Flow and Virtual Machine Placement in Wireless Cloud Data Centers

Heejun Roh[1], Kyunghwi Kim[1], Sangheon Pack[2], and Wonjun Lee[1(✉)]

[1] Network Research Laboratory, Korea University, Seoul 02841, Korea
wlee@korea.ac.kr
[2] Mobile Network and Communications Laboratory,
Korea University, Seoul 02841, Korea

Abstract. Virtualization for cloud computing has been driving data centers to contain massive and diverse applications in a distributed manner. However, since existing network architectures do not supply enough network capacity for virtual machine (VM) interconnections, enhancing the network capacity with augmented wireless links has recently attracted a lot of research interests. Especially, architectural design and link scheduling of wireless data center networks (WDCNs) are of their main interests. However, the potential of WDCNs is under-estimated, since existing research efforts do not reflect flexibility of VM placement. To this end, in this paper, we explore another feasibility of WDCNs to combine dynamic VM placement algorithms. We design a low-complexity flow placement algorithm considering augmented wireless links with interference constraints, and discuss a set of VM placement algorithms under the flow placement algorithm. Our extensive evaluation of the algorithms in WDCNs with 60 GHz wireless links shows that combination of the flow and VM placement algorithms achieves better performance.

Keywords: Data center · Routing algorithm · Virtual machine placement

1 Introduction

Recently, with the advance of virtualization technology, a lot of cloud data centers have been designed and constructed to support massive and diverse applications in a concurrent and distributed manner. The cloud data centers have several benefits from large economies of scale and scale-out based dynamic computing resource allocation on demand. With large demands of cloud-based services such as world-wide consumer applications and data-intensive tasks, applicability of cloud data centers are rapidly growing.

However, current architecture of cloud data centers are still requiring a better solution for network capacity. Typically, a data center holds a huge amount of servers (10 K to 100 K) and they are interconnected by a data center network (DCN). In conventional data centers, DCNs have tree-like topologies allowing oversubscription due to cost reduction and the ratio of oversubscription increases

J.-H. Lee and S. Pack (Eds.): QShine 2016, LNICST 199, pp. 138–148, 2017.
DOI: 10.1007/978-3-319-60717-7_14

rapidly as traffic moves to a root [8]. It implies that *network capacity of a DCN is highly constrained by its oversubscription, and the DCN may not support bandwidth demands.* Since there are applications which require interactive processing across thousands of machines in cloud data centers, and it is highly difficult to predict dynamics of traffic demands, a lot of research works for wired DCNs have been carried out, such as network architectures [2,8], virtual machine (VM) management [14], traffic measurements [4,12], and flow scheduling [3,7].

More recently, with the advent of Gigabit-capable wireless links, a radical but novel approach to increase the network capacity with the augmented wireless links has attracted many researchers. Especially, with promise of a 60 GHz wireless standard [1] to give its data rate up to 6.76 Gbps, and expectation of unit cost of the 60 GHz devices (less than $ 10) [9], several research proposals for wireless data center networks (WDCNs) with 60 GHz wireless links are suggested to resolve some issues such as architecture [9], link designs and measurements [9,16], and link scheduling [6].

However, utilizing 60 GHz wireless links into a WDCN can cause tight constraints on network topology. Current WDCN architectures [9,16] use highly directional horn antennas which require fixed topology of wireless links. Although electronically steerable array antennas may give more flexibility, beam training for 60 GHz wireless links which form quite narrow directional beams ($1°$ in the worst case) incurs significant delay ranged from 10 ms to 1 s in average [13]. Therefore, efficient optimization techniques for utilizing the wireless links is essential for better network performance.

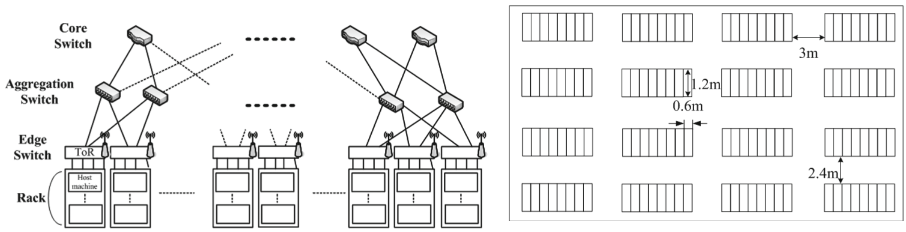
To this end, in this paper, we explore another feasibility of WDCNs by combining dynamic VM placement algorithms for cloud data centers. Public Infrastructure-as-a-Service (IaaS) cloud data centers, such as Amazon EC2 and Microsoft Azure, allow an application to dynamically utilize VM instances placed over a set of servers [10]. Therefore, using dynamic VM placement algorithms determine traffic patterns of WDCNs, and it is trivial that the dynamic VM placement gives another dimension of flexibility into WDCNs. However, due to highly dynamic traffic patterns of cloud applications, it is uncertain that which placement of VMs is preferable to enhance performance of WDCNs.

In this context, we first design a low-complexity flow placement algorithm considering augmented wireless links with interference constraints, and discuss a set of VM placement algorithms under the flow placement algorithm. In the flow placement, we exploit link utilization level to adaptively disperse traffic load across a WDCN. In the VM placement, to effectively exploit the extra capacity of wireless links in the WDCN, we propose a new metric for VM placement based on an in-depth study of applying wireless links to adapt to dynamic traffic patterns and provide better traffic locality. The new metric is designed to reflect influence of wireless links and applied to construct hierarchical VM clusters, each of which shares the traffic locality. We evaluate the placement algorithms by extensive simulations of wireless data center networks with 60 GHz wireless links, and the evaluation results clearly validate advantages of our approach.

The remainder of this paper is organized as follows. Section 2 explains our system model, and Sects. 3 and 4 present the details of the proposed routing and placement algorithms. Section 5 provides the performance evaluation of the proposed algorithms. Finally, we conclude this paper in Sect. 6.

2 System Model

In this section, we briefly describe our system model for data centers with wireless links before describing the proposed algorithms. Typically, data center network architectures consist of switches at multiple tiers, and there are multiple paths between a pair of host machines to be robust to congestion and link failures. Therefore, we adapt the common architecture for wired data center networking as shown in Fig. 1a. That is, the wired network fabric consists of wired links and three types of switches at each tier in the tree structure; edge, aggregation, and core switches.



(a) Network architecture (b) Data center layout

Fig. 1. Network architecture and layout of WDCNs

In this paper, especially in our simulation, we assume that racks are deployed as in Fig. 1b. A rack consists of tens of host machines connected to an edge switch with wired links, and the size of a rack is 0.6 m $\times$ 1.2 m. Racks are grouped into 4×4 rows, and each row contains 8 racks without gap. Rows are separated by 3 m and 2.4 m aisles. However, it should be note that the proposed algorithms can be applied into different data center network architectures and layouts.

Different from the common data center architectures, in our system model, each edge switch is equipped with one or more 60 GHz wireless devices, and they are fixed on top of racks (ToR) to achieve line of sight (LoS) communication, as in [9]. This assumption is a sufficient condition for stable 60 GHz communication in wireless data center networks. Since the signal strength of indoor LoS 60 GHz communications degrades rapidly with increasing distance, and its path loss model fits the Friis mode with exponent 2, performance of the wireless links is predictable and reliable without severe interference problem, which is suitable for data center networking. Of course, relaxing this assumption [16] can be a promising direction for our future work.

Table 1. Notations

Symbol	Definition	Symbol	Definition
$\boldsymbol{V}$	Virtual machine set	$\boldsymbol{M}$	Host machine set
$\mathbf{\Phi}$	Virtual machine placement	$\Phi(V_x) = M_i$	$V_x \in \boldsymbol{V}$ placed at $M_i \in \boldsymbol{M}$.
$\boldsymbol{L} = \boldsymbol{L^E} \cup L^W$	Whole link set	$\boldsymbol{F}$	Set of elephant flows
$\boldsymbol{\Omega}$	Path assignment set	$\boldsymbol{Q}$	Channel set
$\boldsymbol{L^E}$	Wired link set	$\boldsymbol{L^W}$	Wireless link set
$H^E_{i,j} \in \mathbf{H^E}$	Hop count between M_i and M_j using only $\boldsymbol{L^E}$	$H^W_{i,j} \in \mathbf{H^W}$	Hop count between M_i and M_j using $\boldsymbol{L}$
$T(l)$	Traffic rate sum at link l	$R(l)$	Link capacity of l
$U(l) = T(l)/R(l)$	Link utilization level of l	$\boldsymbol{I}(l)$	Interference link set of l
$\boldsymbol{P^E_{i,j}}(\in \boldsymbol{P^E})$	Path set between M_i and M_j (only wired links)	$\boldsymbol{P^W_{i,j}}(\in \boldsymbol{P^W})$	Path set between M_i and M_j (with wireless links)
δ	Distance metric	θ	Wireless link threshold

On the other hand, there are several views of virtualization to run multiple applications over host machines in cloud data centers. Amongst them, we focus on VM which explicitly consumes computing and storage resources of a host machine. A host machine can run a set of VMs which are owned by different applications, but the number of VMs in the host machine is constrained by its resource capacity. To simplify our discussion in VM management, in this paper, we treat each VM as a *slot* which consumes the same amount of computing and storage resources in a host machine. That is, host machines have the maximum number of slots, and we assume that the numbers are the same, which is widely used in the previous studies on traffic-aware VM placement such as [14].

To describe the proposed algorithms clearly, we introduce several notations which is summarized in Table 1. We also define the following terms:

- External machine ($M_0 \in \boldsymbol{M}$): There is a virtual host machine containing a VM ($V_0 \in \boldsymbol{V}$) beyond the gateways to reflect the external traffic of DCNs.
- Path sets ($\boldsymbol{P^E}$ and $\boldsymbol{P^W}$): Each host machine pair has two path sets; one with only wired (or Ethernet, for notational convenience) links ($\boldsymbol{P^E}$), and the other containing wireless links as well ($\boldsymbol{P^E}$). There are equal-cost (hop count) paths in each set, which are shortest paths between a pair of host machines. Note that we assume that multi-hop wireless communication is not taken into account to avoid the significant increase of complexity and overhead.
- Mice and elephant flows: Long-lived, high-throughput flows are called as elephant flows. These flows are distinguished by the host NIC bandwidth share [3] or the transferred traffic size [7]. We adopt the latter approach in this paper.

3 Flow-Based Routing Algorithm

Our routing algorithm exploits a central controller that gathers flow statistics and computes routes for elephant flows individually. In the proposed routing protocol, mice flows are transferred over Equal Cost Multi Path (ECMP) routing [11] without governing of the central controller, but elephant flows are delivered through a link load aware best path decision by the central controller. The proposed routing protocol allocates a channel for a wireless link based on the channel usage in its interference link set and estimates the achievable data rate based on the Signal-to-Interference-plus-Noise Ratio (SINR). The interference link set depends on the beamwidth of directional antenna. The protocol also presents threshold θ that adaptively allows a path set including wireless links.

The goal of our routing algorithm is to leverage the benefits of path diversity for providing better performance in terms of aggregation throughput and completion time of traffic demands, and the following constraints are considered:

$$(1)\ T(l) \leq R(l), \forall l \in \boldsymbol{L}, \qquad (2)\ l(q) \in \{0,1\}, \forall l \in \boldsymbol{L}^{\boldsymbol{W}},$$

$$(3)\ \sum_{q \in \boldsymbol{Q}} l(q) \leq 1, \forall l \in \boldsymbol{L}^{\boldsymbol{W}}, \qquad (4)\ l(q) + \sum_{l' \in \boldsymbol{I}(l)} l'(q) \leq 1, \forall l \in \boldsymbol{L}^{\boldsymbol{W}}.$$

$T(l)$ and $R(l)$ are the sum of traffic demands and the link capacity of l, respectively. The value of $R(l)$ is fixed for the wired links, while that is determined by SINR for the wireless ones. The sum of traffic demands at each link cannot exceed its capacity, and we use the constraint of (1). The other constraints (2)–(4) stem from the channel allocation of wireless links. The value of $l(q)$ becomes 1 when the channel q is allocated to link l, and otherwise 0 (2). Only one channel can be assigned to each link (3), and each channel can be used once at the same time for the links in the same interference link set (4).

Packets of a new flow are forwarded by hash-based path calculation (such as ECMP) at the beginning by default, and this avoids the flow setup of the control plane. If the flow becomes a large one, an elephant flow, over a given threshold size by any detection mechanisms, it is reported to a central controller to find the best available path. This path is computed for network load balance based on link utilization levels as shown in Algorithm 1. The proposed algorithm involves channel allocation and a threshold within the whole path set.

4 Virtual Machine Placement

In this section, we briefly explain our VM placement algorithms to improve the traffic locality by minimizing the network traffic at aggregate and core switches.

(1) New Communication Distance Metric Design

Wireless devices are added to provide *extra* capacity for data center networks and the use of wireless links is limited by channel allocation and SINR. Thus, our

Algorithm 1. Threshold-based Best Path Search

Require: θ, $\boldsymbol{\Phi}$, $\boldsymbol{F}$, $\boldsymbol{P^E}$, $\boldsymbol{P^W}$, and $\boldsymbol{\Omega}$

for $f \in \boldsymbol{F}$ **do**
  $\{M_u, M_v\} \leftarrow \{\Phi(V_x), \Phi(V_y)\}$ for f with $\{V_x, V_y\}$; p.best $\leftarrow \{\}$; p.load $\leftarrow \infty$
  for $p \in \boldsymbol{P}^{\boldsymbol{E}}_{u,v}$ **do**
    if $\max(p.U(l)) + f.\text{rate} < p.\text{load}$ **then**
      p.best $\leftarrow p$; p.load $\leftarrow \max(p.U(l)) + f.\text{rate}$
  if $p.\text{load} > \theta$ **then**
    for $p \in \boldsymbol{P}^{\boldsymbol{W}}_{u,v}$ **do**
      if $p.l^W = 0$ **then** ▷ If channel is not allocated,
        Check available channels; Check SINR
      if p is available **then**
        if $\max(p.U(l)) + f.\text{rate} < p.\text{load}$ **then**
          p.best $\leftarrow p$; p.load $\leftarrow \max(p.U(l)) + f.\text{rate}$
  f.path $\leftarrow p$.best; $p.U(l) \leftarrow p.U(l) + f.\text{rate}$ for $\forall l \in p$.best

routing algorithm utilizes the paths containing wireless links ($\boldsymbol{P^W}$) adaptively with regard to the path utilization level, and the impact of the paths in $\boldsymbol{P^W}$ is smaller than those in $\boldsymbol{P^E}$. In this context, we apply a weighting factor ($0 < \omega < 1$) for the hop distance of the paths in $\boldsymbol{P^W}$ for a new metric. The following metric is defined for each pair of machines (M_i and M_j) to leverage the features of network topology including wired and wireless links: $\delta_{i,j} = H^E_{i,j} - \sum_{k=1}^{K} \omega^k / H^{W_k}_{i,j}$.

In the above equation, $\boldsymbol{P^{W_k}}$ denotes the shortest path set including k wireless links, and $H^W_{i,j}$ is (i,j)-th element of the hop count matrix $\mathbf{H^W}$ for $\boldsymbol{P^W}$. We set $H^W_{i,j} = \infty$ when $\boldsymbol{P^W} = \emptyset$. We use positive constant K ($K > 1$) for the fine-grained clustering of VM placement. ω^k is used to penalize paths with many wireless links, since such paths may generate longer hop distance and they can interrupt the wireless links in the interference link sets.

(2) Hierarchical Slot Clustering

Since our system model assumes that no dynamic network topology, the communication distance between two host machines (and thus between two slots) are fixed. In [14], given that VMs' traffic information is known with this property, a generic VM placement problem is mathematically formulated with its NP-hardness and a slot clustering algorithm with pre-determined number of clusters is proposed as a part of approximate algorithm of the VM placement problem. However, determining the number in advance is difficult, especially in WDCNs.

To this end, we suggest hierarchical slot clustering algorithm. Our slot clustering algorithm generates hierarchical clusters based on the above communication distance metric. Our slot clustering algorithm tries to minimize the maximum value of the new metric in a bottom-up, greedy manner (from rack level to the root level). With the clusters, VMs can be assigned with better traffic locality, and we exploit them in the following placement and migration algorithms.

(3) Initial VM Placement

Estimating the traffic demands of new VMs to be placed is difficult and requires an explicit indication of applications, and studies in [7,15] have indicated its impracticality. Thus, we only use traffic statistics of the other VMs already placed in a data center. The algorithm finds available clusters to accommodate the new VMs from the clusters constructed by the proposed clustering algorithm. After that, the best one with the minimum traffic load is selected to place the new VMs.

(4) VM Migration

To minimize the network traffic at aggregate and core switches, we include a VM migration algorithm based on the traffic matrix at long-term scale. Our algorithm is based on Swap algorithm for the capacitated max k-uncut problem proposed in [5] to maximize the weight sum of the edges within partitions. The migration algorithm operates in a top-down manner (from the root level to the rack level). At each level, the cluster pair with the maximum communication cost pair is selected for the VM migration in a greedy manner. When swapping VMs between the selected cluster pair, VM pairs with higher communication cost gain are preferred for exchanging their slot positions.

5 Performance Evaluation

We use the data center layout illustrated in Fig. 1b, which describes the rack size and the aisle width. A row has 8 racks, and each edge switch is equipped with wireless devices. Each rack consists of 40 host machines connected to an edge switch with 1 Gbps wired links, thus the data center consist of over 5 K host machines. 10 Gbps wired links connect three types of switches, and the tree topology described in Fig. 1a is applied. For the wireless devices, we use a 25 dBi gain horn antenna with 3 dB beamwidth of 30°, and the transmission power is set to 10 mW. Data rates are determined based on [1], and the feasible data rate of each link is calculated with the physical interference mode using SINR. Each wireless device can use one channel of three 2.16 GHz channels at a time if available.

As described in [16], small obstacles (even antennas on ToRs) can produce multipath fading and degrade the signal strength due to the small wavelength of 60 GHz links (5 mm), and the achievable transmission rate is also deteriorated. Thus, 2D beamforming generates a large set of interference links with SINR degradation, and we apply 3D beamforming [16] in our simulation.

To setup flows and packets for our simulator, we generated job trace files by referring the analyzed data in [4,8,12]. Most flows show small size under 10 KB, and last a few hundreds of millisecond. The analyzed result of measured data in [8,12] illustrates that more than 85%, 90% and 99% of flows are less than 100 KB, 1 MB, and 100 MB, respectively. On the other hand, more than 90% of bytes are carried by the flows between 100 MB and 1 GB. External traffic is transmitted and received through gateways beyond the core switches.

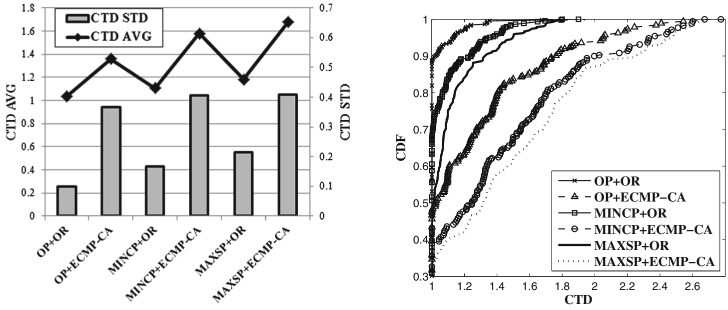
We present the following algorithms as comparison targets and implement in our simulator to compare with the proposed algorithms. Note that we denote **o**ur **p**lacement and **r**outing algorithms as **OP** and **OR** respectively.

- *MINimum Cost Placement with old metric* (**MINCP**): We apply the old metric considering hop distances of the shortest paths with wired links only. The algorithm calculates the communication cost by multiplying the metric and the traffic demand at large time scale. It constructs and divides clusters to minimize the communication cost for the placement and migration of VMs.
- *MAXimum Slot placement* (**MAXSP**): This algorithm utilizes as small number of racks as possible, satisfying the required number of virtual machines, to maximize the locality of virtual machines for each job.
- *ECMP with channel allocation* (**ECMP-CA**): We present a routing protocol to disperse the network traffic probabilistically. This algorithm finds a path for each flow randomly, and conducts a channel allocation when a path containing wireless links is chosen. If there is no available channel for the wireless links, the algorithm checks another path repeatedly, until the feasible path is chosen.

We exploit placement/routing sets combining the above algorithms and the proposed algorithms; OP + OR, OP + ECMP-CA, MINCP + OR, and MINCP + ECMP-CA, MAXSP + OR, MAXSP + ECMP-CA in the following simulation results. The proposed algorithms are compared with base algorithms in terms of the *completion time of demands* (**CTD**) and the *aggregate throughput* (AT). Note that in this paper, CTD is the normalized CTD (CTD/CTD_{ideal}) in other studies, where CTD_{ideal} is the CTD in an non-oversubscribed network [9].

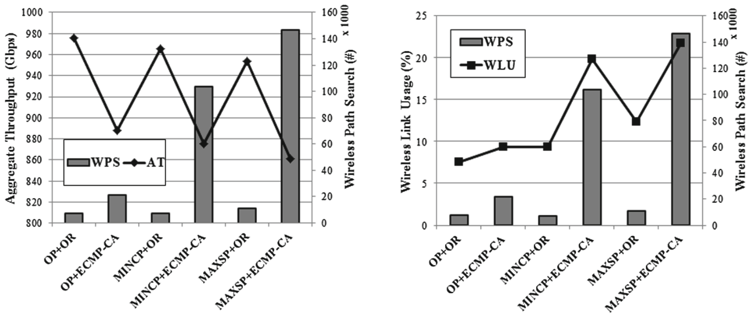
In this paper, the proposed routing protocol exploits the path sets including wireless links adaptively with threshold θ. Thus the threshold may have an effect on the performance of our routing protocol. We perform an extensive evaluation and choose the value of θ as 0.6 in the following simulation tests because OP + OR with the value provides the highest average aggregate throughput.

Figure 2 includes the simulation results. As illustrated in Fig. 2a, our placement and routing set (OP + OR) shows the lowest average (**AVG**) of CTD (1.02) and the lowest standard deviation (**STD**) of CTD (0.069). With our placement algorithm, our routing protocol reduces CTD AVG by 23.7% compared to ECMP-CA. MINCP can show lower CTD AVG and CTD STD with our routing protocol than ECMP-CA. OR decreases CTD AVG and CTD STD by 29.9% and 59.1% respectively with MINCP. We can also improve MAXSP by applying our routing protocol compared to exploiting ECMP-CA. Consequently we can infer that our routing protocol shows better CTD compared to ECMP-CA regardless of the placement algorithm. On the other hand, with our routing protocol, our placement algorithm decreases CTD AVG by 6.30% and 12.22% against MINCP and MAXSP respectively. With ECMP-CA, OP also reduces CTD AVG by 13.89% and 18.82% compared to MINCP and MAXSP respectively. In other words, our placement can improve CTD with not only OR but also ECMP-CA.



(a) CTD statistics

(b) CDF of CTD



(c) AT and WPS

(d) WPS and WLU

Fig. 2. OP + OR and comparison targets

Figure 2b shows the cumulative distribution function for each placement and routing pair. OP + OR increases most rapidly, while MAXSP + EMCP-CA is the slowest-growing. The maximum size of CTD AVG of OP + OR is 1.750, and that of MAXSP + EMCP-CA is 2.675. From Fig. 2a and c, there is a clear inverse correlation between the aggregate throughput (AT) and CTD. Therefore enhancing aggregate throughput by splitting traffic flows evenly and improving traffic locality is a critical factor determining network performance in data centers, which can reduce CTD. That is why OP + OR can complete traffic demands earlier than the others with the highest aggregate throughput.

Finally, we report the number of wireless path sets (WPS) as a metric for search space complexity and the wireless link usage ratio (WLU) as a metric for inefficient power consumption in Fig. 2c. In the figure, ECMP-CA retrieves the paths including wireless links more frequently. When either OR or ECMP-CA is applied, OP utilizes lower WLU but provides higher aggregate throughput. It is because that improving traffic locality with θ decreases the network load by reducing the distance between flow pairs. To this end, the routing protocols can mitigate the congestion earlier and forward more packets.

6 Conclusion

Wireless data center networks have challenges with regard to the dynamic topology, and it makes the system management more complicated, such as routing and virtual machine placement problems. In this paper we propose a routing algorithm including the threshold-based best path search algorithm and virtual machine placement algorithms to take the effect of wireless links into account. For better traffic locality, we present a new cost metric for the slot clustering of placement algorithms. The simulation results show that the protocol set including our placement algorithms and routing protocol provides the best performance in terms of CTD, WPS, WLU, and aggregation throughput.

Acknowledgments. This research was partly sponsored by NRF grant funded by the MSIP, Korea (No. 2013R1A2A2A01014000), and the MSIP, Korea, under the ITRC support program (IITP-2016-R0992-16-1012) supervised by the IITP.

References

1. The 802.11ad Standard. IEEE, December 2012
2. Al-Fares, M., Loukissas, A., Vahdat, A.: A scalable, commodity data center network architecture. In: Proceedings of ACM SIGCOMM. pp. 63–74, August 2008
3. Al-Fares, M., Radhakrishnan, S., Raghavan, B., Huang, N., Vahdat, A.: Hedera: dynamic flow scheduling for data center networks. In: Proceedings of NSDI, April 2010
4. Benson, T., Akella, A., Maltz, D.A.: Network traffic characteristics of data centers in the wild. In: Proceedings of ACM IMC, pp. 267–280, November 2010
5. Choudhury, S., Gaur, D.R., Krishnamurti, R.: An approximation algorithm for max k-uncut with capacity constraints. Optimization **61**(2), 143–150 (2012)
6. Cui, Y., Wang, H., Cheng, X.: Wireless link scheduling for data center networks. In: Proceedings of ICUIMC, February 2011
7. Curtis, A.R., Kim, W., Yalagandula, P.: Mahout: low-overhead datacenter traffic management using end-host-based elephant detection. In: Proceedings of IEEE INFOCOM, pp. 1629–1637, April 2011
8. Greenberg, A., Hamilton, J.R., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D.A., Patel, P., Sengupta, S.: VL2: a scalable and flexible data center network. In: Proceedings of ACM SIGCOMM, pp. 51–62, August 2009
9. Halperin, D., Kandula, S., Padhye, J., Bahl, P., Wetherall, D.: Augmenting data center networks with multi-Gigabit wireless links. In: Proceedings of ACM SIGCOMM, pp. 38–49, August 2011
10. He, K., Fisher, A., Wang, L., Gember, A., Akella, A., Ristenpart, T.: Next stop, the cloud: understanding modern web service deployment in EC2 and Azure. In: Proceedings of ACM IMC, October 2013
11. Hopps, C.: Analysis of an Equal-Cost Multi-Path Algorithm. RFC 2992, November 2000
12. Kandula, S., Sengupta, S., Greenberg, A., Patel, P., Chaiken, R.: The nature of data center traffic: measurements & analysis. In: Proceedings of ACM IMC, November 2009

13. Kim, J., Molisch, A.F.: Fast millimeter-wave beam training with receive beamforming. J. Commun. Netw. **16**(5), 512–522 (2014)
14. Meng, X., Pappas, V., Zhang, L.: Improving the scalability of data center networks with traffic-aware virtual machine placement. In: Proceedings of IEEE INFOCOM, March 2010
15. Wang, G., Andersen, D.G., Kaminsky, M., Papagiannaki, K., Ng, T.E., Kozuch, M., Ryan, M.: c-Through: part-time optics in data centers. In: Proceedings of ACM SIGCOMM, pp. 327–338, August 2010
16. Zhou, X., Zhang, Z., Zhu, Y., Li, Y., Kumar, S., Vahdat, A., Zhao, B.Y., Zheng, H.: Mirror mirror on the ceiling: flexible wireless links for data centers. In: Proceedings of ACM SIGCOMM, pp. 443–454, August 2012