Dynamic Virtual Network Traffic Engineering with Energy Efficiency in Multi-Location Data Center Networks

Mirza Mohd Shahriar Maswood[1*+], Chris Develder[2], Edmundo Madeira[3§], Deep Medhi[1+§]
[1]University of Missouri–Kansas City, USA, [2]Ghent University – iMinds, Belgium,
[3]University of Campinas, Brazil
Email: mmnt7@mail.umkc.edu, chris.develder@intec.ugent.be, edmundo@ic.unicamp.br, dmedhi@umkc.edu

***Abstract*—For cloud enterprise customers that require services on demand, data centers must allocate and partition data center resources in a dynamic fashion. We consider the problem in which a request from an enterprise customer is mapped to a virtual network (VN) that is allocated requiring both bandwidth and compute resources by connecting it from an entry point of a data center to one or more servers, should this data center be selected from multiple geographically distributed data centers. We present a dynamic traffic engineering framework, for which we develop an optimization model based on a mixed-integer linear programming (MILP) formulation that a data center operator can use at each review point to optimally assign VN customers. Through a series of studies, we then present results on how different VN customers are treated in terms of request acceptance when each VN class has a different resource requirement. We found that a VN class with a low resource requirement has a low blocking even in heavy traffic, while the VN class with a high resource requirement faces a high service denial. On the other hand, cost for the VN with the highest resource requirement is not always the highest in the heavy traffic because of the significantly high service denial faced by this VN class.**

***Index Terms*—Data Center Networks, Resource Optimization and allocation on-demand, Denial of Service, Energy Efficiency, Virtual Network**

I. Introduction

With the increasing dependency on various services such as e-commerce, electronic libraries, video streaming, and audio-video conferencing, the need for both compute and storage has significantly increased. To cater to these needs, cloud data centers have become a popular platform in recent years. Today, companies such as Amazon, Google, Facebook, and Yahoo! routinely use data centers for storage, web services, and large-scale computations [1], [2], [3]. Because of this increase in the use of data centers, a cost-effective system design for storage and processing data has become a challenging problem. This increasing need for equipment such as routers, switches, and server racks in data centers also incurs significant power consumption that contributes to the operational cost of data centers.

There has been significant work so far to improve the capability of data centers by increasing the utilization of the servers and reducing operational cost. However, little research has been conducted on dynamic traffic engineering for handling requests for different customers and how both network resources in the data center and compute resources at the hosts are allocated for such customers. We also observe that most work related to traffic engineering of intra DC networks considers east-west traffic, i.e., the intra-data center traffic between hosts. In our work, we focus instead on enterprise customers' requests that result in north-south traffic in data centers. We focus on serving different customer groups using virtual networks (VNs) at data centers through dynamic traffic engineering by allocating both network bandwidth and processing resources efficiently, while factoring in energy consumption. We present a dynamic traffic engineering framework where virtual network customers are served at review points. At each review point, we propose to solve a traffic engineering problem for arriving requests from virtual network customers. The requests that are admitted by this process then use resources for a certain duration. An important contribution in this work is that we consider a request to consist of a two-tuple demand, one for data center network bandwidth and the other for the processing demand at the end hosts.

Thus, our work is different from existing work in two distinct ways. First, we consider the north-south traffic environment where each request consists of a two-tuple demand model. Furthermore, we consider this for virtual network customers by taking issues such as power consumption at the hosts into consideration. We present a novel mixed-integer linear programming (MILP) formulation to solve at each review point that minimizes a composite objective from a traffic engineering point of view to satisfy the virtual network customers by using minimum resources from data centers. Our formulation allows for the flexibility that requests arriving at a review point may be allocated to any of the available data centers; for the selected data center, it may use any of the entry points for the north-south traffic at the north end, and any of the hosts available at the south-end.

The second contribution of this work is to present an insight on how different VN customers are affected in terms of resource allocations with north-south traffic in data centers. That is, there are a number of questions to which we seek

Acknowledgement: *Partially supported by a School of Graduate Studies Research Grant at the University of Missouri–Kansas City, +Partially supported by National Science Foundation Grant # 1526299, §Partially supported by CNPq (National Counsel of Technological and Scientific Development), Brazil.

978-0-9883045-1-2/16 $31.00 © 2016 ITC

to find answers. How are the cost and blocking affected as the request arrival rate from VN customers increases? When the bandwidth demand and the resources per request vary uniformly from an average value, how are the cost and blocking affected compared to when the bandwidth demand and resources for each request were kept fixed? Furthermore, how are different VN classes affected in terms of cost and blocking when each class has a different bandwidth and CPU resource demand? Does the system favor one VN class over another? If so, by how much? Finally, we wish to know how much the power consumption is reduced by our optimization model. By considering a number of cases in a systematic manner, we were able to answer to these questions.

The rest of the paper is organized as follows. In Section II, we present the optimization formulation of the traffic engineering problem to be solved at each review point. In Section III, we present the simulation setup and results of our analysis. The related work is discussed in Section IV. Finally, in Section V, we summarize our concluding remarks and discuss potential future work.

II. Model Formulation

Our dynamic traffic engineering approach considers new request arrivals at random from customers, for which the resource allocation (both data center network bandwidth and host resources) is done at review point $t \in T$, where T is a discrete temporal window for dynamic traffic engineering consisting of review points. The duration of a new VN request that uses the data center is assumed to be random. Note that since the data center is set up to serve VN customers, at any time instant, there are existing VN tunnels and host resources allocated for prior requests. Thus, any (micro-)workload that needs immediate access to resources, that is, workload that cannot wait until the next review point, is assumed to be served by existing VN channels and host resources assigned to the customers that were set up at earlier review points. Since such immediate workloads are served through existing resources, they are not modeled in our case. In other words, the scope of our work is to consider new requests at review points that are major requests requiring allocation of new bandwidths, virtual network tunnels and new resources.

We optimally solve the resource allocation problem for traffic engineering at each review point t. For this, we first present a mixed-integer linear programming (MILP) formulation in which we attempt to accommodate as many requests as possible while minimizing the resources requirement towards satisfying those requests in order to reduce the overall cost. To illustrate our approach, consider the single data center network topology shown in Fig. 1, which depicts just one site of the multi-location data center that our model considers. The entry point in a data center is then the north-end and the serving host is the south-end of the north-south traffic. Our approach assumes that there is a central controller that is responsible for solving the proposed optimization model and setting up the allocations. For instance, this can be accomplished by using a software-defined network (SDN) based approach.

In our model, each request consists of 2-tuple $\langle h, r\rangle$ where h is the bandwidth demand of the request and r is the processing resources required from a serving host. Thus, at a particular review point t, if a VN customer $v \in V$ has a request, the request tuple is further represented by $\langle h^v(t), r^v(t)\rangle$, which is to be served by data center $d \in D$. While the bandwidth demand needs to be satisfied by the capacity of the links within the data center $l \in L_d$ from the entry point $i \in I_d$ to a server $j \in J_d$, the processing resources must be satisfied by the servers' available resources. We assume that there is a given set of paths $P_{ij}^{vd}(t)$ from the entry point i to server j, which could be potentially different at each review point t.

For energy consumption, we consider that every server can run at a given set of CPU frequencies $f \in F$. At each particular frequency, a server works at a particular processing capacity a_{jf}^d. A specific amount of power b_{jf}^d is required to run the server at that frequency. If we run the server at the highest frequency, it offers the highest processing capacity, but consumes the highest amount of power. All notations used in our model are summarized in Table I.

We now present the constraints in our formulation. First, one DC out of the N DCs ($D = \{DC_1, ..., DC_N\}$) is at most selected to meet the request for a VN v at review point t:

$$\sum_{d \in D} u^{vd}(t) \le 1, \quad v \in V \tag{1}$$

The total link bandwidth demand must then be served by the chosen data centers:

$$\sum_{d \in D} s^{vd}(t) = h^v(t), \quad v \in V \tag{2}$$

Once a data center is responsible to fulfill the link bandwidth demand from a VN, then this data center must be the one from which the capacity is allocated:

$$s^{vd}(t) \le h^v(t) u^{vd}(t), \quad v \in V, d \in D \tag{3}$$

The total amount of the link bandwidth demand from a particular VN v that will be served by a particular data center d is the summation of the bandwidth that is allocated from all chosen entry points i to all chosen servers j of data center d at review point t:

$$\sum_{i \in I_d} \sum_{j \in J} y_{ij}^{vd}(t) = s^{vd}(t), \quad v \in V, d \in D \tag{4}$$

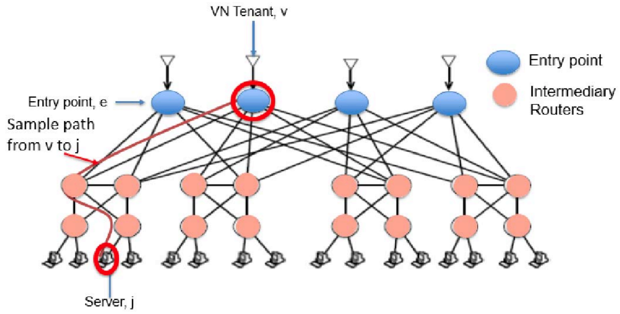

Fig. 1. Data Center Topology [4]

TABLE I
NOTATIONS USED IN FORMULATION

Constants/Parameters:
D = Set of data centers, $N = \#(D)$
J_d = Set of servers in one data center
I_d = Set of entry points in one data center
V = Set of virtual networks
F = Set of frequencies in which a particular server can run
L_d = Set of links in one data ceneter
$P_{ij}^{vd}(t)$ = Set of paths from an entry point i to a server j in a data center d for a VN v at time t
M = A large positive number
ε = A very small positive number
b_{jf}^{d} = Power consumption in server j of data center d at frequency f
$h^{v}(t)$ = Bandwidth demand for a VN v at time t
$r^{v}(t)$ = CPU processing capacity demand for VN v at time t
a_{jf}^{d} = Capacity of server j of data center d at frequency f
$c_{l}^{d}(t)$ = Available capacity on link l of a data center d at time t
$\delta_{ijpl}^{vd}(t)$ = Link-path indicator: 1 if a particular path p which is set up from an entry point i to a server j uses link l of a data center d in order to satisfy a request generated by a VN v that enters entry point i of that data center d at time t to be served, 0 otherwise
$\beta^{d}(t) =$ Normalized cost of a data center d at review point t
α, μ, γ: Weight parameters related to 3 optimization objectives
Variables:
$u^{vd}(t)$ = Binary decision variable to choose a data center d to satisfy a request from a virtual network v at review point t
$s^{vd}(t)$ = Bandwidth allocation going to data center d for virtual network v at time t
$y_{ij}^{vd}(t)$ = Bandwidth allocation for a request from VN v from an entry point i to a server j of data center d at review point t
$\widetilde{y}_{ij}^{vd}(t)$ = Binary decision variable to select a request to be satisfied from a VN v which comes to an entry point i and served by a server j of a data center d at review point t (this parallels $y_{ij}^{vd}(t)$)
$x_{ijp}^{vd}(t)$ = Bandwidth allocation in path p, if a request comes to an entry point i of a data center d is transferred to a server j uses path p at review point t
$z_{l}^{vd}(t)$ = Bandwidth needed on link l of a data center d for a VN v at time t
$e_{j}^{d}(t)$ = CPU processing capacity requirement from a server j of a data center d to satisfy the request coming from a VN customer at time t.
$g_{ij}^{vd}(t)$ = Server resource (CPU processing capacity) allocation for a VN v through an entry point i to a server j of data center d at review point t
$w_{jf}^{d}(t)$ = Binary decision variable to choose the optimum frequency f from the range of available frequencies of server j of data center d to meet the required demand of CPU processing capacity at review point t

Next, we introduce a binary shadow variable $\widetilde{y}_{ij}^{vd}(t)$ corresponding to $y_{ij}^{vd}(t)$ to track one-to-one mapping from a particular entry point i to a particular server j at any review point t by using a large positive number M and a small positive number ε:

$$y_{ij}^{vd}(t) \leq M\widetilde{y}_{ij}^{vd}(t), \quad j \in J_d, i \in I_d, v \in V, d \in D \quad (5)$$

$$y_{ij}^{vd}(t) \geq \varepsilon\widetilde{y}_{ij}^{vd}(t), \quad j \in J_d, i \in I_d, v \in V, d \in D \quad (6)$$

Here, (5) and (6) together addresses the requirement that $\widetilde{y}$ is 1 when the corresponding variable y has a positive flow; otherwise, $\widetilde{y}$ as 0 when y is 0.

The bandwidth that is allocated to a particular path from entry point i to server j of a particular data center d is given by using the path flow variables x_{ijp}^{vdj}:

$$\sum_{p \in P_{ij}^{vd}(t)} x_{ijp}^{vd}(t) = y_{ij}^{vd}(t), \quad j \in J_d, i \in I_d, v \in V, d \in D \quad (7)$$

If any bandwidth is allocated on a particular path p to satisfy a portion of the request of bandwidth demand h^v from any VN v, then all the links associated with that path p have to carry that portion of demand h^v. Therefore, we can determine the link flow on l for tuple $\langle v, d\rangle$:

$$\sum_{i \in I}\sum_{j \in J}\sum_{p \in P_{ij}^{vd}(t)} \delta_{ijpl}^{vd}(t) x_{ijp}^{vd}(t) = z_{l}^{vd}(t) \qquad d \in D, l \in L_d, v \in V \quad (8)$$

while the total amount of bandwidth required in one link l of a data center d to satisfy the requests of all VNs must not exceed the capacity of that link of this data center:

$$\sum_{v \in V} z_{l}^{vd}(t) \leq c_{l}^{d}(t), l \in L_d, d \in D \quad (9)$$

Next we address resource allocation of $r^v(t)$ to the appropriate tuple $\langle d, i, j\rangle$, ensuring this in accordance with shadow variable $\widetilde{y}$:

$$\sum_{d \in D}\sum_{i \in I_d}\sum_{j \in J_d} g_{ij}^{vd}(t) = r^{v}(t), \quad v \in V \quad (10)$$

$$g_{ij}^{vd}(t) \leq M\widetilde{y}_{ij}^{vd}(t), \quad j \in J_d, i \in I_d, v \in V, d \in D \quad (11)$$

$$g_{ij}^{vd}(t) \geq \varepsilon\widetilde{y}_{ij}^{vd}(t), \quad j \in J_d, i \in I_d, v \in V, d \in D \quad (12)$$

$$\sum_{v \in V}\sum_{i \in I_d} g_{ij}^{vd}(t) = e_{j}^{d}(t), j \in J_d, d \in D \quad (13)$$

In (13), $e_j^d(t)$ represents the total amount of resources required from a particular server j to satisfy the requests of all VNs that use the server coming through all entry points of a particular data center. The total resources required by a particular server must be less than or equal to the available resources of that particular server of a data center:

$$e_{j}^{d}(t) \leq \sum_{f \in F} a_{jf}^{d} w_{jf}^{d}(t), j \in J_d, d \in D \quad (14)$$

Finally, a particular server j running at a particular frequency f can produce a particular capacity a_{jf}. However, a server cannot run at more than one frequency at a time:

$$\sum_{f \in F} w_{jf}^{d}(t) \leq 1, j \in J_d, d \in D \quad (15)$$

For the goal of the optimization problem, we considered three cost components in the objective function: the network bandwidth cost, the server resource cost, and the data center location cost. These three sources of costs are assigned different weight parameters, α, μ, γ, to understand the influence of each term on the overall decision. Briefly, the goal is to accommodate as many requests as possible and this can be accomplished by minimizing the amount of resources used. Furthermore, since resources are of different types, we take an utility function-based approach by assigning weights to different components that form the objective function. That is, the objective function can be written as:

$$\min \alpha \sum_{d\in D}\sum_{v\in V}\sum_{l\in L_d} z_l^{vd}(t) + \mu \sum_{d\in D}\sum_{j\in J}\sum_{f\in F} b_{jf}^d w_{jf}^d(t) + \gamma \sum_{d\in D} \beta^d(t) u^{vd}(t) \quad (16)$$

To summarize, our unified formulation addresses decision choices at three different levels: data center, entry point, and then the destination server. Secondly, we take power consumption into account in determining the right frequency for operating a server. Finally, we consider three cost components in the composite objectives.

III. Simulation Study Setup and Result Analysis

To conduct our study, we chose the data center topology shown in Fig. 1. We set a maximum of two data centers ($N = 2$) to be selected. Each data center is considered to be identical in this study; each consisted of $I_d = 4$ entry points and $J_d = 16$ servers and all links inside the data center are set with the same capacity. We set $P_{ij}^{vd}(t) = 4$ paths from an entry point to a server among which only one path will be used for a specific request for the duration of this request. Parameter values used for the DCs are summarized in Table II.

We consider $V = 3$ virtual network customers that generate the requests. Recall that a request is represented by the tuple $\langle h, r\rangle$. We vary $\langle h, r\rangle$ for different simulation cases, while the arrival is generated randomly. Specifically, we assume that the request arrivals follow a Poisson process. We varied the arrival rate from 0.2 to 1.0 in increments of 0.2 for each VN customer. The service duration for the request arrivals is assumed to follow the negative exponential distribution with an average value of 5 time units measured in terms of the number of discrete review points.

To solve the optimization model at each review point t, we use an AMPL/CPLEX (v 12.6.0.0) tool environment. For the experiments we conducted, solving the MILP model using CPLEX took 1 second on an average at each review point. In most cases, the MILP problem was solved optimally. For the instances when it was not solved optimally, the highest optimality gap was found to be 2.73%. Through initial experimentation, we determined the weight factors for each term in the objective (16) and set them as $\alpha = 0.3, \mu = 0.05, \gamma = 8.1$ since we found these values to provide a proper balance among the three cost components, without any one term being more dominant than the other two terms.

Note that with an increase in the arrival load, the system may not have sufficient capacity to accommodate all requests. Thus, our simulation environment also records any request that was not satisfied by the system by tracking the blocked requests to determine the blocking rate. Through initial experimentation, we first determined the warm-up time for the simulation and then collected the data for a steady-state region after the warm-up time. For each arrival rate, we used 10 seeds and report the results on the average value. We also computed the confidence interval and found the 90% confidence interval to be approximately 5% in cost variation for low arrival rates to 2.5% for high arrival rates. Since our optimization model considers the power consumption factor, we use the power consumption and processing capacity of a particular server that runs at a specific frequency, as shown in Table III.

TABLE II
DC related parameters

Number of links in each DC	56
Capacity of each link	20
Number of nodes in each DC	36
Number of Entry points	4
Number of Servers	16
Normalized cost of using each DC, β^d	1

In Table IV, we summarize the four cases we studied. These studies reflect a number of systematic changes to understand the impact. First, we started with the case of all demands being homogeneous for VN customers, i.e., we set $\langle h, r\rangle = \langle 10, 1.65\rangle$ (Case-H). In the next case, we assigned the demand to be uniformly chosen at random from the discrete values in the range given by $\langle h, r\rangle = \langle [8, 12], [0.55, 2.75]\rangle$ where average $h = 10$ and the mid-point of $r = 1.65$; we refer to this case as Case-R. In the next case, we consider the variation between different VN customers while keeping the request to be same within each VN, i.e., $\langle h^1, r^1\rangle = \langle 8, 0.55\rangle$, $\langle h^2, r^2\rangle = \langle 10, 1.65\rangle$, $\langle h^3, r^3\rangle = \langle 12, 2.75\rangle$ (Case-VH). We can see that VN-1 requires the least resources for a request, while VN-3 requires the most resources. Finally, we consider the variation of the demand to be uniformly chosen at random within each VN from a range, i.e., $\langle h^1, r^1\rangle = \langle [7, 9], [0.35, 0.75]\rangle$, $\langle h^2, r^2\rangle = \langle [9, 11], [1.45, 1.85]\rangle$, $\langle h^3, r^3\rangle = \langle [11, 13], [2.55, 2.95]\rangle$ (Case-VR).

The choice of the above parameters in our study was motivated by the set of questions we posed in the Introduction. While we discuss a number of results using the above parameter values to answer these questions, we have two main postulates: *Postulate-1:* we postulate that when the bandwidth demand and the resources per request vary uniformly from an average value, the cost and the blocking would be higher compared to when the bandwidth demand and resources for each request is fixed; *Postulate-2:* we postulate that by taking three values for $\langle h, r\rangle$ in increasing order, the VN class with the lowest resource requirement would receive better treatment (lower blocking and cost) by the network than the other.

A. Cost and Blocking

Case-H is the baseline case where all services are homogenous. The cost and blocking are shown in Fig. 2 and Fig. 3. Not surprisingly, as the arrival rate increases, the cost of the network increases while the blocking also increases.

If we now consider the first variation Case-R from Case-H, where the average demand and resource requirements are the same as Case-H except that the value taken by each request is chosen uniformly from a range, we can see that the cost increase is similar between Case-H and Case-R, while Case-R has a higher cost at the lower arrival rates that changes at higher arrival rates. On the other hand, blocking for case-R is noticeably higher than that for Case-H for all arrival rates.

TABLE III
CPU FREQUENCIES, CAPACITIES AND OPERATIONAL COST [5]

Frequency Option	1	2	3	4	5	6	7	8
Frequency (GHz)	1.4	1.57	1.74	1.91	2.08	2.25	2.42	2.6
Normalized Capacity	.5385	.6038	.6692	.7346	.8	.8645	.9308	1
Power Consumption (watts)	60	63	66.8	71.3	76.8	83.2	90.7	100

TABLE IV
VALUES OF THE GENERAL PARAMETERS USED FOR THIS RESEARCH IN DIFFERENT CASES.

Cases	Parameters	Values
Case-H: Homogenous Bandwidth and CPU Processing Capacity for each request from all 3 VNs	Bandwidth Demand from VN-1, VN-2 and VN-3	10
	CPU Processing Capacity Demand from VN-1, VN-2 and VN-3	1.65
Case-R: (Bandwidth and CPU processing capacity demand is from a same range of value for each request for all VNs)	Bandwidth Demand	$unif\{8, 9, 10, 11, 12\}$
	CPU Processing Capacity Demand	$unif\{0.55, 1.10, 1.65, 2.20, 2.75\}$
Case-VH: Different Bandwidth and CPU Processing Capacity demand for different VNs while the demand is fixed within each VN	Bandwidth Demand-VN-1	8
	Bandwidth Demand-VN-2	10
	Bandwidth Demand-VN-3	12
	CPU Processing Capacity Demand-VN-1	0.55
	CPU Processing Capacity Demand-VN-2	1.65
	CPU Processing Capacity Demand-VN-3	2.75
Case-VR: Different Bandwidth and CPU Processing Capacity demand for different VNs while with random within a fixed range for each request from a particular VN	Bandwidth Demand-VN-1	$unif\{7, 8, 9\}$
	Bandwidth Demand-VN-2	$unif\{9, 10, 11\}$
	Bandwidth Demand-VN-3	$unif\{11, 12, 13\}$
	CPU Processing Capacity Demand-VN-1	$unif\{0.35, 0.55, 0.75\}$
	CPU Processing Capacity Demand-VN-2	$unif\{1.45, 1.65, 1.85\}$
	CPU Processing Capacity Demand-VN-3	$unif\{2.55, 2.75, 2.95\}$

Re-visiting Postulate-1, we can see that our result confirms Postulate-1 for blocking. On the other hand, in regard to cost, Postulate-1 does not hold for cost in a highly loaded environment when the blocking for Case-R is so high that the average number of requests admitted to the system is much less than that for Case-H, which in turn means that the cost incurred is lower. Certainly, this raises the question on why the cost of Case-R is higher compared to Case-H at a low arrival rate. This is because at a low arrival rate, the blocking is low: this allows requests from higher end of resources to be admitted, thus incuring higher cost. In terms of distribution of the cost components, the comparison between Case-H and Case-R is shown in Fig. 4; the pattern of the different cost components has a similar behavior like the total cost.

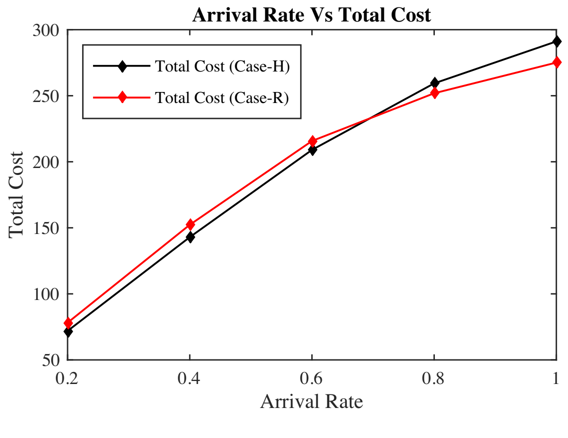

Fig. 2. Total Cost: Case-H vs. Case-R

Next we compare Case-VH vs. Case-VR. Note that in both these cases, the demand requirement for each VN class is different. The second difference is that for Case-VH, the demand and processing requirement for each VN class is kept the same, they are uniformly varied within the VN class in Case-VR. First, we discuss blocking. From Fig. 5, we can see that the average blocking for Case-VH is lower than that for Case-VR as the arrival rate increases. This is in line with what we observed comparing Case-R against Case-H. More notably, it is important to see how the blocking behavior changes from one VN class to another VN class. Recall that VN-1 requires the least resources per request while VN-3 requires the most

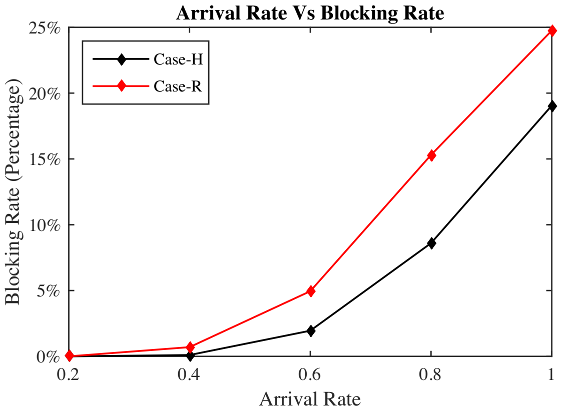

Fig. 3. Blocking: Case-H vs. Case-R

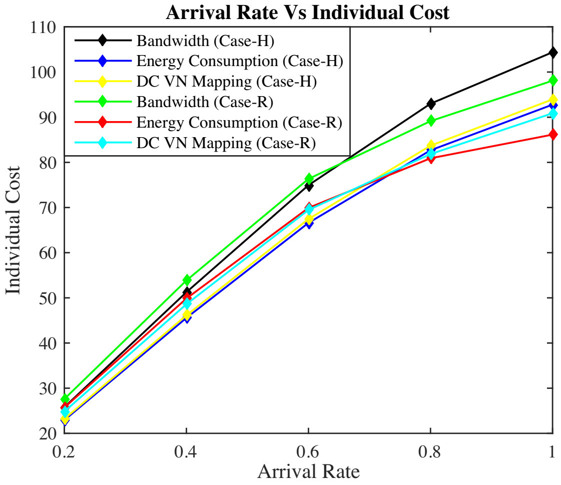

Fig. 4. Each VN Cost: Case-H vs. Case-R

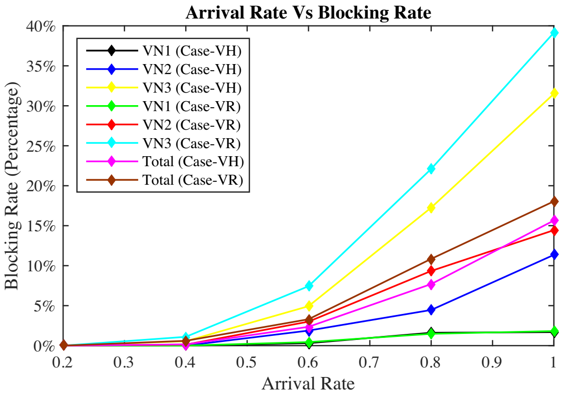

Fig. 5. Total and VN Blocking: Case-VH vs. Case-VR

resources. This is reflected when we observe the blocking for each VN class. For VN-1, the blocking is less than 2% at even the highest arrival rate (1.0), while for VN-2, the blocking is

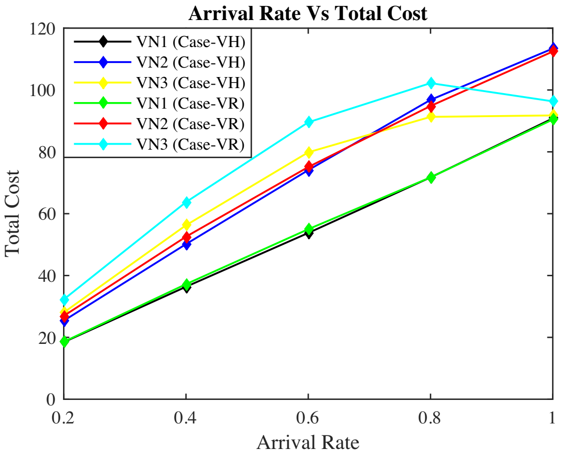

Fig. 6. Each VN cost: Case-VH vs. Case-VR

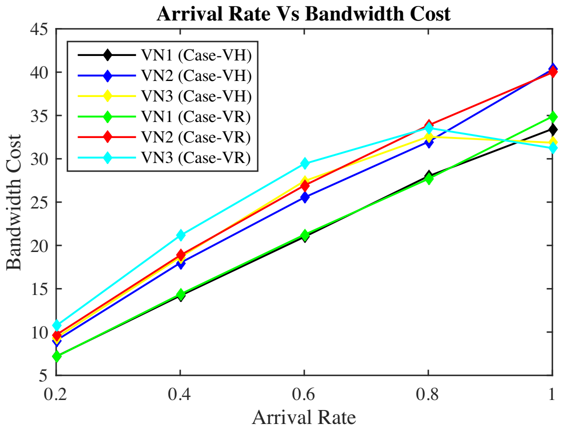

Fig. 7. Bandwidth Cost for each VN: Case-VH vs. Case-VR

around 12%, and it is significantly high at 32% for VN-3. In other words, in a congested situation, the network favors admitting requests that require less resources. We notice this difference starting from a low arrival rate of 0.4.

Now consider the variation from the resource requirement being fixed within each VN class against the same being uniformly random ("H" vs. "R"). We found that there is little difference in blocking for VN-1 between Case-VH and Case-VR. On the other hand, this difference is noticeable for VN-2, and quite prominent for VN-3. In other words, when the request is randomly distributed within a range with the VN class, this behavior is similar to what we noticed when we compared Case-R against Case-H. The main difference is that the observation is much more pronounced for VN-3 as this class requires significantly more resources.

Next, consider the cost of providing connectivity to each VN customer (Fig. 6). The cost of provisioning VN-1 is always the lowest regardless of the arrival rate. However, with VN-2 and VN-3, we notice that the provisioning cost is higher for VN-3 for lower arrival rates, but at a higher arrival rate,

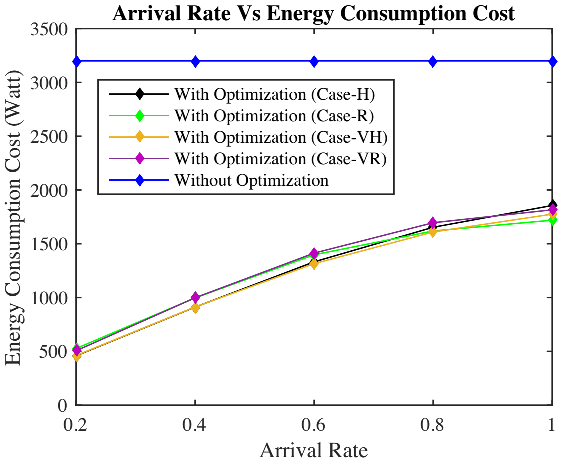

Fig. 8. Energy Cost for all Cases with Energy optimization and without

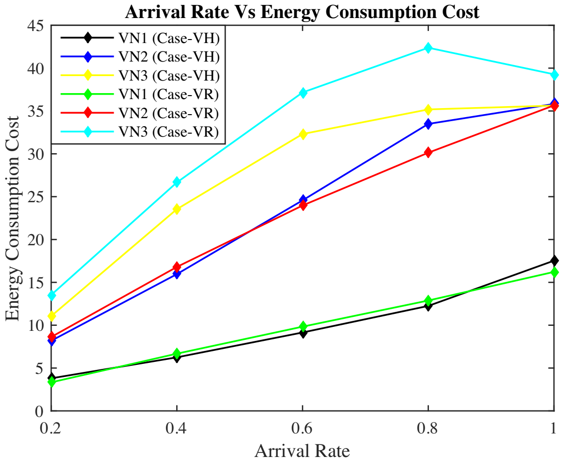

Fig. 9. Energy Cost Comparison: Case-VH vs. Case-VR

this is not so. This difference in cost can be explained by the observation that the blocking for VN-3 is significantly higher than that for VN-2 at a higher arrival rate to the point that the network is denying many VN-3 requests and in turn, the cost has also dropped. Revisiting Postulate-2, we find that this holds for blocking; however, for cost, Postulate-2 hold for a lower arrival rate, but not at the higher arrival rate. The basic reason is the same as the one explained with Postulate-1 for Case-H and Case-R. We also plot just the bandwidth cost for each VN when comparing these two cases in Fig. 7. We found the behavior to be similar except that for VN-3, the bandwidth cost is slightly higher for Case-VH then that for Case-VR at the highest arrival rate (heavy traffic).

B. Energy Consumption

We next focus on energy consumption. As we stated earlier, our model takes the energy issue into consideration. We first solved the optimization model using energy as the only cost in the objective and compared it again if the hosts were to continually run at the higher power consumption level. This is shown for all four cases in Fig. 8. We observe that our approach reduces the energy consumption to about one-sixth of the maximum energy cost at low arrival rate to two-thirds at the highest arrival rate. We also note that when the model is simply optimized for energy cost, the energy cost is not much different between the four cases.

Next we consider difference in the cost of energy consumption among VN classes by considering Case-VH and Case-VR (Fig. 9) when the entire objective function is optimized. We note that the randomness in resource requests around the average does not have much impact on VN-1 compared to if the resource request were fixed. On the other hand, for the highest VN class, VN-3, this variation in request makes a noticeably larger impact on the energy consumption cost. It may be noted that the energy cost drops off near the highest arrival rate. This is aligned with the cost phenomenon discussed with regard to Fig. 6.

IV. Related Work

Early research on data center networks investigated architectural construction, operation and scalability of DCs [6], [7], [8], [9], [10], [11]. Joint VM placement and routing for data center traffic engineering was addressed by [12]. Similar to [12], we also consider our problem from a traffic engineering point of view but we do not focus on VM placement; rather, we keep routing flexible in such a way that no dedicated server is required to satisfy demand from a particular VN. Any idle server is able to handle the request from any VN tenant. To satisfy a particular request, a server is chosen based on the resource demand and available resources of the server. Unlike their work, we take bandwidth guarantee into consideration. The issue of multiple service classes with heterogeneous requirements have been addressed for access control [13], [14]; however, they do not consider two-tuple demands nor the implication of network routing.

In [4], the authors presented a formulation to optimize the link cost in one data center, while we consider connecting multiple data centers. Unlike [4], we also take two issues into account, which are energy consumption by the servers, and the DC VN mapping cost. [5] discussed the servers' operational cost optimization without taking data center architecture into consideration. and they did not consider the on-demand model either. Furthermore, in our case, we combine three cost components (reducing link costs, power cost, and the DC VN mapping cost) together and impose weight parameters on each of these components to reflect their relative importance. Another novel contribution beyond the state-of-the-art of this research is the dynamic nature of our model to provide on-demand service considering north-south traffic and finding the optimal resource requirement to contain service blocking within a tolerable range. Moreover, we can also identify which servers are not used to serve the VN requests at a particular time, which can give us the opportunity to keep those servers in a lower power consumption mode.

V. Conclusion and Future Work

In this work, we presented a dynamic traffic engineering framework for resource allocation due to north-south traffic in a multi-location data center environment. We presented a novel MILP formulation that is solved in this framework at each review point. Our approach is geared for enterprise customers that require resource guarantees from data centers.

We then conducted a systematic study to understand the cost and blocking relation in normal traffic to overload traffic conditions by considering a number of cases. This sequence of considered cases allowed us to answer a number of questions when resource requirements may vary for each request as well as may differ between different customers. In general, we observed that VN customers with the lowest resource requirements face the lowest blocking as the traffic is increased in the system. For VN customers with high resource requirement, blocking is significantly higher for heavy traffic to the point where the cost incurred to serve this customer classes' accepted requests can be less than other customer classes.

There are several future directions we wish to address. In our current model, we do not factor in that a blocked request could incur a penalty cost due to loss in revenue. Secondly, we do not allow partial fulfillment of a request if there is not sufficient resources to fully consider a request. We also plan to consider large-scale network cases to mimic real world data center networks by considering a large number of virtual networks and with a large number of servers for data centers to test the scalability of our model and present the performance of the model with respect to optimality and solution time. Furthermore, we plan to add performance evaluation on the loads to a data center based on its geographical distance from different VNs. In addition, we intend to propose a heuristic and show the comparison between the solution achieved by using the heuristic and CPLEX form the point of view of computational complexity and performance such as different types of costs and blocking probability. These aspects will be addressed in a future work.

References

[1] "Amazon elastic computer cloud (Amazon EC2)." [Online]. Available: http://aws.amazon.com/ec2/

[2] D. F. Carr, "How Google works," *Baseline Magazine*, vol. 6, no. 6, 2006.

[3] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[4] M. A. Owens and D. Medhi, "Temporal bandwidth-intensive virtual network allocation optimization in a data center network," in *2013 IEEE International Conference on Communications (ICC)*. IEEE, 2013, pp. 3493–3497.

[5] H. Qian and D. Medhi, "Server operational cost optimization for cloud computing service providers over a time horizon," in *Proceedings of the 11th USENIX conference on Hot topics in management of internet, cloud, and enterprise networks and services*, 2011.

[6] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4. ACM, 2008, pp. 63–74.

[7] G. P. Alkmim, D. M. Batista, and N. L. da Fonseca, "Approximated algorithms for mapping virtual networks on network substrates," in *2012 IEEE International Conference on Communications (ICC)*. IEEE, 2012, pp. 1460–1465.

[8] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: a scalable and flexible data center network," in *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4. ACM, 2009, pp. 51–62.

[9] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "BCube: a high performance, server-centric network architecture for modular data centers," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 63–74, 2009.

[10] F. Hao, T. Lakshman, S. Mukherjee, and H. Song, "Enhancing dynamic cloud-based services using network virtualization," in *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*. ACM, 2009, pp. 37–44.

[11] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "PortLand: a scalable fault-tolerant layer 2 data center network fabric," in *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4. ACM, 2009, pp. 39–50.

[12] J. W. Jiang, T. Lan, S. Ha, M. Chen, and M. Chiang, "Joint VM placement and routing for data center traffic engineering," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2876–2880.

[13] B. Kraimeche and M. Schwartz, "Analysis of traffic access control strategies in integrated service networks," *IEEE Transactions on Communications*, vol. 33, no. 10, pp. 1085–1093, 1985.

[14] D. Medhi, A. van de Liefvoort, and C. Reece, "Performance analysis of a digital link with heterogeneous multislot traffic," *IEEE Transactions on Communications*, vol. 43, pp. 968–976, 1995.