This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

Dynamic Scheduling for Wireless Data Center Networks

Yong Cui, *Member, IEEE,* Hongyi Wang, Xiuzhen Cheng, *Member, IEEE,* Dan Li, and Antti Ylä-Jääski

Abstract—Unbalanced traffic demands of different data center applications is an important issue in designing Data center networks (DCN). In this paper, we present our exploratory investigation on a hybrid DCN solution of utilizing wireless transmissions in DCNs. Our work aims to solve the congestion problem caused by a few hot nodes to improve the global performance. We model the wireless transmissions in DCN by considering both the wireless interference and the adaptive transmission rate. Besides, both throughput and job completion time are considered to measure the impact of wireless transmissions on the global performance. Based on the model, we formulate the problem of channel allocation as an optimization problem. We also design an approximation algorithm with an approximation bound of $1/2$ and a genetic algorithm to address the scheduling problem. A series of simulations are performed to evaluate and demonstrate the effectiveness of our wireless DCN scheme.

Index Terms—Data Center Networks, wireless communication, dynamic scheduling, evolutionary computing and genetic algorithms

1 Introduction

With the development of cloud computing, more and more data centers are built to provide various cloud applications such as search, e-mail, and distributed file systems. As the infrastructure of data centers, data center networks (DCN) are constructed to provide a scalable structure and an adequate network capacity to bear the services.

However, current DCN, which evolves from Enterprise LAN networks, comes across more and more difficulties with the growth of cloud computing. To begin with, the rapidly increasing size of data centers brings new challenges. Nowadays, large-scale data centers usually consist of thousands of servers. For traditional Ethernet solutions, expensive high-end switches and a huge number of wires are necessary to support so many servers, which leads to a lot of troubles in wiring and maintaining.

On the other hand, data center applications with unbalanced traffic distributions suffer from inadequate network capacity. Based on the traffic statistics obtained from a real-world data center, typical data center applications such as map-reduce [1] usually generate a traffic demand with only a few nodes being hot (i.e., these nodes need to transmit a large volume of traffic). Figure 1 shows an example of traffic demand matrix, where darker points stand for higher traffic demands. Although the matrix is quite sparse, those hot nodes are likely to cause loss on edge links and therefore put off the completion of a job. Furthermore, the non-deterministic distribution of hot nodes makes it impossible to set up additional wired links in advance to relieve the congestion. As a result, the network is prone to suffer from hot spot congestion, which cannot be efficiently relived by deterministic wired topology [2], [3], [4].

To solve these problems, we propose to utilize wireless transmissions in DCNs. Compared with wired connections, wireless links have advantages in several aspects. First, they are free of wiring and thus the maintenance is much more convenient. Second, direct links between servers are easy to achieve with wireless techniques inside a data center, which can avoid the extra cost in multi-hop transmissions. Third, dynamic wireless connections can be set up on-demand. Therefore, it is possible to adjust the topology dynamically to provide more network capacity for hot nodes.

Moreover, the state-of-art high-frequency wireless techniques [5] make it possible for wireless transmissions to provide as enough bandwidth as Ethernet. In particular, IEEE 802 has been working on the standards of the communications at 60GHz, named with IEEE 802.11ad. Prototype devices have also been produced. In brief, wireless transmissions offer an available solution for the non-deterministic unbalanced traffic distribution of data center applications.

Nevertheless, due to limited channel resources and their susceptibility to interference, it's hard to build large-scale data center networks with wireless alone. Based on these observations mentioned above, we propose a hybrid DCN solution in which wireless networks work as a supplementary of Ethernet structure to ad-

- *Y. Cui is with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, P.R.China. E-mail: cuiyong@tsinghua.edu.cn*
- *H. Wang is with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, P.R.China. E-mail: wanghongyi09@mails.tsinghua.edu.cn*
- *X. Cheng is with The George Washington University, Washington, DC 20052, USA. E-mail: cheng@gwu.edu*
- *D. Li is with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, P.R.China. E-mail: lidan@csnet1.cs.tsinghua.edu.cn*
- *A. Ylä-Jääski is with the Department of Computer Science and Engineering, Aalto University, FIN-02015 HUT, Finland. E-mail: antti.yla-jaaski@aalto.fi*

Digital Object Indentifier 10.1109/TPDS.2013.5

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

dress the congestion problem. By shifting the workload of hot nodes with wireless communications, the performance of the hybrid DCN can be significantly improved.

There are some challenges in the design of hybrid DCNs. To begin with, wireless links should be arranged appropriately to improve the performance. A lot of factors are involved in the wireless scheduling. For example, wireless links should be set up to solve the congestion of hot servers. Channels should be allocated to avoid interference between wireless links.

Besides, the scheduling of wireless communications should coordinate with the ethernet transmissions. In other words, the performance of wireless networks (typically measured by throughput) and the global job completion time should be jointly considered.

In this paper, we focus on the scheduling problem of wireless transmissions in the hybrid DCN. To the best of our knowledge, this is the first work that gives detailed technical approach of wireless DCN. Our contribution mainly lies in the following aspects. First, we perform a novel problem formulation for wireless DCN. A realistic interference formalization and the adaptive transmission rate are considered in the model. In addition, we also pay attention to the coordination of the throughput of wireless networks and the global job completion time. Second, we design an approximation algorithm to tackle the channel allocation problem and prove that the lower bound of the algorithm performance is $1/2$. Furthermore, we propose another genetic-algorithm-based approach to provide high efficiency in handling evolving traffic demands. Elaborate simulations are conducted to evaluate algorithms and demonstrate the performance enhancement resulted from exploiting wireless transmissions in DCNs.

The rest of the paper is organized as follows. Section 2 discusses the most related work. Our system modeling is elaborated in Section 3. Section 4 describes the centralized scheduling mechanism. The details of the our scheduling algorithms are depicted in Section 5 and Section 6. Simulation methods and results are presented and analyzed in Section 7. Section 8 concludes the paper.

2 Related Work

There has been a lot of research on the interconnection architectures and the routing mechanisms of DCNs.

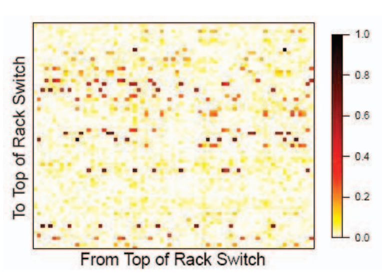


Fig. 1. Matrix of application demands between top of rack switches [6]

Some of them extend existing tree-based topologies to improve scalability and throughput. Fat-tree [7] groups servers into pods and establishes multiple paths between the core layer and the aggregation layer of a typical tree-based data center architecture. Based on the fat-tree topology, Portland [8] is proposed to support various requirements of data center applications such as virtual machine migration. VL2 [9] is based on Clos Networks, in which new addressing and routing mechanisms are designed to provide high capacity and performance isolation between different services.

Moreover, researchers also try to develop new topologies rather than extend existing architectures. DCell [10] takes a structure composed of one switch and k servers as a basic unit and constructs high level topologies recursively by connecting basic units together with direct links between servers. FiConn [11] is an extension of DCell but it only utilizes the backup port of each server rather than add new NICs. BCube [12] introduces more switches to improve the bottleneck problem of DCell and develops a modularized data center solution. It achieves load balancing and a graceful performance degradation under various faulty conditions.

Besides the schemes based on Ethernet, work has also been done to make use of other transmission media. Ramachandran *et al.* propose to employ 60GHz communications in DCNs [13]. This work designs a clean-slate wireless-based DCN architecture and presents a lot of relevant challenges. However, it does not provide detailed technical approaches. Flyway [6] is the first one that combines wireless networks with existent Ethernet-based DCNs. Yet, it only performs an initial problem formulation and many important factors, such as interference, are not considered in the scheduling mechanism. Our previous works [14] [15] investigate the scheduling problem of wireless DCN by considering both the limit of radios and the wireless interference. However, only a node-exclusive interference model is considered in the scheduling, which is far from a real-world interference pattern. Another work [16] addresses these problems with a physical interference model and a genetic scheduling algorithm. In this paper, we further extend this work.

In addition to wireless, there are some literatures that propose to utilize optical circuit switches for high-speed direct communications between racks [17], [18], [19]. In general, all the work on optical-based DCN exploits scheduling mechanisms that adapts optical circuit to traffic demands to maximize the throughput. This idea is similar to the scheduling schemes of wireless DCN.

Channel allocation is a well-known NP-hard problem [20], and is usually handled by heuristic algorithms [21]. Queue theory is used to analyze and design the schedule policies for maximum throughput in multi hop radio networks [22]. Approaches based on genetic algorithm (GA) have also been proposed to handle the channel allocation problem in various wireless networks. Zomaya *et al.* [23] highlight the potential of using GA to deal with wireless resource allocation and design a GA method with an improved mutation operator to address the problem efficiently. Patra *et al.* [24] improve the algorithm by introducing a new pluck operator. Ding *et al.* [25] utilize GA to assign partially overlapping channels in WMN. Our approach is different from the existing ones in that the channel allocation problem in a wireless DCN is different from those in conventional wireless networks as mentioned in Section 1. Moreover, we design our own crossover and mutation operators to improve the performance of GA.

3 System Model

3.1 Wireless Transmission

In this paper, we aim to propose a wireless DCN approach such that the adoption of wireless transmissions should be independent of the implementation of a DCN. Therefore, the basic unit of a wireless DCN should not be restricted to be a server or a rack. Instead, we formalize it as an abstract concept with the following definition.

Definition 1: A *wireless transmission unit* (*WTU*) refers to a group of servers that use the same set of antennas to transmit flows to other servers outside the group.

Typically, a rack is taken as a unit. For solutions that does not adopt traditional tree-based topologies, we can map the notion of wireless transmission unit to a specific structure in the corresponding solution. For example, the BCube0 structure in BCube is an ideal candidate for a wireless transmission unit.

Based on Definition 1, we divide the traffic in the network into two categories: one is the inter-unit traffic and the other is the intra-unit traffic. Note that wireless links are employed for transmitting the inter-unit traffic. Assume v_1 and v_2 are two units. Let $t(v_1, v_2)$ denote the traffic demand from v_1 to v_2. The distribution of inter-unit traffic can be illustrated with a wireless transmission graph.

Definition 2: A *wireless transmission graph* is a directed graph $G = (V, E)$, where V denotes the set of units and E denotes the set of flows.

Each node v in the graph corresponds to a physical unit with antennas. We use $\omega(v)$ to denote the number of antennas belonging to v. An edge $e = (v_1, v_2)$ presents in the graph if and only if there is a flow from v_1 to v_2.

3.2 Channel Allocation and Interference

For a given wireless transmission graph, channels are assigned to the edges to bear the wireless transmissions. In this paper, we assume that only orthogonal channels are employed, i.e., transmissions on different channels would not interfere with each other. Let C denote the set of orthogonal channels. A channel allocation scheme can be expressed with a binary matrix S, in which rows correspond to the set of flows E and columns correspond to the set of channels C. If a channel c is assigned to flow e, the value of $S_{e,c}$ is 1; otherwise, $S_{e,c}$ is 0.

If all the elements in a row of e are 0, it indicates that no channel is allocated for e and we define e as an *idle edge*. Otherwise, there is a channel assigned to e and we define e as an *active edge*. Let E_a denote the set of active edges. Note that a link can occupy at most one channel. In other words, there is at most one non-zero element in each row of S.

One important issue in channel allocation is the possible interference between wireless transmissions. We employ the following definition to formulate the interference relationship.

Definition 3: The *conflict edge* of e in a wireless transmission graph is the edge whose transmission causes interference on e.

The determination of a conflict edge involves the physical position of the transmitting nodes and the channel assigned to the flows. With respect to physical position, we adopt the interference range model, in which a sender causes interference on the nodes inside its interference range. Note that our model does not rely on certain antenna techniques. The interference range of the nodes with omni-directional antennas is usually defined as a unit disk while that of directional antennas depends on the relative position of two endpoints and the corresponding patterns. Whichever antenna technique is utilized, we can just adopt the corresponding interference range model to find the conflict edge set of each edge.

Data transmissions in DCN should be reliable so acknowledgement is required. We transmit data packets and acknowledgement packets at reversed edges. Thus, the transmission on an edge $e = (v_1, v_2)$ is unidirectional, i.e. packets are only sent from v_1 to v_2. Based on the interference range of a node, we can induce the interference range of an edge: An edge $e = (v_1, v_2)$ is in the transmission range of edge $\bar{e} = (\bar{v}_1, \bar{v}_2)$ if v_2 is in the transmission range of $\bar{v}_1$.

If e is in the transmission range of $\bar{e}$, we consider $\bar{e}$ as the *potential conflict edge* of e. If $\bar{e}$ is the potential conflict edge of e and there exists a channel c such than $S_{e,c} = S_{\bar{e},c} = 1$, then $\bar{e}$ is the conflict edge of e. Let $\Gamma(e)$ denote the conflict edge set of e and $\Gamma_0(e)$ be the potential conflict edge set. Since all the nodes are static, potential conflict edge sets can be precomputed for a given wireless transmission graph.

3.3 SINR and Data Rate

In the research on wireless networks, the protocol interference model and the physical interference model are often used to formulate the interference. In the protocol interference model, the transmission of an edge is blocked if one of its conflict edges is active. On the other hand, simultaneous transmissions are admitted in the physical interference model as long as the signal to interference and noise ratio (SINR) at the receiver is larger than a threshold. We adopt the later model in this work. Therefore, the transmission on e can successfully

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

performed if and only if:

$$\mathrm{SINR}(e) = \frac{P_S(e)}{\sum_{\bar{e}\in\Gamma(e)} P_I(\bar{e}) + N_0} \geq T_{\mathrm{SINR}} \tag{1}$$

where $P_S(e)$ denotes the signal power received by v_2, N_0 is the environment noise, and $P_I(\bar{e})$ denotes the interference power caused by $\bar{e}$ and received by v_2. For a given edge e, the edges in $\Gamma(e)$ may cause interference of different intensity on e. We define the intensity of interference as follows.

Definition 4: If $\bar{e}$ is in the potential conflict edge of e, the *interference factor* between $\bar{e}$ and e is the ratio between the power input to the transmitting antenna of $\bar{e}$ and the power received by the receiving antenna of e on the same channel.

The interference factor can be computed according to Friis transmission equation as shown in (2), where $\frac{P_r}{P_t}$ is the ratio of the power received by the receiving antenna P_r and power input to the transmitting antenna P_t; G_t and G_r are the antenna gains of the transmitting and receiving antennas, respectively; λ is the wavelength and R is the distance; and the exponent α is typically in the range of 2 to 5 as an estimation to the pass-loss effect.

$$\frac{P_r}{P_t} = G_r G_t \left(\frac{\lambda}{4\pi R}\right)^{\alpha} \tag{2}$$

For simplicity, we assume that all the antennas have the same gain. If $\bar{e} = (\bar{v}_1, \bar{v}_2)$ is the conflict edge of e, the power of interference caused by $\bar{e}$ is expressed with (3), where $R(e, \bar{e})$ denotes $R(\bar{v}_1, v_2)$.

$$P_I(\bar{e}, e) = \frac{G_t G_r \lambda^{\alpha}}{(4\pi)^{\alpha}} \frac{P_t(\bar{e})}{R(e, \bar{e})^{\alpha}} \tag{3}$$

Let $C_I = (G_t G_r \lambda^{\alpha})/(4\pi)^{\alpha}$. The interference factor between $\bar{e}$ and e can be expressed with (4).

$$I(\bar{e}, e) = \frac{C_I}{R(e, \bar{e})^{\alpha}} \tag{4}$$

Similar to the computation of the interference factor, the signal power received by v_2 can also be computed with the Friis equation. In short, the SINR of e can be expressed with (5), where $R(e)$ denotes $R(v_1, v_2)$.

$$\mathrm{SINR}(e) = \frac{C_I P_t(e)/R(e)^k}{\sum_{\bar{e}\in\Gamma(e)} I(\bar{e}, e) P_t(\bar{e}) + N_0} \tag{5}$$

SINR is not only the necessary condition of a successful transmission but also the factor that has an impact on the data rate of wireless links. For example in 802.11, coding and modulation are selected based on SINR and thus lead to different data rates. This mechanism is based on Shannon theorem, as given in (6), where $Capacity$ is the upper bound of the data rate and B is the channel bandwidth.

$$Capacity = B \log_2(1 + \mathrm{SINR}) \tag{6}$$

We assume that the data rate of e is proportional to its capacity and that all the channels have the same bandwidth B. The data rate can be calculated with (7), where β is the ratio between capacity and data rate.

$$r(e) = \begin{cases} \beta B \log_2(1 + \mathrm{SINR}(e)) & \text{if } \mathrm{SINR}(e) \geq T_{\mathrm{SINR}} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

4 Scheduling Mechanism

Based on the aforementioned modeling of wireless DCN, we propose a centralized scheduling mechanism for wireless transmissions. When acquiring information, we adopt a popular management model of OpenFlow [26] for our central controller to gather flow statistics from all the servers in a way similar to [27], [28], [29]. Openflow can help achieve effective centralized scheduling for large data centers with low overhead as shown in [27], [29].

By implementing the OpenFlow protocol, switches in DCN can both get statistics of flows and open a secure channel to the central controller. So injection of a new flow or completion of a transmission can be detected and signalled to the central controller, and up-to-date traffic distributions can also be estimated and sent to the central controller. Then a periodical or trigger-based schedule mechanism can be used to tackle the congestion problem. As we mainly focus on the distribution of hot nodes, we can trigger scheduling when a new hot node is observed (i.e. the aggregate flow rate of a node grows beyond a specified threshold). We call this mechanism *hot-node-based triggering*. Periodical scheduling can help balance the workload in DCN at a regular interval while hot-node-based triggering can adapt to the dynamism of the workload more effectively.

The scheduling mainly consists of two parts: the first step is to construct a wireless transmission graph based on the traffic information; and the second one is to perform channel allocation in the wireless transmission graph. We provide the details of the two steps in this section.

4.1 Constructing a Wireless Transmission Graph

Selecting Transmissions to Construct the Edge Set: At the beginning of each scheduling procedure, the central controller converts the traffic demands to a digraph for latter scheduling. Efforts should be made to reduce the size of the channel allocation problem as there are thousands of concurrent flows in DCN. First, we select flows belonging to those hot nodes with a high priority. It is the high traffic of sparse hot nodes that causes congestion and puts off the completion of a job. Therefore, limited wireless channel resources should be used to serve these hot spots to solve the congestion problem. Second, flows whose distance between the source and the destination is beyond the valid range of the wireless transmissions (about 10m for 60GHz communications) are removed. It is impossible for the wireless antennas to perform the transmission on these long edges, so we don't need to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

consider the interference issue for them. Note that the nodes on these long edges will remain in the graph if they are involved in other short edges and the potential interference on these nodes will still be considered by setting up the conflict edge sets for those short edges.

Some flows are precluded from scheduling while others should be considered. For the requirement of continuity, the on-going wireless transmissions should not be interrupted so the corresponding edges should be included in the graph.

Assigning Weight to the Edges The total throughput of the network is usually taken as the metric of performance in scheduling schemes. However, it is not the case for our problem. As discussed above, nodes with a higher volume of traffic usually finish their transmissions later due to the limit of bandwidth and consequently, putting off the global job completion time. Another fact is that some flows are expected to experience much longer delay than others via Ethernet because of the static topology and the routing mechanisms. Under these conditions, it is obvious that setting up wireless links for some flows is more profitable than doing this for others.

We formalize this property as the *utility* of the flow, which reflects the contribution to the global performance made by transmitting the traffic via wireless links. In a wireless transmission graph, each edge e is associated with a weight $u(e)$ that denotes the utility of the corresponding flow.

In this work, we employ network delay to estimate the utility of a transmission. Intuitively, a flow with a high network delay, caused by either congestion or a long wired transmission path, is preferred in assigning channel resources. Therefore, we define that the utility is directly proportional to the network delay as (8), where $d(e)$ is the network delay of e and μ is a positive coefficient. Note that utility is non-dimensional.

$$u(e) = \mu d(e) \tag{8}$$

As long as each edge of the wireless transmission graph is assigned with a weight, our approach can be applied to generate a channel allocation scheme.

4.2 Channel Allocation Problem

After constructing the wireless transmission graph, the controller needs to find a proper channel allocation scheme. We formalize channel allocation as an optimization problem whose variable is the allocation scheme. In the optimization objective, we employ the weighted throughput, defined in Definition 5, to estimate the impact of a wireless transmission on the global performance.

Definition 5: The weighted throughput of a transmission is the product of its throughput and its utility.

Let E_o denote the set of on-going wireless transmissions and $c_o(e)$ denote the channel assigned to the on-going transmission e. Let $E_s(v)$ denote the set of edges whose source point is v, and $E_d(v)$ be the set of edges whose destination point is v. Based on the above analysis, we derive the optimization problem of channel allocation as (9). In (9), the first constraint provides the valid values for the elements in S; the second one indicates that a transmission can occupy no more than 1 channel; the third one ensures that the number of active edges belonging to a node do not exceed the number of its antennas; the last two constraints keep the on-going transmissions active.

$$\max \sum_{e\in E} u(e)r(e) \tag{9}$$

s.t.

$$\begin{aligned}
&S_{e,c} \in \{0,1\} && \forall e \in E, \forall c \in C\\
&\sum_{c\in C} S_{e,c} \le 1 && \forall e \in E\\
&|\{e | e \in (E_s(v) \cup E_d(v)) \cap E_a\}| \le \omega(v) && \forall v \in V\\
&S_{e,c_o(e)} = 1 && \forall e \in E_o\\
&\mathrm{SINR}(e) \ge T_{\mathrm{SINR}} && \forall e \in E_o
\end{aligned}$$

Let $f(S) = \max \sum_{e\in E} u(e)r(e)$. We can further convert (9) to (10) by expressing the constraints with S. It is clear that all the constraints are linear.

$$\max f(S) \tag{10}$$

s.t.

$$\begin{aligned}
&S_{e,c} \in \{0,1\} && \forall e \in E, \forall c \in C\\
&\sum_{c\in C} S_{e,c} \le 1 && \forall e \in E\\
&\sum_{c\in C} \sum_{e\in E_s(v)\cup E_d(v)} S_{e,c} \le \omega(v) && \forall v \in V\\
&S_{e,c_o(e)} = 1 && \forall e \in E_o\\
&\sum_{\bar{e}\in\Gamma_0(e)} S_{\bar{e},c} I(\bar{e},e) \le \frac{C_I}{R(e)^k T_{\mathrm{SINR}}} - \frac{N_o}{P_t} && \forall e \in E_o
\end{aligned}$$

5 Approximation Algorithm

To handle the channel allocation problem, we design an approximation algorithm based on a relaxation-rounding technique [30]. The main idea is to address the original integer optimization problem by solving another relaxed problem. More specially, we first relax the $0-1$ binary variable with real number variables ranging from 0 to 1. The induced relaxed problem can be solved in polynomial time. With the solution of the relaxed problem, we use a rounding procedure to generate the solution to the original problem. The processes of relaxation and rounding are depicted as follows.

5.1 Relaxation

To relax (10), we introduce relaxation variables $\hat{S}$, which satisfy $\hat{S}_{e,c} \in [0,1]$ for any (e,c) in $E\times C$, to substitute for the variables S in (10). The relaxed problem is expressed as (11). Note that the first constraint is different from that of (10), providing the valid range of the relaxation variables.

$$\max f(\hat{S}) \tag{11}$$

s.t.

$$\begin{aligned}
&\hat{S}_{e,c} \in [0,1] && \forall e \in E, \forall c \in C\\
&\sum_{c\in C} \hat{S}_{e,c} \le 1 && \forall e \in E\\
&\sum_{c\in C} \sum_{e\in E_s(v)\cup E_d(v)} \hat{S}_{e,c} \le \omega(v) && \forall v \in V\\
&\hat{S}_{e,c_o(e)} = 1 && \forall e \in E_o\\
&\sum_{\bar{e}\in\Gamma_0(e)} \hat{S}_{\bar{e},c} I(\bar{e},e) \le \frac{C_I}{R(e)^k T_{\mathrm{SINR}}} - \frac{N_o}{P_t} && \forall e \in E_o
\end{aligned}$$

As all the constraints are linear equations, the optimal solution of (11) can be found in polynomial time. Each element in $\hat{S}$ denotes the tendency to assign a channel to a certain edge.

5.2 Rounding

In this part, we derive the solution of (10) by rounding $\hat{S}$ with the algorithm proposed in . A bipartite graph is constructed based on the solution of (11) and then maximum weighted matching is performed on that graph to generate the solution of (10).

Let $\ddot{G} = (\ddot{U}, \ddot{V}, \ddot{E}, \ddot{W}(\ddot{E}))$ denote the bipartite graph, where $\ddot{U}$ and $\ddot{V}$ are two groups of nonadjacent nodes and $\ddot{E}$ denotes the set of edges. Each edge $\ddot{e} \in \ddot{E}$ is associated with a weight $\ddot{w}(\ddot{e})$ and $\ddot{W}(\ddot{E})$ denotes the collection of $\ddot{w}(\ddot{e})$.

First, we construct $\ddot{U}$, which denotes the transmissions in the network. For each transmission e, we can estimate the data rate of each transmission based on (5) and (7). Let $\hat{r}(e)$ denote the estimated data rate of transmission e. Note that in computing $\hat{r}(e)$, we assume transmission e is assigned with channel c as long as $\hat{S}_{e,c} > 0$. Under this assumption, each transmission is able to occupy multiple channels. Although it is not the real case, the estimated results can be referred to predict the contribution of each transmission to the global weighted throughput. We arrange the elements in E in the descending order of their estimated weighted throughput. Let $E' = (e'_1, e'_2, \cdots, e'_n)$ denote the sorted transmission sequence. For each element in E', we add a corresponding node to $\ddot{U}$ sequentially. Let $\ddot{u}_e$ denote the node corresponds to the transmission e.

Then, we create the nodes in $\ddot{V}$ based on C and $\hat{S}$. Let $\Delta_c = \lceil \sum_{e\in E} \hat{S}(e,c) \rceil$. Each channel c corresponds to Δ_c nodes in $\ddot{V}$, denoted by $\ddot{V}_c = \{\ddot{v}_{c,1}, \ddot{v}_{c,2}, \cdots, \ddot{v}_{c,\Delta_c}\}$.

With $\ddot{U}$ and $\ddot{V}$, we can set up edges between them. Let $\ddot{e}_{e,c,i}$ denote the edge between $\ddot{u}_e$ and $\ddot{v}_{c,i}$. For each channel c, if $\Delta_c \le 1$, there is only one node $\ddot{v}_{c,1}$ in $\ddot{V}_c$. For each e satisfying $\hat{S}(e,c) > 0$, add edge $\ddot{e}_{e,c,1}$ to $\ddot{E}$ and set $\ddot{w}(\ddot{e}_{e,c,1})$ to $\hat{S}(e,c)$. Otherwise, if $\Delta_c > 1$, there are multiple nodes in $\ddot{V}_c$. Let j_i denote the minimum number satisfying $\sum_{j=1}^{j_i} \hat{S}(e'_j, c) \ge i$. For $j \in \{j_{i-1}+1, j_{i-1}+2, \cdots, j_i - 1\}$, add edge $\ddot{e}_{e'_j,c,i}$ to $\ddot{E}$ and set its weight to $\hat{S}(e'_j, c)$; for e'_{j_i}, add edge $\ddot{e}_{e'_{j_i},c,i}$ and set its weight to $1 - \sum_{j=j_{i-1}+1}^{j_i-1} \ddot{w}(\ddot{e}_{e'_j,c,i})$. Specially, if $\sum_{j=1}^{j_i} \hat{S}(e'_j, c) > i$, add edge $\ddot{e}_{e'_j,c,i+1}$ and set its weight to $\sum_{j=1}^{j_i} \hat{S}(e'_j, c) - i$.

After constructing $\ddot{G}$, we perform maximum weighted matching on the bipartite graph. Let $\ddot{E}_M$ denote the results of matching. For each $\ddot{e}_{e,c,i} \in \ddot{E}$, set $S(e,c)$ to 1; then, all the unmodified elements in S are set to 0. Thus, we obtain the solution of (10). Let S_a denote the solution attained by our approximation approach and S^* denote the optimal integer solution of (10). We have the following theorem to estimate the lower bound of S_a.

Theorem 1: $f(S_a) \ge \frac{1}{2} f(S^*)$.

Proof: According to [30], the rounding result satisfying $f(S_a) + f_{\min} \ge f(\hat{S})$, in which $f_{\min}$ is the sum of the minimum transmission weighted throughput of each channel and $\hat{S}$ is the optimal fraction solution of (11). It is obvious that $f(S_a) \ge f_{\min}$. Therefore, we have $f(S_a) \ge \frac{1}{2} f(\hat{S}) \ge \frac{1}{2} f(S^*)$. □

Theorem 1 indicates that the weighted throughput achieved by the channel allocation scheme acquired with our approximation algorithm is at least $1/2$ of that of the theoretical optimal scheme.

6 Genetic Algorithm

In addition to the approximation approach, we employ genetic algorithm (GA) to address the channel allocation. The concept of genetic algorithm is to simulate the process of natural evolution, in which the individuals with a higher matching level are more likely to survive.

GA is widely used for optimization and has proved to be better than some simple heuristic algorithms, such as greedy search [23], [25]. It can reserve good local optimum allocation schemes which are likely to be part of the global allocation due to the limited interference range of wireless transmission. On the other hand, GA performs well in handling the traffic demand evolution. After the short interval between two scheduling operations, the traffic between a certain node pair would usually not experience great change unless bursty traffic happens. Since the ratio of bursty traffic is low [2], [3], for most node pairs, the traffic of a period is strongly correlated with the previous period. Therefore, the optimal scheme for the previous period is expected to yield an ideal solution for the current period. The convergence can be accelerated considerably by taking the final generation of previous period as the initial generation of current period. We define this approach as *inheriting GA search*.

Before presenting the detailed procedure, we first describe the problem mapping and the design of the main operators (selection, crossover and mutation) of GA.

In our channel allocation problem, we denote the channel assigned to a wireless link as a *DNA*. A channel allocation scheme is taken as an *individual*. A group of channel allocation schemes forms a *generation*. Let X denote an individual and $\mathbb{X}$ denote a generation.

According to the problem mapping, the DNAs of an individual can be encoded as an integer string. An important issue in DNA encoding is whether it can coordinate with the crossover operator to reserve the merits of the parent individuals. Usually, the merits of a scheme involves the channels assigned to a group of interfering transmissions. In this work, we adopt the single-point crossover. Therefore, the DNAs corresponding to the interfering transmissions should be put together so that the channels of these edges are easily preserved during crossover. A feasible approach is to perform depth-first-search in the conflict graph and number each transmission in order. The DNAs of the scheme can be encoded in the ascending order of the number.

Selection The basic idea of selection is to evaluate the fitness of all the candidate individuals. In our channel allocation problem, the fitter individual stands for the scheme that achieves a higher total weighted throughput. Therefore, we take the total weighted throughput as the fitness function f. The roulette wheel selection is adopted as the selection operator, where the selection probability of an individual X in a generation $\mathbb{X}$ is proportional to its fitness value $f(X)$. Each individual can be selected multiple times. Thus, candidate individuals with lower fitness are more likely to be eliminated.

Crossover We adopt the single-point crossover in our algorithm, in which two parent individuals are cut off at the same point and the offsprings are produced by combing different parts of the parent individuals together. In order to speed up convergence, we introduce a greedy heuristic rule, which tends to select the point that can generate offsprings with the highest fitness. Note that not all the offsprings generated by the single-point crossover are feasible solutions. The crosspoint is acceptable only if both offsprings are feasible. For each pair of parent individuals, it takes $O(n)$ time to find the best crossover point.

Mutation In GA, each generated offspring mutates at a certain probability to turn into a new individual. The mutation usually involves changing part of the DNAs. In this work, we take the optimal solution in the neighborhood of the original individual as the new individual so that the mutation can encourage the convergence of the iteration. The *k-neighborhood* ($k \in \{1, 2, \cdots, n\}$) of a solution scheme X is defined as the set of solutions in which each solution has at most k elements that are unequal to the corresponding elements in X. We just traverse the k-neighborhood of the original individual and find the best one, which takes $O(\binom{n}{k}|C|^k)$ time. Similar to crossover, we should also ensure the feasibility of the new solution in mutation.

Figure 2 shows the details of the overall procedure of our GA. In the algorithm, m feasible schemes are taken as the initial generation. Typically, these schemes can be randomly generated. Taking the final generation of the previous period as the current initial generation is an alternative optimization. For each generation, we first compute the selection probability of each individual in the current generation based on their fitness. After that, selection is executed based on the selection probability to get m new individuals. These selected individuals are randomly paired and crossover is performed over each pair. Each offspring individual experiences the mutation at the probability of p_M. Then, these offspring individuals are taken into the next iteration. The iteration is terminated if no evolution occurs during the last l generations, where a generation is considered evolutionary if the highest fitness of its individuals is higher than that of the previous generation. At last, the individual with the highest fitness in the final generation is taken as the solution.

Input: m individuals $\mathbb{X} = \{X_1, X_2, \cdots, X_m\}$; mutation probability p_M; neighborhood size k; termination threshold l
Output: the optimal solution Y

$\mathbb{X} \leftarrow \{X_1, X_2, \cdots, X_m\}$
repeat
    $\mathbb{X}_1 \leftarrow Selection(\mathbb{X})$
    Divide individuals $\mathbb{X}_1$ into pairs randomly; denote the set of pairs as $\mathbb{X}_p$
    $\mathbb{X}_2 \leftarrow \{Crossover(X_i, X_j)|(X_i, X_j) \in \mathbb{X}_p\}$
    $\mathbb{X}_3 \leftarrow \{Mutation(X, p_M, k)|X \in \mathbb{X}_2\}$
until No evolution occurs for l generations
$Y \leftarrow \arg\max_{X \in \mathbb{X}} fitness(X)$
return Y

Fig. 2. GA-Allocation Algorithm

7 Performance Evaluation

In this section, we evaluate the performance of our algorithm and the effectiveness of wireless transmissions with a series of simulations. We first describe the details of the scenario and the methodologies, and then analyze the experiment result.

7.1 Evaluation Setup and Methodologies

The experiments are performed in a simulating data center composed of 64 racks. Typically, each rack contains 20 servers, so there are more than 1000 servers in total. The racks are connected to 8-port switches and form a 3-tier tree structure. The racks are arranged in grid with the distances between two racks being 2m. The data rate of wired links is set to 1Gbps and the propagation delay is set to 2μs. The maximum number of channels is assumed to be 4 according to the specifications of existing prototype devices of 60GHz communications [31].

With regard to the input of the network, we generate inter-rack TCP traffic, whose distribution follows the property that a small number of racks account for the majority of the data, to mimic a real data center application. Specifically, we mainly refer to two traffic demand matrices. In the first matrix (denoted by M_1), the traffic

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

of hot nodes are dominant (10 racks with 95% of the total traffic); it is a typical unbalanced traffic demand matrix [6]. The distribution of another matrix (denoted by M_2) is a bit more balanced, with 20 racks generating 70% of the total data. Both the input traffic matrices are randomly generated according to the distributions. By employing the two traffic matrices, we aim to explore the performance of our wireless DCN scheme under different input traffic.

Our experiment mainly consists of three parts. In the first part of the experiments, we study the performance of GA in addressing the scheduling schemes. Since GA cannot guarantee a performance bound, we investigate its optimality and convergence speed by measuring the fitness of the optimal individual as well as the total number of generations of the search. The algorithm is run 20 times to acquire the average value of these two metrics. The impact of mutation probabilities is studied and inheriting search is evaluated by handling a sequence of evolving input traffic matrices. We assume a low ratio (5%) of randomly selected node pair may generate bursty traffic in the traffic evolution.

Furthermore, the weighted throughput achieved by both GA and the approximation algorithms are evaluated under different network parameters (input traffic, number of channels and number of antennas). To further investigate the performance of our scheduling algorithm, we select a representative scheduling algorithm GreedyPhysical [32] as comparison. The GreedyPhysical algorithm is also based on physical interference model and it schedules channel resources in a greedy manner.

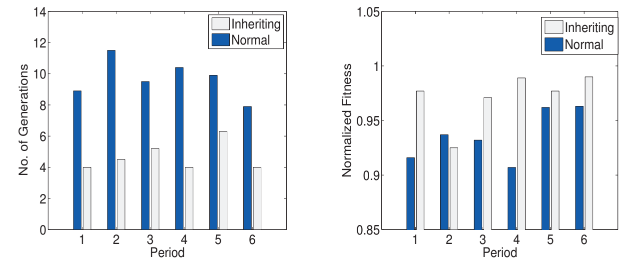
In the third part, we use the QualNet simulator to construct the simulating DCN and emulate the data transmissions in a real DCN to evaluate the effectiveness of wireless DCN. In order to demonstrate the performance enhancement caused by utilizing wireless transmissions, we compare the simulating results of wireless DCN with Ethernet-based DCN.

7.2 Simulation Results

Table 1 lists the performance of our GA-based scheduling over different mutation probabilities. As the probability increases, the convergence speed also increases while the fitness of the solutions falls. The phenomenon results from the local optimal property of the mutation. In mutation, we traverse a small neighborhood to find a better scheme. Therefore, as the frequency of mutation grows, the probability that the scheme becomes a local optimal solution also increases. Consequently, although it speeds up the convergence, converging to local optimal solution rapidly probably leads to unsatisfactory solutions. As an extreme example, if p_m is 1, the algorithm degenerates to a greedy heuristic. In short, selecting a proper mutation probability is a trade-off between the convergence speed and the optimality of the solution.

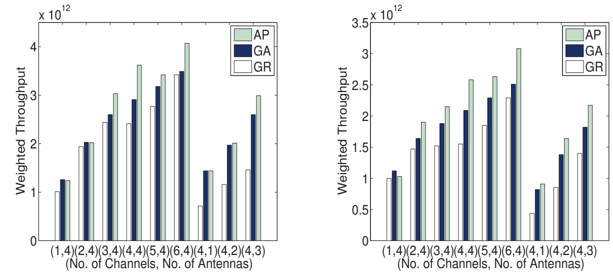
TABLE 1
Performance of GA *vs.* Mutation Probability

Mutation Probability	0.1	0.3	0.5	0.7	0.9
Generation Count	7.6	5.9	6	5.2	4.1
Normalized Fitness	1.00	9.55	8.96	8.62	8.05



(a) Convergence Time (b) Fitness

Fig. 3. Performance of GA in handling traffic demand evolution



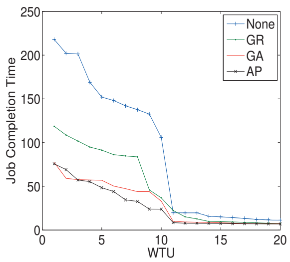
(a) Weighted Throughput under M_1 (b) Weighted Throughput under M_2

Fig. 4. Weighted throughput over channels and antennas; GR, GA and AP corresponds to the GreedyPhysical algorithm, our genetic algorithm and approximation algorithm respectively

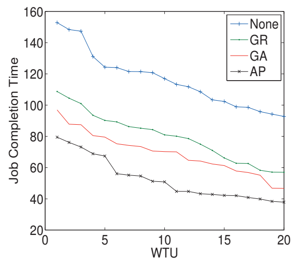
Figure 3 shows the performance of our algorithm in handling traffic demand evolution. Compared with the normal GA search, inheriting GA search not only takes a shorter convergence time but also achieves a higher fitness. Even if a few nodes randomly generate outburst traffic, the GA-based algorithm with inheriting search can still maintain high performance. The results indicate that inheriting search benefits from the solutions of the previous search because those solutions provide optimized channel allocation schemes for evolving traffic demands. Therefore, inheriting GA search usually leads to a higher fitness and a shorter convergence time than the normal GA search.

The comparison of the weighted throughput of the GreedyPhysical algorithm as well as our scheduling algorithm is illustrated in Figure 4. In general, both the approximation algorithm and GA outperform the greedy algorithm and the former achieves the highest weighted throughput. These phenomena agree with the results of the network simulations. Moreover, the gaps

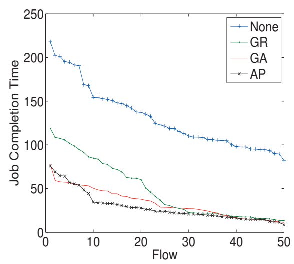
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



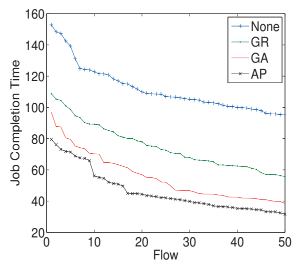
(a) Job Completion Time of WTUs under M_1



(b) Job Completion Time of WTUs under M_2

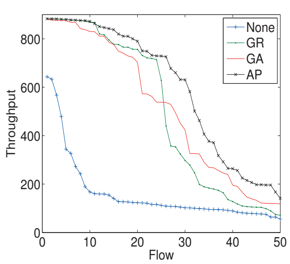


(c) Job Completion Time of Flows under M_1

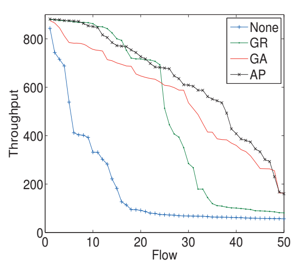


(d) Job Completion Time of Flows under M_2

Fig. 5. Job Completion Time of Wireless DCN; None denote the test case that only leverage Ethernet



(a) Throughput of Flows under M_1



(b) Throughput of Flows under M_2

Fig. 6. Throughput of Wireless DCN

among different scheduling algorithms grow as channels and antennas increase. An intuitive explanation is that the scheduling problem gets more complex when more available wireless transmission resources are provided. Thus, simple heuristics such as greedy algorithms are prone to converge to a local optimal solution and can hardly achieve a high performance.

Figure 5a and Figure 5b illustrate the job completion time of WTUs under different input traffic demands. All the WTUs are sorted in the descending order by the individual job completion time and we only involve the top 20 WTUs in the figures for the sake of visual clarity. As shown in these figures, utilizing wireless transmissions reduces the job completion time considerably; our scheduling algorithms lead to better results than the GreedyPhysical algorithm and the approximation algorithm especially does well in accelerating the transmissions.

With regard to different traffic distributions, only a small number of WTUs experience a long job completion time under M_1 while the other WTUs complete their jobs rapidly because those hot servers generate the majority of the traffic. Thus, assigning wireless links to those WTUs can significantly reduce the completion time. Yet, it is notable that the flows that do not belong to hot nodes can also benefit from wireless transmissions even if there is no wireless link assigned to them because wireless transmissions decrease the traffic load on the Ethernet. That explains why the network can benefit a lot from wireless under a relatively balanced traffic M_2.

Figure 5c and Figure 5d further demonstrate the reduction of job completion time at the flow-level. We select 50 flows that experience the longest completion time. These curves are similar to those in the previous figures, in which wireless reduces the overall completion time and our scheduling algorithms outperform the GreedyPhysical algorithm.

In addition to the job completion time, we also take throughput as another metric of performance. Since different flows have different transmission times, we define the throughput of a flow as the ratio between its traffic and job completion time. Figure 6 shows the distributions of the throughput of all the test cases. We only involve the 50 flows with the highest traffic in the figures because the throughput of some small flow appear to be very high due to the short completion time and appending the results of those small flows to the figure affects the visual clarity. The flows are arranged in the descending order by their throughput.

On the whole, the throughput of a large number of flows benefits a lot from wireless transmissions. With regard to different scheduling algorithms, a certain flow does not necessarily get a higher throughput in a better scheduling because the data rate of wireless links varies based on the interference. Thus, the throughputs of some flows under our GA appear to be lower than that under the GreedyPhysical algorithm. Yet, as the approximation algorithm achieves the highest weighted throughput and it also leads to the highest throughput, we can still infer that optimizing the weighted throughput has a significant impact on the network throughput, especially for an unbalanced traffic distribution. Obviously, by considering the delay of different transmissions, our approaches alleviate the congestion of hot nodes effectively.

Another notable phenomenon is that only a few large flows get relatively high throughput in Ethernet-based DCN, which is reflected by a sudden fall of the curve marked by None in both Figure 6a and Figure 6b. An explanation for this trend is that the transmission paths of these flows are not congested by other large ones. For example, these flows could be between the WTUs belonging to the same branches of the tree topology. Thus, these large flows can achieve a much higher throughput than others. With the help of wireless, the throughput of other large flows are highly enhanced.

8 Conclusion

In this paper, we present an exploratory investigation on utilizing wireless in DCNs. Different from existing works, we take wireless interference and SINR-based date rate into account to construct a generic model for wireless DCNs. Besides, we consider the coordination of the throughput of wireless networks and the global performance. A new metric is proposed to measure the contributions of wireless transmissions. Based on these formulations, we study the channel allocation problem. We design an approximation algorithm that provides a performance bound of $1/2$; we also explore a GA-based approach to handle evolving traffic demands efficiently. A series of simulations are performed to evaluate our wireless DCN scheme. According to the results, the global performance is improved considerably by wireless transmissions in terms of both throughput and job completion time; in addition, our scheduling algorithms are effective and efficient in tackling the channel allocation problem.

Acknowledgments

This work is supported by NSF of China (61120106008, 60911130511, 60873252), 973 Program of China (2009CB320503, 2011CB302800), and the US NSF grant CNS-0831852.

References

[1] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.
[2] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of datacenter traffic: Measurements & analysis," in *IMC '09*, 2009.
[3] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," in *WREN '09*. ACM, 2009, pp. 65–72.
[4] X. Wen, K. Chen, Y. Chen, Y. Liu, Y. Xia, and C. Hu, "Virtualknotter: Online virtual machine shuffling for congestion resolving in virtualized datacenter," in *ICDCS '12*, 2012.
[5] P. Smulders, "Exploiting the 60ghz band for local wireless multimedia access: Prospects and future directions," *IEEE Communications Magazine*, vol. 40(1), pp. 140–147, 2002.
[6] J. P. S. Kandula and P.Bahl, "Flyways to de-congest data center networks," in *HotNets 09*, 2009.
[7] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *SIGCOMM '08*. ACM, 2008, pp. 63–74.
[8] 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 *SIGCOMM '09*. ACM, 2009, pp. 39–50.
[9] 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 *SIGCOMM '09*. ACM, 2009, pp. 51–62.
[10] 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 *SIGCOMM '08*. ACM, 2008, pp. 75–86.
[11] S. Li, D. Chuanxiong Guo Haitao Wu Kun Tan Yongguang Zhang Lu, "Ficonn: Using backup port for server interconnection in data centers," in *INFOCOM '09*, 2009, pp. 2276–2285.
[12] 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.
[13] K. Ramachandran, R. Kokku, R. Mahindra, and S. Rangarajan, "60 GHz data-center networking: Wireless => worry less?" *NEC Technical Report*, 2008.
[14] Y. Cui, H. Wang, and X. Cheng, "Wireless link scheduling for data center networks," in *ICUIMC '11*. ACM, 2011, pp. 44:1–44:9.
[15] Y. Cui, H. Wang, X. Cheng, and B. Chen, "Wireless data center networking," *Wireless Communications, IEEE*, vol. 18, no. 6, pp. 46–53, 2011.
[16] Y. Cui, H. Wang, and X. Cheng, "Channel allocation in wireless data center networks," in *INFOCOM '11*. IEEE, 2011, pp. 1395–1403.
[17] J. P. S. Kandula and P.Bahl, "Your data center is a router: The case for reconfigurable optical circuit switched paths," in *HotNets 09*, 2009.
[18] N. Farrington, G. Porter, S. Radhakrishnan, H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, "Helios: a hybrid electrical/optical switch architecture for modular data centers," in *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4. ACM, 2010, pp. 339–350.
[19] G. Wang, D. Andersen, M. Kaminsky, K. Papagiannaki, T. Ng, M. Kozuch, and M. Ryan, "c-through: Part-time optics in data centers," in *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4. ACM, 2010, pp. 327–338.
[20] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 2, pp. 50–65, 2004.
[21] H. Skalli, S. Ghosh, S. Das, L. Lenzini, and M. Conti, "Channel assignment strategies for multiradio wireless mesh networks: issues and solutions," *Communications Magazine, IEEE*, vol. 45, no. 11, pp. 86–95, 2007.
[22] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *Automatic Control, IEEE Transactions on*, vol. 37, no. 12, pp. 1936–1948, 1992.
[23] A. Zomaya and M. Wright, "Observations on using genetic-algorithms for channel allocation in mobile computing," *TPDS 2002*, vol. 13, no. 9, pp. 948–962, 2002.
[24] S. Patra, K. Roy, S. Banerjee, and D. Vidyarthi, "Improved genetic algorithm for channel allocation with channel borrowing in mobile computing," *IEEE Transactions on Mobile Computing*, pp. 884–892, 2006.
[25] Y. Ding, Y. Huang, G. Zeng, and L. Xiao, "Channel assignment with partially overlapping channels in wireless mesh networks," in *WICON '08*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 1–9.
[26] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
[27] A. Curtis, W. Kim, and P. Yalagandula, "Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection," in *INFOCOM '11*. IEEE, 2011, pp. 1629–1637.
[28] A. Tavakoli, M. Casado, T. Koponen, and S. Shenker, "Applying nox to the datacenter," *HotNets '09*, 2009.
[29] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks," in *Proceedings of the 7th USENIX conference on Networked systems design and implementation*. USENIX Association, 2010, pp. 19–19.
[30] D. Shmoys and É. Tardos, "An approximation algorithm for the generalized assignment problem," *Mathematical Programming*, vol. 62, no. 1, pp. 461–474, 1993.
[31] L. Caetano and S. Li, *Sibeam Whitepaper: Benefits of 60 GHz*, 2005.
[32] G. Brar, D. Blough, and P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," in *MobiCom '06*. ACM, 2006, pp. 2–13.

Yong Cui Yong Cui, Ph.D., Associate Professor in Tsinghua University, Council Member in China Communication Standards Association, Co-Chair of IETF IPv6 Transition WG Softwire. Having published more than 100 papers in refereed journals and conferences, he is also the winner of Best Paper Award of ACM ICUIMC 2011 and WASA 2010. Holding more than 40 patents, he is one of the authors in RFC 5747 and RFC 5565 for his proposal on IPv6 transition technologies. His major research interests include mobile wireless Internet and computer network architecture.

Hongyi Wang Hongyi Wang received his bachelor degree in Computer Science from Tsinghua University, China in 2009. He is pursuing the master degree in the Department of Computer Science and Technology at Tsinghua University, supervised by Prof. Yong Cui. He won the best paper award of ACM ICUIMC 2011. His research interests include data center networking, wireless networks and mobile system.

Xiuzhen Cheng Xiuzhen Cheng received her MS and PhD degrees in computer science from the University of Minnesota – Twin Cities, in 2000 and 2002, respectively. She is an associate professor at the Department of Computer Science, The George Washington University, Washington DC. Her current research interests focus on cognitive radio networks, mobile handset networking systems (mobile health and safety), wireless and mobile computing, sensor networking, wireless and mobile security, and algorithm design and analysis. She has served on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops. She also has chaired several international conferences. She worked as a program director for the US National Science Foundation (NSF) from April to October in 2006, and from April 2008 to May 2010. She received the NSF CAREER Award in 2004. She is a senior member of IEEE and a member of ACM.

Dan Li Dr. Dan Li joined the Computer Science Department of Tsinghua University in Mar 2010, where he is now an associate professor. From Jan 2008 to Feb 2010, he worked as an associate researcher in the Wireless & Networking Group of Microsoft Research Asia. He received his M.E. degree and Ph.D from Tsinghua University in 2005 and 2007 respectively, both in computer science. Before that, he spent four undergraduate years in Beijing Normal University and got a B.S. degree in 2003, also in computer science. His current research interests include computer network architecture, data center networks and green networking. He has published more than 30 technical papers in referred conferences and journals. He serves as a TPC member for international conferences such as INFOCOM, GLOBECOM, ICCCN. He is a member of IEEE and ACM.

Antti Ylä-Jääski Prof Dr.Tech. Antti Ylä-Jääski received his PhD in ETH Zuerich 1993. Antti has worked with Nokia 1994-2009 in several research and research management positions with focus on future Internet, mobile networks, applications, services and service architectures. He has been a professor for Telecommunications Software, Department of Computer Science and Engineering, Aalto University since 2004. Antti has supervised 188 master's thesis and 12 doctoral dissertations during his professorship until end of 2011. He has currently five ongoing research projects in the areas of Green ICT, mobile computing, services and service architectures: "Future Internet", "Cloud Software", "Internet of Things", "Massive Scale Machine-to-Machine Service" and "Energy-Optimized Mobile Computing". Antti has published over 50 international conference and journal articles.