

Impact of Ethernet Multipath Routing on Data Center Network Consolidations

Dallal Belabed, Stefano Secci, Guy Pujolle, Deep Medhi*

Sorbonne Universities, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France. Email: firstname.lastname@upmc.fr

* U. Missouri-Kansas City, 5100 Rockhill Road, Kansas City, MO 64110-2499 USA. Email: dmedhi@umkc.edu

Abstract—With the advent of network virtualization, data center networking is reaching a high level of management complexity. Indeed, interconnection networks in data center networks (DCN) are no longer just based on flat over-provisioned pipes, but are increasingly facing traffic engineering (TE) issues that commonly characterize long-haul provider networks. TE objectives, however, are opposite to energy efficiency (EE) objectives commonly chased by virtual machine (VM) consolidations. Moreover, the specific topologies of DCNs and the systematic use of multipath forwarding make the joint TE and VM consolidation optimization complex. The contribution of this paper is twofold. First, we propose a repeated matching heuristic for the DCN optimization problem with multipath capabilities, which also scales well for large topologies without discarding both TE and EE objectives. Second, we assess the impact of multipath forwarding on TE and EE goals. Extensive simulations show us that multipath forwarding is beneficial only when EE is not the primary goal in network-aware VM consolidations, and that it can be counterproductive when instead the EE is the primary goal of such optimizations.

I. INTRODUCTION

The emergence of cloud computing and network virtualization offers several advantages to both customers and data center (DC) providers, such as the opportunity for customers to improve service efficiency while minimizing capital expenditures, and for DC providers to increase the customer base without increasing facility size and power consumption. For the DC provider these improvements come, however, at the expense of higher DC network management complexity.

At present, interconnection links in data center networks (DCN) are no longer just based on flat over-provisioned pipes, but are increasingly facing TE issues that commonly characterize long-haul provider networks. With the growth of customer volumes, cabling density, storage consolidation traffic, and the increasing need of serving differentiated services and elastic demands, modern DC architectures are now facing important TE issues in terms of link and node utilization [1]. From a networking perspective, such rising TE requirements in DC networking can be met in two ways.

On the one hand, novel layer-2 multipath forwarding technologies allow to better balance the load on available links. Many Ethernet forwarding proprietary and standard solutions for DCNs exist. Among the standards, we can mention the Transparent Interconnection of Lots of Links (TRILL) [2], IEEE 802.1aq Shortest Path Bridging (SPB)[3] protocols and at some extent also OpenFlow [4]. The impact of Ethernet-level multipath forwarding protocols strictly depends on the

DCN topology. The legacy 3-layer [5] architecture is the most common DC architecture; however, it is loosing interest because with network virtualization the impact of inter-rack traffic, in support of consolidation procedures, is overcoming the amount of external traffic so that flat topologies become once more interesting. Topologies such as BCube [6], DCell [7] and fat-tree [8] are gaining momentum, indeed. The flat nature of these topologies can also give virtual bridging a higher importance in the DCN interconnect.

On the other hand, TE requirements can be met by adaptively migrating virtual machines (VMs), catalyst of significant traffic, to VM containers (virtualization servers) in the DCN that are topologically attractive. VM placement engines (e.g., VMware Capacity Planner, IBM CloudBurst) typically are aware of CPU, memory, storage and energy constraints of VM containers and are not, however, aware of network link states since the legacy hypothesis is to consider the DC fabric of unlimited network capacity. Such a hypothesis is now becoming inappropriate for DCNs for the above-mentioned reasons. Performing VM consolidation aware of both container and link states is, however, complex and known to be NP-hard [1]. The complexity does naturally increase when considering multipath capabilities. In this context, the contribution of this paper is twofold:

- We design a repeated matching heuristic, with multipath forwarding meeting TE and EE objectives, to allow scaling with large DCN sizes.
- We run our heuristic on realistic DCN instances, showing that multipath forwarding can be counterproductive when TE is not the primary goal of DCN optimizations, and that it grants only moderate gains when TE is the primary goal.

In the following, Section II presents the background of our work. Our heuristic in Section III, and simulation results in Section IV. Section V concludes the paper.

II. RELATED WORK

In recent years, a relevant amount of work has addressed the VM placement problem in DCNs. Only some works take into consideration network constraints in the VM placement problem.

Some of these studies exclude dynamic routing; for example in [9] the authors propose a VM placement solution considering network resource consumption, wherein the objective is set as the minimization of the average forwarding latency.

Others studies take only the network resources into account, as in [9] [10], or just consider the traffic volume to reduce the number of containers as in [11] where authors propose a VM placement considering a non-deterministic estimation of bandwidth demands, formulating the problem as a Stochastic Bin Packing problem, and introducing a new heuristic approach to resolve it. In [12], authors revisit the virtual embedding problem by distinguishing between server and switching nodes with respect to the common formulation; they do not handle the link capacity constraints, and they also do not consider multipath forwarding (load balancing), but a multipath routing with single egress path.

Only [13], the authors minimize the energy consumption of active servers, bridges and links, to maximize the global EE. The authors convert the VM placement problem into a routing problem, so as to address the joint network and server optimization problem (with there is no tread-off between network and server objectives).

To the best of our knowledge, our study is the first one addressing VM placement optimization considering the tread-off between network and server objectives under TE and EE constraints and considering multipath forwarding.

III. HEURISTIC APPROACH

Modeling the VM placement problem as an optimization problem we would have a NP-hard MCF formulation [14] [1] with specific variables and parameters to model VM mobility, multipathing forwarding, TE and EE objectives.

However, given the elasticity related to VM migrations and multipathing, requiring double mapping between VMs and VM containers, and between VM containers and usable paths, one would obtain a non-linear problem that cannot be linearized (single mapping could be linearized but not double mapping). Classically, mapping problems can be revisited as facility location problems, and when capacity constraints need to be verified as a function of the type of mapping, there are similarities with the capacitated facility location problem [15] and, in particular, with the single source facility location problem (SSFLP) [16], [17]. It is easy to derive that the DCN optimization problem can be reduced to the SSFLP and hence is NP-hard. Recently, modeling an optical network dimensioning problem as a facility location problem, authors in [18] extended a primitive repeated matching heuristic described in [16], [17] to solve the SSFLP and proved it can reach optimality gaps below 5% also for many instances of the problem. Motivated by those results, we redesigned the repeated matching heuristic to our DCN problem. Nevertheless, the double mapping we have to handle in our problem and the multiple capacity constraints to care about (at both link and server sides) make this a problem much more difficult to solve and for which comparison to the optimum is not possible. Given the outstanding performance of the repeated matching heuristic in [18], the approach is appropriate as we find confirmation later in the obtained results.

A. Problem formulation

Recall that DCN communications are between VMs, which can be hosted behind the same VM container or behind distant containers interconnected by a DCN path. Certainly, external communications can be modeled introducing fictitious VMs and VM containers acting as egress point from a functional standpoint. When multipath is enabled, multiple paths can be used. When communicating VMs are not colocated, inter-VM communication should involve a pair of containers and at least a DCN path between them.

Let a virtual node be designated by v , $v \in V$, and a VM container node pair be designated by cp , $cp \in T^C$, such that $cp(c^i, c^j)$, i.e., a container pair is composed of two containers c^i and c^j . When $c^i = c^j$ the container pair cp is said to be *recursive*. A subset of container node pairs is designated by D^C , so, $D^C \subseteq T^C$. Let the k^{th} path from RB r^1 to RB r^2 be designated by $rp(r^1, r^2, k)$. A set of RB paths is designated by D^R so that $D^R \subset T^R$.

- **Kit ϕ :** A Kit ϕ is composed of a subset of VMs D^V , a VM container pair $cp \in T^C$ and a subset of RB paths D^R . In a Kit ϕ , each VM $v \in D^V$ is assigned to one of the containers in a pair cp (c^1, c^2). A container pair cp (c^1, c^2) is connected by each RB path $rp(r^1, r^2, k) \in D^R$, such that c^1 and c^2 are respectively mapped to r^1 and r^2 . The Kit is recursive when its cp is recursive, and is such a case D^R must be empty. When multipath is not enabled, $|D^R| = 1$. The Kit is denoted by $\phi(cp, D^V, D^R)$.
- **Feasible Kit:** A Kit $\phi(cp, D^V, D^R)$ is said to be feasible if D^V is not empty, i.e., $D^V \neq \emptyset$, if memory and power demands of each VM are satisfied, restricted to D^V and cp , finally in case of non-recursive Kit, the link capacity constraints between VM containers are satisfied, restricted to D^V , D^R and cp .
- **L_1, L_2, L_3 , and L_4 :** L_1 is the set of VMs not matched with a container pair. L_2 is the set of VM container pairs not matched with a RB path. L_3 is the set of RB paths not matched with a container pair. L_4 is the set of Kits.
- **Packing Π :** A Packing is a union of Kits in L_4 . A Packing is said to be feasible if its Kits are feasible and L_1 is empty.

B. Matching Problem

The DCN optimization problem can be reformulated as a matching problem between these elements. The classical matching problem can be described as follows. Let A be a set of q elements h_1, h_2, \dots, h_q . A matching over A is such that each $h_i \in A$ can be matched with only one $h_j \in A$. An element can be matched with itself, which means that it remains unmatched. Let $s_{i,j}$ be the cost of matching h_i with h_j . We have $s_{i,j} = s_{j,i}$. We introduce the binary variable $z_{i,j}$ that is equal to 1 if h_i is matched with h_j and zero otherwise. The matching problem consists in finding the matching over A that minimizes the total cost of the matched pairs.

$$\begin{aligned}
& \min \sum_{i=1}^q \sum_{j=1}^q s_{i,j} z_{i,j} \quad (1) \\
& \text{s.t. } \sum_{j=1}^q z_{i,j} = 1, \quad i = 1, \dots, q \quad \sum_{i=1}^q z_{i,j} = 1, \quad j = 1, \dots, q \quad (2) \\
& z_{i,j} = z_{j,i}, \quad i, j = 1, \dots, q \quad z_{i,j} \in \{0, 1\}, \quad i, j = 1, \dots, q \quad (3)
\end{aligned}$$

In our heuristic, one matching problem is solved at each iteration between the elements of L_1, L_2, L_3 and L_4 . At each iteration, the number of matchable elements is $n_1 + n_2 + n_3 + n_4$ where n_1, n_2, n_3 and n_4 are the current cardinalities of the four sets, respectively. For each matching iteration, the costs $s_{i,j}$ have to be evaluated. The cost $s_{i,j}$ is the cost of the resulting element after having matched h_i with h_j , where $h_i, h_j \in \{L_1, L_2, L_3, L_4\}$ with element h_j . The costs $z_{i,j}$ are stored in a matrix Z . The dimension of the cost matrix Z is $(n_1 + n_2 + n_3 + n_4) \times (n_1 + n_2 + n_3 + n_4)$. Note that this dimension reduces at almost each iteration due to the matching. Z is a symmetric matrix. Given the symmetry, only ten blocks have to be considered. The notation $[L_i - L_j]$ is used hereafter to indicate the matching between the elements of L_i and the elements of L_j .

Selecting the least cost matching vector enables solution improvements via set transformations in next iterations. Obviously, $L_1 - L_1, L_2 - L_2$ and $L_3 - L_3$ matchings are ineffective. To avoid a matching, e.g., because infeasible, its cost is set to infinity. Matching corresponding to other blocks without L_4 lead to the formation of Kits. Other matchings involving elements of L_4 shall lead to improvement of current Kits, also generating local improvements due to the selection of better VM containers or RB routes; note that for these block local exchange, linear optimization problems are to be solved to determine an exchange of VMs, VM containers and Kits between the heuristic sets while satisfying computing capacity constraints.

The Kit cost function has to appropriately model two opposite forces the EE and the TE objective. The overall Kit cost is not meant to represent a direct monetary cost, but it is such that the repeated matching promotes less expensive and more efficient Kits. Therefore, to align with our objective, and remembering that the cost of a Packing corresponds to the cost of its Kits, we set the cost of a Kit $\phi(cp, D^V, D^R)$ as:

$$\mu(\phi) = (1 - \alpha)\mu^E(\phi) + \alpha\mu^{TE}(\phi) \quad (4)$$

where α is the trade-off scaling factor between the EE and the TE components, that are, respectively:

$$\mu^E(\phi) = \sum_{c^i \in cp} \left(\frac{K_{c^i}^P}{\sum_{v \in D_i^V} d_v^P} + \frac{K_{c^i}^M}{\sum_{v \in D_i^V} d_v^M} \right) \quad (5)$$

$$\mu^{TE}(\phi) = \max_{(n_i, n_j) \in rp, rp \in \phi} U_{n_i, n_j}(\Pi) \quad (6)$$

Note that computing capacity constraints are indirectly enforced within the $L_4 - L_4$ matching cost computation.

$U_{n_i, n_j}(\Pi)$ is the link utilization of each link used by the current packing Π solution, so that the maximum link utilization experienced by the Kit's RB paths can be minimized. In our heuristic, in order to linearly compute RB paths' link utilization, the aggregation and core links of RB paths are considered as congestion free, while access container-RB links are considered as prone to congestion, which generally adheres to the reality of most DCNs today as access links are typically 1 G Ethernet links while aggregation/core links reach the 10 Gbps and 40 Gbps rates. This is a realistic approximation acceptable in a heuristic approach, especially because it allows significant decrease in the heuristic's time complexity.

C. Steps of the repeated matching heuristic

Due to the advantage of repeated matching between the different sets as above described, we can get rid of much of the complexity of the problem. Its steps are as follows: the **step 0** of the algorithm starts with a degenerate Packing with no Kits and all other sets full. In **step 1** a series of Packings is formed. First in **step 1.1** the cost matrix Z is calculated for every block, in **step 1.2** the least cost matching vector is selected. and finally in **step 1.3** the algorithm go back to 1.1 for a new iteration unless the Packing cost has not changed in the last three iterations. In **step 2** the heuristic stops, and in the case L_1 is not empty a local incremental solution is created assigning VMs in L_1 to enabled and available VM containers or, if none, to new containers.

The least cost matching computation is Step 1.2 can be hard to solve optimally because of the symmetry constraint (3). In our heuristic, we decided to solve it in a suboptimal way to lower the time complexity. We have implemented the algorithm in [19], based on the method of Engquist [20]; its starting point is the solution vector of the matching problem without the symmetry constraint (3), obtained with the algorithm described in [21] that was chosen for its speed performance; its output is a symmetric solution matching vector.

Designing the matching costs in an efficient and rational way, the Packing cost across iterations should be decreasing, monotonically starting by the moment when L_1 gets empty; moreover, the Step 2 should be reached and the heuristic converges, and L_1 at the last step should be empty.

IV. SIMULATION RESULTS

We implemented our heuristic using Matlab, we used CPLEX for the computation of matching costs of some blocks. The adopted VM containers correspond to a Intel Xeon 5100 servers with 2 cores of 2.33GHz and 20GB RAM, able to host 16 VMs. We study the different forms of multipath. In fact our model encompasses the following cases.

- 1) Multipathing between RBs (MRB).
- 2) Multipathing between containers and RBs (MCRB).
- 3) Both Multipathing modes (MRB-MCRB).

We compare the three cases to the unipath case, under the following topologies: 3-layer, fat-tree, BCube and DCell.

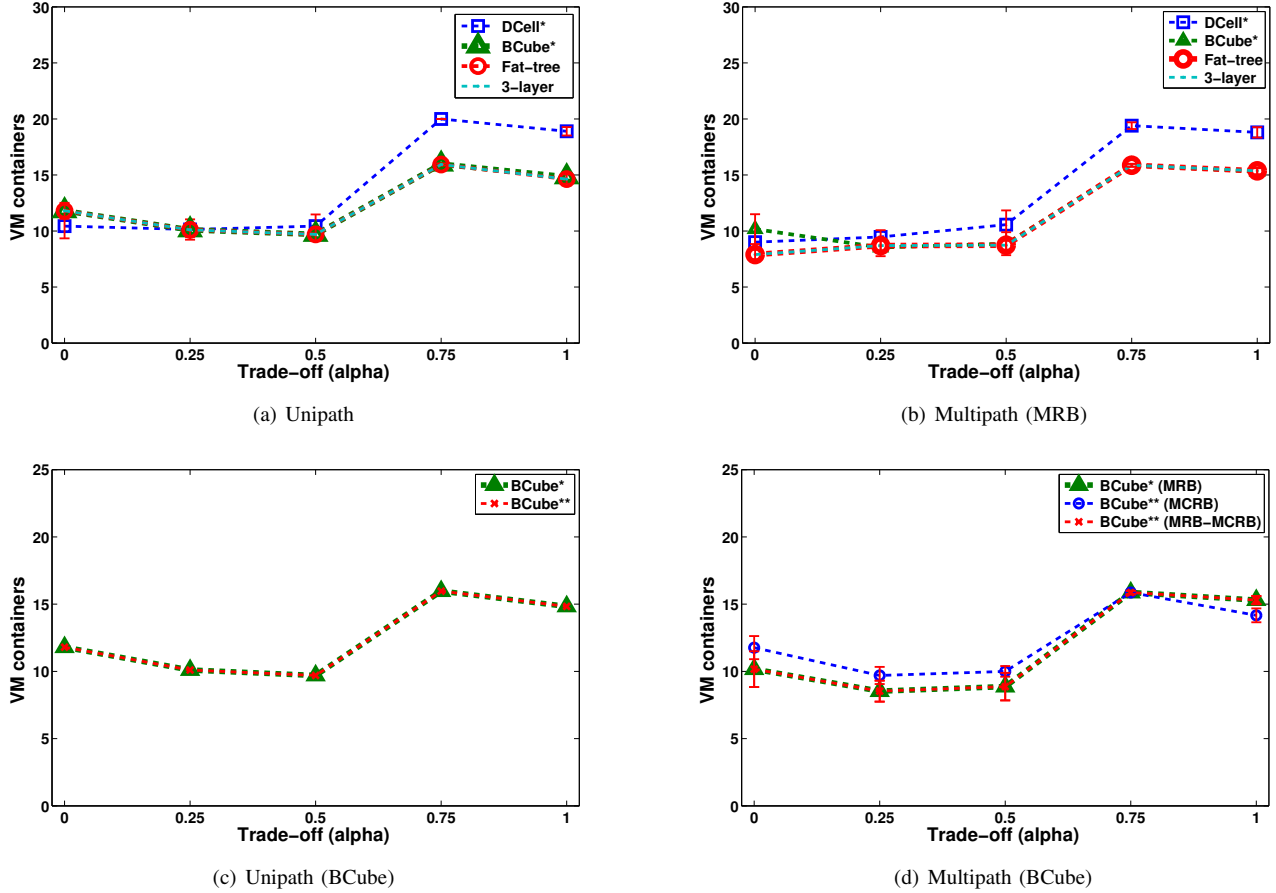


Fig. 2. Number of enabled VM containers.

We note that BCube and DCell have a server centric architecture, which means their servers have also a bridge role. Furthermore, direct links between bridges are missing. However, for the sake of comparison, we modify the conventional BCube and DCell architectures by adding links between bridges, while maintaining the flat nature of these topologies. For BCube and DCell, instead of connecting $BCube_0$ or $DCell_0$ containers with the higher level bridges, we connect $BCube_0$ or $DCell_0$ bridge with the higher level bridges. By doing this they can work without virtual bridge (marked as $BCube^*$ (Figs. 1a) and $DCell^*$ (Figs. 1c)).

For evident topological reasons with 3-layer, fat-tree and DCell topologies there is no multipath between containers and RBs. In fact there are no multiple links between containers and RBs, only BCube has that specificity. Finally, in order to allow multipath between containers and RBs, without virtual bridging, we add to the original BCube topology, a links between switch as added in $BCube^*$, we call this topology $BCube^{**}$ (Figs. 1b), where container multipath or both multipath mode can be enabled.

We simulate with 16 VM containers, which can be enabled, 20 VM containers for the DCell topology. All DCN are loaded at 85% in terms of computing and network capacity. Note that with all topologies we allowed for a certain level of

overbooking; the capacity of the access link is set to $1Gbps$.

We build a IaaS-like traffic matrix as in [9], with clusters of up to 30 VMs communicating with each-other and not communicating with other IaaS's VMs. Within each IaaS, the traffic matrix is built accordingly to the traffic distribution of [22]. We built 30 different instances with different traffic matrices. The results reported in the following are shown with an interval of confidence of 95%. Our heuristic is fast (reaches roughly a dozen of minutes per execution) and successfully reach a steady state (three iterations leading to the same solution, characterized by a feasible Packing).

1) *Energy efficiency considerations*: Figs. 2 illustrates the results in terms of enabled VM containers for different values of the trade-off parameter α , ranging from a null value giving full importance to EE goal, to a maximum value giving importance to the TE goal, with a step of 0.25. We report the results including the case when multipath is not enabled (i.e., $|D^R| = 1$ for all Kits) and the case where it is enabled. Observing the results we can assess that:

- as expected, when EE is discarded ($\alpha = 1$), the maximum number of VM containers is enabled at maximum;
- as expect, the curves of all topologies are similar for RB multipath. The DCell curve is slightly higher then the others curves, which is easily explained by the number

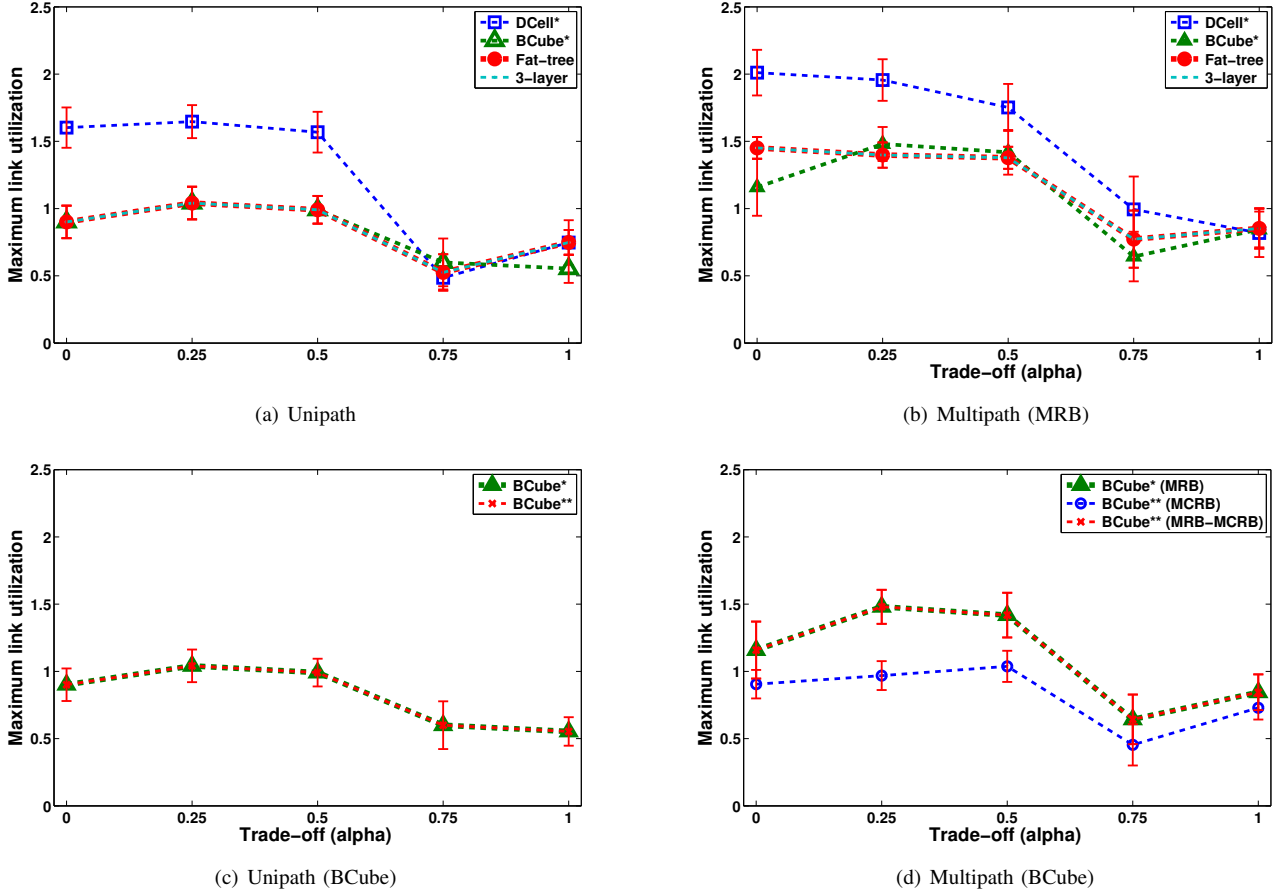


Fig. 3. Maximum link utilization

of container within DCell topology, which is equal to 20;

- the enabling of multipath routing decreases roughly by maximum 30% for MRB the number of enabled VM containers, and only 20% for MCRB when EE is considered as an important objective;
- the impact of multipath routing becomes negligible when EE is not considered important;
- MRB-MCRB gives the same effect as enabling MRB;

These results are not intuitive. On the one hand, decreasing the access link bottleneck by enabling multipath L2 routing seems to free VMs allowing a better VM containers consolidation and hence allowing switching off unused containers, leading to energy gains. On the other hand, multipath communications appears to be not useful for that goal, when switching off VM containers is either not interesting or not possible.

A. Traffic engineering considerations

As already mentioned, EE goals are expected to be opposite to TE goals. Fig. 3 reports the maximum link utilization for both the unipath and the multipath cases under different trade-off coefficients.

As expected, the curve decreases with α oppositely to the previous curve (fig. 2) that represents the number of enabled VM containers. Observing the results we can assess that:

- we remark a counter-intuitive aspect for MRB. The unipath case guarantees better TE performance when TE is not considered as an important goal in DCN optimization (i.e., when $\alpha \rightarrow 0$);
- MCRB gives the best result for TE goal regardless α value;
- MRB mode induces unacceptable TE performance when TE is not the primary goal;
- the curves of all topologies are really similar in case of unipath. We note that DCell topology has the worst curve when EE is the goal, and all curve converge when all the importance is given to the TE goal;
- MRB-MCRB gives the same effect as enabling MRB;

V. CONCLUSION

Data Center Networking is a challenging field of applications of old and new technologies and concepts. In this paper, we investigate how traffic engineering and EE goals in virtual machine consolidations can coexist with the emergence of Ethernet multipath routing protocols. We provide a repeated matching heuristic of the VM consolidation problem supporting multipath.

Through extensive simulation of realistic instances with legacy 3-layer and novel flat DC topologies, we discovered

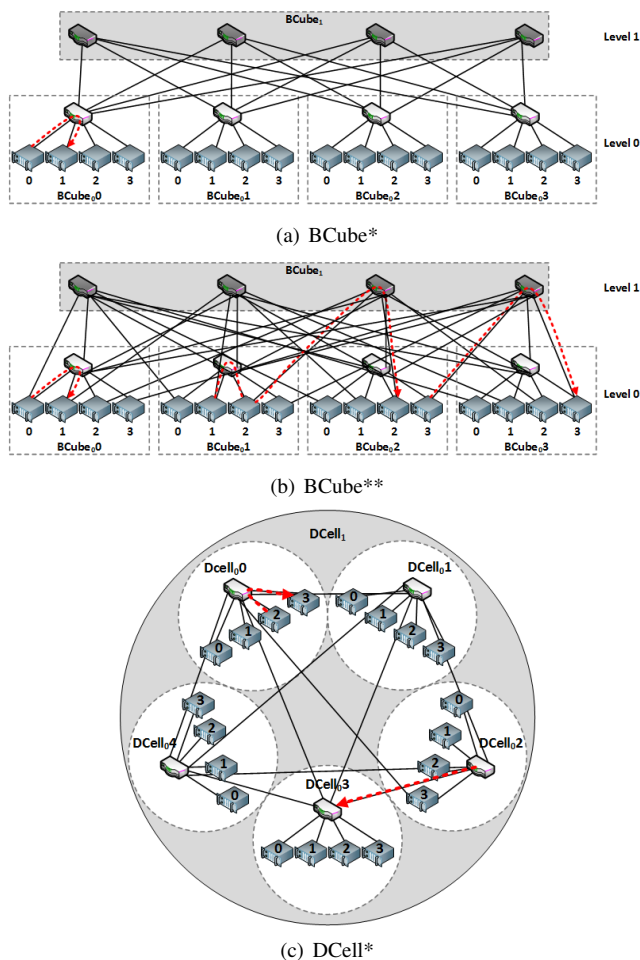


Fig. 1. BCube*, BCube** and DCell* topologies

that when TE is not the primary goal of DNC optimization, multipath routing can be counter-productive and can lead to saturation at some access links, while being able to decrease by roughly 40% the number of enabled VM containers. When TE is the primary goal instead, it grants only a moderate gain while significantly decreasing EE. This is an important result when considering the adoption of Ethernet multipath routing in DCs where access links are not over-provisioned and network-aware VM consolidations are performed.

ACKNOWLEDGMENT

This work was partially supported by the French “Investissement d’Avenir” NU@GE project, Fonds pour la Société Numérique (FSN) project and by the Systematic FUI 15 project RAVIR (<http://www.ravir.io>).

REFERENCES

- [1] Q. Zhang, L. Cheng, and R. Boutaba, “Cloud computing: state-of-the-art and research challenges,” *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.
- [2] J. Touch and R. Perlman, “Transparent interconnection of lots of links (TRILL): Problem and applicability statement,” *RFC 5556*, 2009.
- [3] M. Seaman, “IEEE 802.1aq shortest path bridging,” *IEEE std*, 2006.

- [4] N. McKeown and et al., “Openflow: enabling innovation in campus networks,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
- [5] “Data center architecture overview,” in *Cisco Data Center Infrastructure 2.5 Design Guide*. Cisco, 2011, pp. 7–16.
- [6] C. Guo and et al., “Bcube: a high performance, server-centric network architecture for modular data centers,” in *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4. ACM, 2009, pp. 63–74.
- [7] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, “Dcell: a scalable and fault-tolerant network structure for data centers,” in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4. ACM, 2008, pp. 75–86.
- [8] 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.
- [9] X. Meng, V. Pappas, and L. Zhang, “Improving the scalability of data center networks with traffic-aware virtual machine placement,” in *Proc. of IEEE INFOCOM 2010*, 2010, pp. 1–9.
- [10] O. Biran and et al., “A Stable Network-Aware VM Placement for Cloud Systems,” in *Proc. of IEEE/ACM CCGrid 2012*, 2012, pp. 498–506.
- [11] M. Wang, X. Meng, and L. Zhang, “Consolidating virtual machines with dynamic bandwidth demand in data centers,” in *Proc. of IEEE INFOCOM 2011*, 2011, pp. 71–75.
- [12] M. G. Rabbani, R. P. Esteves, M. Podlesny, G. Simon, L. Z. Granville, and R. Boutaba, “On tackling virtual data center embedding problem,” in *IM*, F. D. Turck, Y. Diao, C. S. Hong, D. Medhi, and R. Sadre, Eds. IEEE, 2013, pp. 177–184. [Online]. Available: <http://dblp.uni-trier.de/db/conf/im/im2013.html#RabbaniEPSGB13>
- [13] H. Jin, T. Cheo, D. Levy, A. Smith, D. Pan, J. Liu, and N. Pissinou, “Joint host-network optimization for energy-efficient data center networking,” *IEEE IPDPS, Boston, MA*, 2013.
- [14] M. Pióro and D. Medhi, *Routing, flow, and capacity design in communication and computer networks*. Elsevier/Morgan Kaufmann, 2004.
- [15] M. Balinski, “On finding integer solutions to linear programs,” in *Mathematica*, May 1964.
- [16] M. Rönnqvist, S. Tragantalerngsak, and J. Holt, “A repeated matching heuristic for the single-source capacitated facility location problem,” *European Journal of Operational Research*, vol. 116, pp. 51–68, 1999.
- [17] M. Rönnqvist, K. Holmberg, and D. Yuan, “An exact algorithm for the capacitated facility location problems with single sourcing,” *European Journal of Operational Research*, vol. 113, pp. 544–559, 1999.
- [18] A. Reinert, B. Sansò, and S. Secci, “Design optimization of the petaweb architecture,” *IEEE/ACM Trans. on Netw.*, vol. 17, no. 1, pp. 332–345.
- [19] M. Forbes, J. Holt, P. Kilby, and A. Watts, “A matching algorithm with application to bus operations,” *Australian Journal of Combinatorics*, vol. 4, pp. 71–85, 1991.
- [20] M. Engquist, “A successive shortest path algorithm for the assignment problem,” in *INFOR*, 1982, vol. 20, pp. 370–384.
- [21] R. Jonker and A. Volgenant, “A shortest augmenting path algorithm for dense and sparse linear assignment problems,” *Computing*, vol. 38, pp. 325–340, 1986.
- [22] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta, “V12: a scalable and flexible data center network,” *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 51–62, 2009.