsilon(s^{p})]
\] (8)

where \(\eta, \gamma, \sigma, \varphi\) are parameters to control the importance (i.e. the weight) of each type of resources, as an example, these can be used to provide for priorities when choosing the hosting network switches. The sum of these weight parameters must = 1. With such outputs, among the set of candidate networks, the system finds the total resource availabilities over the chosen candidate networks \(N(s^{p'})\). Now, it solves Equation 9 to find the best candidate network \(G^{h}\) that has the highest resource availability (load-balancing) and the lowest hosting costs (profit maximization).

\[
G^{h} = \max_{s^{p'} \in N(G^{p'})} (\mathcal{I}(s^{p'}) - C(s^{p'}))
\] (9)

### 4. Illustrative Example and Simulation Results

To assess the efficiency of our proposed model, in Figure (1.a) we illustrate an example of a fat-tree VDN structure\(^7\) that looks for a placement over a physical network. This VDN is consisted of: (a) one core switch \((V_{c})\), interconnected through 3 fast-ethernet NICs (100Mbps) to (b) three aggregate switches \((V_{a1}, V_{a2}, V_{a3})\), each is interconnected to a couple of edge switches through 2 fast-ethernet NICs (100Mbps), (c) six edge switches \((V_{e1}, V_{e2},..., V_{e6})\). Each of these edge switches connects one or two servers \((V_{s})\) through standard-ethernet NICs (10 Mbps). The physical network topology (Figure 1.b) is consisted from: (a) two core switches \((P_{c1} and P_{c2})\), interconnected through 10Gbit-ethernet NICs (10Gbps) to (b) six aggregate switches \((P_{a1}, ..., P_{a6})\) that connects (c) eight edge switches \((P_{e1}, ..., P_{e8})\) through Gbit-ethernet NICs (1Gbps). Each edge switch is connected to two heavy-duty servers via fast-ethernet NICs (100Mbps).

### Table 1: VDNs’ Smart Placement Algorithm

<table>
<thead>
<tr>
<th>Algorithm 1: Smart Placement Algorithm for a VDN Topology over a Physical One</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> At time (t), a VDN placement request (k) is arrived to the administrator of the CDN</td>
</tr>
<tr>
<td>1: for each VDN placement request, do:</td>
</tr>
<tr>
<td>1.1: read the entries of (G^{c} = (S^{c}, L^{c}));</td>
</tr>
<tr>
<td>1.2: read the entries of (G^{p} = (S^{p}, L^{p}));</td>
</tr>
<tr>
<td>1.3: brake down the resource vector of request (k) to its implicit requirements:</td>
</tr>
<tr>
<td>({\tau(s^{c}), \omega(s^{c}), \nu(s^{c}), \upsilon(s^{c})});</td>
</tr>
<tr>
<td>2: among the set (S^{p}), find the set of nodes (N(s^{p})), where:</td>
</tr>
<tr>
<td>({\tau(s^{p}) \geq \tau(s^{c}), \omega(s^{p}) \geq \omega(s^{c}), \nu(s^{p}) \geq \nu(s^{c}), \upsilon(s^{p}) \geq \upsilon(s^{c})});</td>
</tr>
<tr>
<td>3: among the set (N(s^{p})), find the set of nodes (N(s^{p'})), that satisfies:</td>
</tr>
<tr>
<td>({L^{p} \geq L^{c}}) at its NICs;</td>
</tr>
<tr>
<td>4: among the set (N(s^{p'})), find the set of candidate placement networks (N(G^{p'}));</td>
</tr>
<tr>
<td>5: having the set (N(G^{p'})), find the optimal placement network (G^{h}) among the set ((G^{p'})) according to Equations 8 and 9</td>
</tr>
<tr>
<td>6: update the values of (R_{b}(s^{p})) and (R_{c}(l^{p}));</td>
</tr>
</tbody>
</table>

---

\(^7\) having the set \(G^{p'}\), find the set of candidate placement networks \(N(G^{p'})\);
An example of a detailed resource vector for a VDN is shown at Table 2. It could be noticed that such resource requirements vary from a layer to another (e.g. core, aggregate, edge), representing the different functional roles of such devices according to their position in the topology.

Table 2. Resource Vector; a detailed resource requirements for a Virtual Datacenter Network (VDN)

<table>
<thead>
<tr>
<th>Virtual Device</th>
<th>Processing Capacity</th>
<th>Internal Memory</th>
<th>Queueing/H.D. Space</th>
<th>NICs Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>One Core Switch</td>
<td>3500MHz</td>
<td>120Mb</td>
<td>30Mb</td>
<td>300Mbps</td>
</tr>
<tr>
<td>Three Aggregate Switches</td>
<td>200MHz</td>
<td>60Mb</td>
<td>10Mb</td>
<td>300Mbps</td>
</tr>
<tr>
<td>Six Edge Switches</td>
<td>1500MHz</td>
<td>40Mb</td>
<td>5Mb</td>
<td>150Mbps</td>
</tr>
<tr>
<td>Servers (e.g. storage ones)</td>
<td>1GHz</td>
<td>2Gb</td>
<td>5Tb</td>
<td>10Mbps</td>
</tr>
</tbody>
</table>

For the VDN shown in Figure (1.a), we run our SPA model to find the best placement map over the physical network shown in Figure (1.b). This physical network represents a cloud-service datacenter network that, in our scenario, provides the end-users with a whole infrastructure as a service. Adopting the proposed methodology presented in section 3.3, we simulate the placement process of the aforementioned network. Consequently, after reading the resource vector from an input text file, the SPA checks the availability of the physical network resources (i.e. resource of its switches, servers, and links). Then it prepares the list of candidate nodes and placement networks according to the detailed placement algorithm depicted in Table 1. By doing so, the simulation results are presented in the following:

Figure 2 shows two different candidate placement networks. In fact, the simulation results provided six candidate networks, but for the space limitation and paper length constraints, we are presenting only a subset of them. In the figures, those chosen nodes and links are presented in black, and those not chosen are in red. Based on the resource availability metrics that we assumed for the physical network components, it is clearly noticed that both of the core switches ($Pc_1$ and $Pc_2$) have sufficient amount of resources to host the virtual switch ($Vc$). This might be expected as core switches in general are always superior in terms of resource availability if compared with the aggregate and edge ones.

Beside the core switch $Pc_1$, the chosen candidate network shown in Figure (2.a) consists of the aggregate switches ($Pa_1$, $Pa_2$, and $Pa_5$), being interconnected to the edge switches ($Pe_1$, $Pe_2$, $Pe_4$, $Pe_5$, $Pe_8$) which are connected to the underlying physical servers. Note that the edge switch $Pe_2$ is chosen to interconnect both aggregate switches $Pa_1$ and $Pa_2$ with the underlying servers $s_3$ and $s_4$, respectively. This implicitly mean that the edge switch $Pe_2$ have available capacity that is sufficient enough to support two virtual switches concurrently (i.e. $Va_1$ and $Va_2$). Figure (2.b) show another candidate network that chose the core switch $Pc_2$ with the aggregate switches $Pa_1$, $pa_2$, and $Pa_4$ instead of $Pa_1$, $Pa_2$, $Pa_5$ that are chosen for the other candidate network in Figure (2.a). Same way, the edge switches $Pe_4$ and $Pe_5$ are chosen instead of the couple $Pe_5$ and $Pe_8$ to be connected with the aggregate switch $Pa_4$. 
In such a case, servers $s_7$ and $s_8$ are expected to host two different virtual servers each. This implicitly means higher resource usage and less residuals. According to the formula illustrated in Equation 9, among the aforementioned four candidate networks, our simulation result shows that the network depicted in Figure (2.b) has been chosen to be the placement network that provides the highest resource availability and the lowest operational costs. Form this, we can conclude using those component (i.e. switches, servers, and links) of the chosen network have the maximum amount of available resources according to Equation 8, and the lowest costs according to Equation 2. This provides for load-balancing, leads to QoS guarantees, and minimizes the network operation costs.

5. Conclusion

Interesting models that tackle the problem of virtual machines placement are already proposed in the literature. However, to the best of our knowledge, no work tackles the problem of placing a full network as one package. In this context, placement models should provide for performance optimality for the whole parties involved in the placement process. Indeed, where is not a wise solution to place a virtual machine in a way that fulfills its performance requirements but causing problems to others. Considering the placement costs is also a crucial factor in such decision, however, other factors like load-balancing and QoS guarantees are also important. Our proposal, the SPA model, provides an approach that allows for smart placement decisions of a whole VDN over a Cloud-service Datacenter Network. SPA delivers placement maps that allow for load-balancing, QoS guarantees, and low operational costs.

References