# Report on Data Centers

Yongsen MA

## I. Literature Review

According to the architecture and topology in these papers, Data Center Networks can be divided into:

1) Static-electrical:
   a) Multi-layer tree:
      i) Fat-tree [1]
      ii) VL2 [2]
   b) Server-centric:
      i) DCell [3]
      ii) BCube [4]
   c) Random graph:
      i) Jellyfish [5]
   d) Multi-tenant:
      i) NetLord [6]
      ii) Oktopus [7]
      iii) FairCloud [8]
2) Flexible:
   a) Optical:
      i) c-Through [9]
      ii) Helios [10]
      iii) OSA [11]
   b) Wireless:
      i) Wireless flyways [12]
      ii) 3D beamforming [13]
      iii) Cylindrical racks [14]

These papers mainly discussed the issues of:

- Topology [1]–[5]
- Addressing [1], [2]
- Mapping [2], [15]
- Routing [3], [4]
- Scheduling [1], [12]
- Failure handling [15]–[17]

### A. Static-electrical Data Centers

The traditional data centers are generally designed in static connection by electrical switches. For the issues of topology, routing and scheduling in these data centers, there are numerous researches in this field including Fat-tree [1], VL2 [2], DCell [3], BCube [4], Jellyfish [5], etc. These work can be classified into multi-layer tree, server-centric and random graph according to the differences in topology graph, which also brings different advantages and disadvantages.

*1) Multi-layer Tree:* The topology of fat-tree is multi-rooted tree which can provide full aggregate bandwidth with low cost and resolved the bottleneck in inter-node communication. But the scale of fat-tree is limited that 48-port switches can only support $48 \cdot \frac{48}{2} \cdot \frac{48}{2} = 27648$ hosts at most.

The conventional data center architecture has limited server-to-server capacity that the over-subscription ratio increases rapidly in upper layer switches. Also, servers in multi-layer tree are location related that nearby servers are responsible for demand peaks or failures resulting in resources wasting. According to the measurement and analysis of traffic patterns, VL2 adopts Virtual Layer 2 Networking to realize the dynamic resource allocation across large server pools. It can assign any server to any service which can address the fluctuating demands of individual services.

The multi-rooted tree topology has poor fault-tolerant features which make it hard to detect and maintain failures. Furthermore, multi-rooted tree can not support one-to-all and all-to-all communications effectively.

*2) Server-centric:* **"DCell: A Scalable and Fault-Tolerant Network Structure for Data Centers", SIGCOMM 2008**

Large-scale, server-centric data centers

Topology: $n$ servers connected to a mini-switch within $DCell_0$, then $DCell_k$ is recursively generated by $t_{k-1}+1$ $DCell_{k-1}$s. The number of servers scales doubly exponentially as the node degree increases.

Addressing: a server in $DCell_k$ is denoted by $[a_k, uid_{k-1}]$, where $a_k$ is the $DCell_k$ this server belongs to and $uid_{k-1}$ is the unique ID of the server inside this $DCell_{k-1}$.

Routing: find the connection $(n_1, n_2)$ between $DCell$s, then find the paths of $src$ to $n_1$ and $n_2$ to $dst$. When failures occur, choosing *proxy* to make local-reroute or jump-up (rack failure).

Problems:

- the bottleneck link locates in the low-level links and traffics are not balanced
- increment expansion requires additional ports which brings extra overhead and complexity

**"BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers", SIGCOMM 2009**

Server-centric modular data centers

Topology: $BCube_0$ is $n$ servers connecting to an $n$-port switch. $BCube_k$ is constructed from $n$ $BCube_{k-1}$s and $n^k$ $n$-port switches.

Addressing: servers: $a_k a_{k-1} \cdots a_0$ ($a_i \in [0, n-1], i \in [0, k]$); switches: $< l, s_{k-1}, s_{k-2} \cdots s_0 >$, where $l (0 \leq l \leq k)$ is the level of the switch.

Routing:

*3) Random Graph:* **"Jellyfish: Networking Data Centers Randomly", NSDI 2012**

## B. Flexible Data Centers

*1) Optical:* **"OSA: An Optical Switching Architecture for Data Center Networks with Unprecedented Flexibility", NSDI 2012**

**"c-Through: Part-time Optics in Data Centers", SIGCOMM 2010**

1) optical circuit switching: high bandwidth but large delay (low bisection bandwidth)
2) convert routing to multiplex
3) separating the networks of packet-switch and circuit-switch increases the configuration complexity

4) buffer adds overhead
5) buffer at end hosts but not switches: requires hosts be programmable
6) weighted perfect matching problem: take traffic loads and capacity into consideration
7) each host runs a management daemon: adds overhead
8) artificially use an Ethernet switch to emulate the optical switch: accuracy

**"Helios: A Hybrid Electrical/Optical Switch Architecture for Modular Data Centers", SIGCOMM 2010**

the available bisection bandwidth is inflexible for predefined topology

1) unlike c-Through, Helios requires no modifications to end hosts but only at switches
2) without debouncing and EDC will improve throughput, but will reduce goodput due to link insertion and signal noise?
3) unidirectional circuits can better adapt to asymmetric traffic demands

*2) Wireless:* **"Mirror Mirror on the Ceiling: Flexible Wireless Links for Data Centers", SIGCOMM 2012**

Wireless Data Center is a new concept which should be explored further. Many issues such as interference, secluding, security, etc. should be standardized. On the other hand, the issues on cost, performance, energy-efficiency, reliability, etc. should be taken into consideration compared with electrical or optical data centers.

1) electrical
2) optical
3) wireless
   a) Link Blockage
   b) Radio Interference

Max concurrent links: Link conflicts (SINR); Greedy scheduling (graph coloring); Assigning radios.

Problems:

- Apart from the concurrent links, the efficient throughput should be explored. For instance, although the concurrent links are more with larger ceiling height $h$, it can decrease the throughput according to the curves of RSS (or Data Rate) vs. distance as shown in Figure 5.

**"Augmenting Data Center Networks with Multi-gigabit Wireless Links", SIGCOMM 2011**

"The base wired network is provisioned for the **average case** and can be oversubscribed. Each ToR switch is equipped withe one or more 60GHz wireless devices." "A central controller monitors DC **traffic patterns**, and switches the beams of the wireless devices to set up flyways between ToR switches that provide **added bandwidth** as needed." So it is ideal for **flexible and energy-efficient** data centers.

Problems:

- Conflict graph: for $N$ racks and $K$ antenna orientations, the input table is very **large** with the size of $(NK)^2$. On the other hand, the propagation conditions are similar, i.e., the table is **sparse**.

### *C. Other issues*

*1) Multi-tenant:* **"FairCloud: Sharing The Network In Cloud Computing", SIGCOMM 2012**

**"Towards Predictable Datacenter Networks", SIGCOMM 2011**

**"NetLord: A Scalable Multi-tenant Network Architecture for Virtualized Datacenters", SIGCOMM 2011**

*2) Failure Handling:* **"Generic and Automatic Address Configuration for Data Center Networks", SIGCOMM 2010**

Basic Procedures:

1) O2 Mapping
    a) Candidate selection via SPLD: **select** candidate with the same SPLD.
    b) Candidate filtering via orbit: **skip** candidate with the same orbit, then $Decomposition()$.
    c) Selective splitting $Refinement^*()$: **split** cells that really connect to the including cell.
2) Malfunction Detection
    a) Anchor pair selection:
    b) Malfunction detection:

Problems:

1) Initial selection of vertex $\nu \in \pi_p^i$
2) Whether it can be resolved by Compress Sensing?
    a) the topology graphs are sparse.
    b) only certain parts are changing in real-time operating (considering certain servers can be turned down for energy-efficiency and demand response).

**"NetPilot: Automating Datacenter Network Failure Mitigation", SIGCOMM 2012**

**"PortLand: A Scalable Fault-tolerant Layer 2 Data Center Network Fabric", SIGCOMM 2009**

## II. Research Issues

### A. Topology Control

1) topology graph is large and sparse
2) topology graph is changing due to errors, multiplexing and demand response
3) capacity differences in up- and down-link (directional graph)
4) capacity differences in connections (weighted graph)

Compress Sensing

### B. Demand Response

according to the traffic measurement and analysis in VL2 and wireless flyways, traffic patterns and loads are dynamic and unpredictable. So demand response can be done by Lyapunov optimization for it only needs the information of current and previous state.

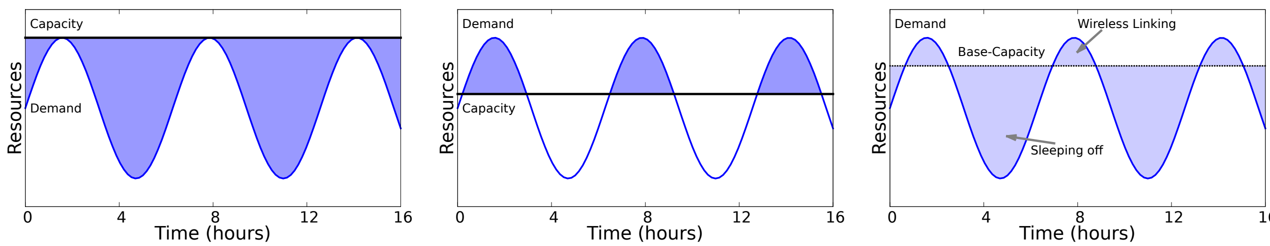
### C. Capacity Scheduling

### D. Resource Allocation

high efficiency: good trade-off between reliability and capacity

## III. Motivation

With the increasing demand of traffic loads and user requirements, it is a great challenge to make data centers agile and energy-efficient. A basic solution is to allow dynamic resource allocation based on flexible data center networks. The recent development of 60GHz wireless technology opens a door for the deployment of flexible data centers. It provides a good choice of adding capacity to data centers with under-provisioning capacity design. However, this will lead to new problems such as topology control and capacity scheduling, apart from the wireless propagation and link adaption in wireless networks. The PHY and MAC standardization of 60GHz networks is still ongoing (WirelessHD and IEEE 802.11ad/WiGig), its application in data centers should be further explored for the specific feature of topology and services.




(a) Provisioning for peak load (b) Fixed underprovisioning (c) Flexible underprovisioning

Fig. 1. Capacity provisioning based on traffic loads

Generally, the capacity of data centers is designed for peak loads, as shown in Figure 1a, but the resources will be wasted during non-peak times [18]. On the other hand, if the capacity is designed in under-provisioning case as shown in Figure 1b, the peak requirements can not be satisfied, leading to potential revenue sacrifice or over-occupied errors. To address these problems, we can employ flexible capacity and dynamic demand response, i.e., 60GHz wireless linking to add capacity for peak loads and sleeping mechanism to improve performance/cost efficiency, as shown in Figure 1c.

The wireless-flexible data centers are generally composed of:

1) Topology and Addressing:
   a) Conflict Graph: Propagation table (wireless), capacity matrix (wired/wireless) and traffic loads table (wired/wireless)
   b) Topology Graph: Physical-specific ID Addresses (PAs), Logical-specific IP Addresses (LAs) and Application-specific IP Addresses (AAs)
      i) PAs: providing distance information for 3D Beamforming
      ii) LAs: providing topology information for Capacity Adaption
      iii) AAs: providing traffic information for Demand Response
2) Routing and Scheduling:
   a) 3D Beamforming: adding additional capacity to neighbor ToRs according to PAs
   b) Demand Response: traffic estimation according to AAs
   c) Capacity Adaption: capacity scheduling according to LAs

## IV. Methodology

The problems of topology control, addressing, routing, and scheduling also exit in traditional data centers, but it brings new challenges when wireless links are introduced. For instance, the topology graph is changing due to wireless links and on-demand response. Also, the capacity matrix and conflict graph of wireless links are closely related to the relative locations of servers, e.g., capacity of 6G for neighbor servers and 1/2G for non-neighbor servers. Furthermore, the measurement and mapping of topology graph, conflict graph and traffic loads is challenging for large scale data centers.

Considering the scale of topology (conflict/topology graph and different addresses) and multi-layer scheduling (physical topology, capacity matrix and traffic patterns), we can employ **Compress Sensing**

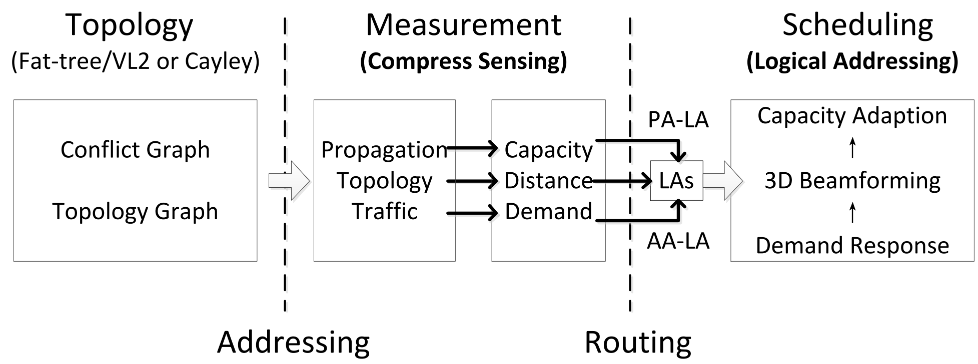



Fig. 2. Framework of flexible data centers based on 60GHz wireless networks

and **Logical Addressing** to address these problems.

**Topology:** multi-layer tree topology such as Fat-tree and VL2 (or cylindrical racks housing pie-shaped servers [14]) with ToRs.

**Addressing:** changing according to topology, PAs and AAs should be mapped to LAs to make 3D beamforming, demand response and capacity scheduling.

**Scheduling:** when the capacity matrix and traffic demands are mapped to LAs, we can determine the scheduling strategy including 3D beamforming and routing paths.

### A. *Compress Sensing*

The conflict graph can be classified into the topology graph, since it represents the wireless connections which provide additional 1/2/6G capacity for different ToRs. It is challenging to measure such a large scale topology graph (PAs and AAs) which usually requires all-to-all communications. Since both conflict graph [12] and topology graph [15] are sparse, we can utilize Compress Sensing to reduce overhead, just as the famous example that using only 3 out of 10 words can reconstruct the whole account password. In this way, we can use small number of samples to get the topology so that all-to-all communications are not necessary.

The requirements and reasons for Compress Sensing are:

1) topology graphs are **large**, and so comes the requirement to reduce measuring and mapping overhead
2) wireless propagation and traffic demand have **dynamic** furthers, so the measurement is challenging
3) topology graphs are **sparse**, which is the prior condition for compress sensing
4) topology graphs only have **local changes** (errors, on-off or wireless) in real-time operating

If the adjacency matrix of a given topology graph $G = (V, E)$ is $A^{N\times N}$, it can be represented by $N$ vertex $A_i$. Then the estimated matrix $\hat{y} \in \mathbf{R}^n$ can be calculated through the measure matrix $\Phi \in \mathbf{R}^{n\times N}(n \ll N)$ as follows:

$$y = \Phi A_i \tag{1}$$

If the measured data is recovered through $\hat{A}_i = f(y) = \Psi y = \Psi\Phi A_i$, the estimated results can be obtained by optimization

$$min \parallel A_i - f(y) \parallel_{l_x} \tag{2}$$

$$s.t. \quad \Phi A_i = y \tag{3}$$

where $l_x(x = 0, 1, 2)$ are the norms of a vector, among which $l_1$ is usually used representing the number of non-zero coordinates. Then $y$ can be easily measured and $A$ can be reconstructed through $y$ if the measure and reconstruct matrix ($\Phi$ and $\Psi$) are suitably designed. Therefore, all-to-all communications are not required which can help to reduce the measurement overhead. The topology graph can be estimated by Compress Sensing due to its large-scale and sparse feature. Furthermore, the propagation table and traffic loads can be measured in this way according to the measurement results in [12] and [2].

The topology that should be measured includes:

- Propagation table → Capacity matrix: the curve is quite certain for high frequency at near distance (do not consider PHY and MAC settings) [12], [13]
- PAs → LAs
- Traffic loads → Demands: the distribution of traffic loads has location differences across AAs and ToRs [2], [12]

Topology changes due to:

- Errors: nodes, links, miswiring
- wireless linking (links)
- Sleeping on-off (nodes)

### *B. Logical Addressing*

Wireless links make data centers flexible, i.e., using topology control and capacity scheduling to make resource allocation responding to traffic patterns and requirements. So the problem is how to allocate wireless links to deal with the problem of demand response when topology graphs are changing. The capacity matrix is composed of wired and wireless links (direct links between neighbor nodes or 3D beamforming for remote nodes), which can be obtained by PAs:

- wired 10G
- wireless direct 6G
- wireless indirect 1/2G

Then the PAs (conflict and topology graphs) and demands estimation (AAs) can be mapped to LAs:

- PA to LA for 3D Beamforming
- AA to LA for Traffic estimation

When the topology graph and traffic demands are mapped to LAs, the capacity scheduling can be made responding to dynamic traffic loads. First, we can get the whole capacity matrix according to propagation table and topology graph. Second, greedy choice of flyways can be made according to capacity matrix an traffic demands. Finally, we can make wireless linking by direct or indirect beamforming if necessary, and make routing and scheduling further.

## REFERENCES


[1] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. A scalable, commodity data center network architecture. ACM SIGCOMM '08, 2008.

[2] Albert Greenberg, James R. Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David A. Maltz, Parveen Patel, and Sudipta Sengupta. VL2: a scalable and flexible data center network. ACM SIGCOMM '09, 2009.

[3] Chuanxiong Guo, Haitao Wu, Kun Tan, Lei Shi, Yongguang Zhang, and Songwu Lu. DCell: a scalable and fault-tolerant network structure for data centers. ACM SIGCOMM '08, 2008.

[4] Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang, Yunfeng Shi, Chen Tian, Yongguang Zhang, and Songwu Lu. BCube: a high performance, server-centric network architecture for modular data centers. ACM SIGCOMM '09, 2009.

[5] A. Singla, C.Y. Hong, L. Popa, and P.B. Godfrey. Jellyfish: Networking data centers randomly. USENIX NSDI '12, 2012.

[6] Jayaram Mudigonda, Praveen Yalagandula, Jeff Mogul, Bryan Stiekes, and Yanick Pouffary. NetLord: a scalable multi-tenant network architecture for virtualized datacenters. ACM SIGCOMM '11, 2011.

[7] Hitesh Ballani, Paolo Costa, Thomas Karagiannis, and Ant Rowstron. Towards predictable datacenter networks. ACM SIGCOMM '11, 2011.

[8] Lucian Popa, Arvind Krishnamurthy, Sylvia Ratnasamy, and Ion Stoica. FairCloud: sharing the network in cloud computing. ACM SIGCOMM '12, 2012.

[9] Guohui Wang, David G. Andersen, Michael Kaminsky, Konstantina Papagiannaki, T.S. Eugene Ng, Michael Kozuch, and Michael Ryan. c-Through: part-time optics in data centers. ACM SIGCOMM '10, 2010.

[10] Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. Helios: a hybrid electrical/optical switch architecture for modular data centers. ACM SIGCOMM '10, 2010.

[11] K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen, and Y. Chen. OSA: An optical switching architecture for data center networks with unprecedented flexibility. USENIX NSDI '12, 2012.

[12] Daniel Halperin, Srikanth Kandula, Jitendra Padhye, Paramvir Bahl, and David Wetherall. Augmenting data center networks with multi-gigabit wireless links. ACM SIGCOMM '11, 2011.

[13] Xia Zhou, Zengbin Zhang, Yibo Zhu, Yubo Li, Saipriya Kumar, Amin Vahdat, Ben Y. Zhao, and Haitao Zheng. Mirror mirror on the ceiling: flexible wireless links for data centers. ACM SIGCOMM '12, 2012.

[14] Ji-Yong Shin, Emin Gün Sirer, Hakim Weatherspoon, and Darko Kirovski. On the feasibility of completely wireless datacenters. ACM ANCS '12, 2012.

[15] Kai Chen, Chuanxiong Guo, Haitao Wu, Jing Yuan, Zhenqian Feng, Yan Chen, Songwu Lu, and Wenfei Wu. Generic and automatic address configuration for data center networks. ACM SIGCOMM '10, 2010.

[16] Radhika Niranjan Mysore, Andreas Pamboris, Nathan Farrington, Nelson Huang, Pardis Miri, Sivasankar Radhakrishnan, Vikram Subramanya, and Amin Vahdat. PortLand: a scalable fault-tolerant layer 2 data center network fabric. ACM SIGCOMM '09, 2009.

[17] Xin Wu, Daniel Turner, Chao-Chih Chen, David A. Maltz, Xiaowei Yang, Lihua Yuan, and Ming Zhang. NetPilot: automating datacenter network failure mitigation. ACM SIGCOMM '12, 2012.

[18] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Commun. ACM*, 53(4):50–58, April 2010.