This paper was presented as part of the main technical program at IEEE INFOCOM 2011

Channel Allocation in Wireless Data Center Networks

Yong Cui
Tsinghua University
Beijing, P.R.China
Email: cuiyong@tsinghua.edu.cn

Hongyi Wang
Tsinghua University
Beijing, P.R.China
Email: wanghongyi09@mails.tsinghua.edu.cn

Xiuzhen Cheng
The George Washington University
Washington, DC 20052, USA
Email: cheng@gwu.edu

***Abstract*—Unbalanced traffic demands of different data center applications are an important issue in designing Data center networks (DCNs). In this paper, we present our exploratory investigation 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 a DCN by considering both the wireless interference and the adaptive transmission rate. Moreover, both throughput and job completion time are taken into account to evaluate the impact of wireless transmissions on the global performance. Based on this model, we formulate the channel allocation in wireless DCNs as an optimization problem and design a genetic algorithm (GA) based approach to address it. To demonstrate the effectiveness of wireless transmissions as well as our GA-based algorithm in a wireless DCN, extensive simulation study is carried out and the results validate our design.**

I. Introduction

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

However, current DCNs, which evolve from the Enterprise LAN networks, come across more and more difficulties with the growth of cloud computing. First, the rapidly increasing size of data centers brings challenges to DCN. By the year of 2006, Google has got over $450,000$ servers in its 30 data centers [1]. For traditional Ethernet solutions, expensive high-end switches and a large number of wires are necessary to construct a DCN containing thousands of servers, which leads to great troubles in wiring and maintenance.

On the other hand, data center applications that cause unbalanced traffic distributions suffer from inadequate network capacity. Based on the traffic statistics obtained from a real-world data center, typical applications such as map-reduce [2] usually generate a traffic demand with only a few nodes being hot (i.e., these nodes need to transmit a high volume of traffic). Figure 1 shows an example traffic demand matrix, where darker points stand for higher traffic demands. Although the matrix is quite sparse, those hot nodes cause loss on edge links with a high probability [3] 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 for certain nodes to alleviate their congestions.

To tackle 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 the maintenance is relatively convenient. Second, direct links between servers are easy to achieve with wireless in the scale of a data center, which can avoid the extra cost of multi-hop transmissions. Moreover, variable wireless connections can be set up on-demand. Therefore, it is possible to adjust the topology dynamically to provide more network capacity for hotter nodes. In brief, the flexibility of wireless transmissions provide a feasible approach to address the non-deterministic unbalanced traffic distribution of data center applications.

Nevertheless, challenges still exist in employing wireless in DCNs. To start with, wireless transmissions should be rapid enough to support high-speed communications. Current DCNs are mostly built based on gigabit Ethernet, whose data rate is highly beyond that of commodity wireless devices.

Besides, delicate wireless scheduling mechanisms are required to effectively enhance the performance of the whole DCN. For example, wireless links should be established appropriately to alleviate the congestion of hot nodes; channels should be allocated properly to avoid interference.

Moreover, the wireless network should coordinate with the global optimization. In other words, the performance of wireless transmissions (typically measured by throughput) and the global job completion time should be jointly considered.

As for data rate, the state-of-art wireless technology has met the requirements of gigabit transmissions. Extremely highly frequency (EHF) communications support directional high-speed transmissions and are expected to be a feasible gigabit wireless solution [4]. In particular, IEEE 802 has been working on the standards of the communications at 60GHz (IEEE 802.11ad); prototype devices have also been produced [5].

With regard to the wireless scheduling, it seems to be similar to that of multi-channel multi-radio wireless mesh networks (WMN). However, they are different essentially. First, one of the most important concerns in the scheduling of WMN is the multi-hop wireless communications. Yet, in research of wireless DCN, we focus on single hop transmissions for high efficiency in demand. Second, while we only pay attention to wireless links in WMN, the joint effort of wireless networks and Ethernet infrastructure should be considered in DCN.

978-1-4244-9921-2/11/$26.00 ©2011 IEEE

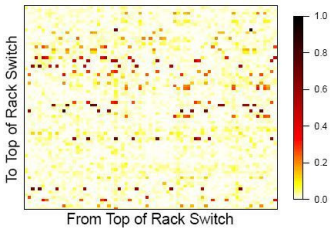


Fig. 1. Matrix of application demands between top of rack switches [6]

Moreover, the nodes in WMN usually execute respective tasks while servers in a DCN cooperate with each other to complete a common job. Therefore, it is difficult to adapt solutions for WMN to wireless DCNs.

In this paper, we focus on utilizing wireless transmissions in DCNs. We propose a hybrid DCN architecture in which wireless networks work as a supplementary to Ethernet infrastructure to address the congestion of hot nodes. To the best of our knowledge, this is the first work that provides detailed technical approach of wireless DCN. The contribution of this work is multi-fold. First, we perform a novel problem formulation for wireless DCN. A realistic interference formalization and the adaptive transmission rate are considered in the model. Furthermore, we pay attention to the joint optimization of the throughput of wireless networks and the global job completion time. Second, we introduce a genetic algorithm (GA) to tackle the channel allocation problem. The GA-based approach can find the solution efficiently, especially when employing inheriting search. Third, we conduct simulation-based performance evaluation. The simulation is carefully designed to mimic the scenarios of data center applications. Various experiments are performed to demonstrate the effectiveness of wireless transmissions as well as our GA-based algorithm.

The rest of the paper is organized as follows. Section II depicts the most related work. Our system model is elaborated in Section III and Section IV describes the centralized scheduling mechanism, including the details of the GA-based algorithm. Simulation methods and results are presented and analyzed in Section V. Section VI concludes the paper.

II. Related Work

There has been a lot of research on the interconnection architectures and the routing mechanisms of DCNs. 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 ones. 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. K. Ramachandran et al. first 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, including interference and number of radios, are not considered in the scheduling mechanism. Therefore, a lot of problems remain to be investigated to substantiate a wireless DCN. Another work [14] proposes to utilize optical circuit switches for high-speed direct communications between racks. The optical switch is scheduled based on the traffic demands to maximize throughput, which is similar to Flyway.

GA-based approaches have been proposed to handle the channel allocation problem in various wireless networks. Zomaya *et al.* [15] 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.* [16] improve the algorithm by introducing a new pluck operator. Ding *et al.* [17] 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 I) and we design our own crossover and mutation operators to improve the performance of the GA algorithm.

III. System Model

A. Wireless Transmission

In this paper, we propose a generic approach to utilize wireless in DCNs such that the adoption of wireless transmissions is 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* refers to a group of servers that uses the same set of antennas to transmit data 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 treat certain specific structures in the corresponding architectures as wireless transmission units. For example, the BCube0 structure in BCube is an reasonable candidate for a wireless transmission unit [12].

Based on Definition 1, we classify 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 as defined in Definition 2.

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 transmissions.

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 the volume of the traffic from v_1 to v_2 is more than 0.

B. Channel Allocation and Interference

For a given wireless transmission graph, channels should be assigned to the edges to carry out wireless communications. In this work, we assume different channels are orthogonal. Let C denote the set of channels. When allocating channels, we assign each edge $e \in E$ with an integer $c(e) \in \{0, 1, \ldots, |C|\}$, in which each non-zero integer corresponds to a certain channel and 0 means assigning no channel to e. Note that not all the wireless transmissions should be carried out simultaneously because some of them may cause serious interference to others and therefore have a negative impact on the global performance. If an edges is assigned a channel, it is called an *active edge*; otherwise, it is an *idle edge*.

The set of channels allocated to all the edges form a channel allocation scheme of the wireless transmission graph. Assuming $|E| = n$ (we follow this assumption in the remaining sections of this paper), the channel allocation scheme can be expressed by a vector $X = (x_1, x_2, \cdots, x_n)$, in which each element x_i stands for the channel assigned to a specific edge e_i.

One of the problems in channel allocation is that the transmission on an edge is possibly interfered by the nearby transmissions working on the same channel.

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

The decision of a conflict edge involves the physical position of the nodes and the assigned channels. With regard to physical position, we adopt the interference range model, in which a sender node causes interference on all the nodes within its interference range. Note that our model does not rely on certain antenna techniques. The interference range of a node with an omni-directional antenna is usually defined as a unit disk while that of a directional antenna depends on the relative position of the two endpoints and the beam-forming patterns. Whichever antenna techniques is employed, we just adopt the corresponding interference range model.

Data transmissions in DCNs should be reliable so acknowledgment is required. We transmit data packets and acknowledgment 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 interference range of another edge $\bar{e} = (\bar{v}_1, \bar{v}_2)$ if v_2 is in the interference range of $\bar{v}_1$.

If e is in the interference range of $\bar{e}$, we consider $\bar{e}$ as a *potential conflict edge* of e. If $\bar{e}$ is a potential conflict edge of e and $c(\bar{e}) = c(e) \neq 0$, 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. The interference relationship can be illustrated with a conflict graph, in which each node denotes a transmission and a directed edge (v_1, v_2) indicates that v_1 potentially interferes v_2.

C. SINR and Data Rate

In the research on wireless networks, the protocol interference model and the physical interference model are often used to determine the effect of interference [18]. 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 T_{SINR}. We adopt the latter model in this work. Thus, the transmission on $e = (v_1, v_2)$ is successfully performed if and only if:

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

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 emitted from 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 emitted from 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.

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

For simplicity, we assume that all the antennas have the same gain and the same transmit power. 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 $R(\bar{v}_1, v_2)$.

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

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 based on the Friis equation. In short, the SINR of e can be computed with (5), where $R(e)$ is equal to $R(v_1, v_2)$.

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

SINR is not only the necessary condition of successful transmissions but also an important factor that influences 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.

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

In this work, we assume the data rate is proportional to the capacity. Assuming all the channels have the same bandwidth B and the rate between the data rate and the capacity is β, the date rate of e can be expressed with (7).

$$r(e) = \beta B \log_2(1 + \mathrm{SINR}(e)) \tag{7}$$

With (5) and (7), we can compute the data rate of each transmission as long as the channel allocation scheme and the interference relationships are specified.

IV. Scheduling Mechanism

Based on the model of wireless data center networks, we propose a centralized scheduling mechanism for wireless transmissions, in which a central controller periodically gathers the information about traffic demands from all the units as well as schedules wireless links for the inter-unit transmissions. The scheduling consists of two steps: the first step is to construct a wireless transmission graph based on the traffic information; and the second step is to perform channel allocation in the wireless transmission graph. We provide the details of the two steps in this section.

A. Constructing A Wireless Transmission Graph

1) Selecting Transmissions: When constructing a wireless transmission graph, the central controller converts the traffic demands to a wireless transmission graph for latter scheduling. Although the converting itself is quite easy, the problem lies in the large number of transmissions. As a well-known NP-hard problem, channel allocation is usually handled by using heuristic algorithms, whose time cost grows with the increase of the number of scheduled objects. For DCNs, the huge number of transmissions leads to an excessively high cost. Therefore, efforts should be made to decrease the size of the channel allocation problem.

A feasible approach is to select a part of the transmissions to construct the graph rather than involve all the transmissions. As for this approach, the key problem is to determine which transmissions to select. Recall the motivations to introduce wireless transmissions into DCNs. It is the high traffic of sparse hot nodes that causes congestion and put off the completion of a job. Therefore, limited wireless channel resources should be used to serve those nodes. In other words, transmissions belonging to the hot nodes should be selected with a high priority.

Furthermore, as mentioned before, the scheduling is a periodical mechanism which means that the channel allocation scheme will be carried out for a period after each allocating operation. Therefore, if the traffic of a transmission is so low that the corresponding wireless link keeps idle for the most of the period, the transmission should be assigned to wired links rather than occupy wireless channel resources.

Besides, a wireless transmission is restricted by the valid transmission range. As for 60GHz communications, the range is about 10m. For $e = (v_1, v_2)$, if the distance between v_1 and v_2 exceeds the valid transmission range, the corresponding edge should be removed from the graph as it is impossible for the antennas to carry out the corresponding wireless transmission.

2) Weighting Transmissions: In conventional wireless scheduling approaches, the total throughput is often taken as the metric of performance. 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, put off the global job completion time. Another example is that some flows are expected to experience much longer delay than others via Ethernet transmission because of the static topology and the routing mechanisms. Under either condition, it is obvious that setting up wireless links for certain transmissions is more profitable even if the corresponding data rate is not as high as that of wired links.

We formalize this property as the *utility* of the transmission, 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 transmission.

In this work, we employ the network delay to estimate the utility of a transmission. Intuitively, a transmission with a high network delay, caused by either congestion or a long transmission path, is suitable to be assigned to wireless transmissions. Therefore, the utility should be directly proportional to the network delay. We define the utility as (8), where $d(e)$ is the network delay of e and μ is a positive coefficient. Note that utility is a scalar variable.

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

Generally speaking, the network delay can be estimated based on the traffic distribution and the Ethernet architecture. Yet, this work does not focus on how to perform the estimation. In fact, our channel allocation algorithm does not rely on how utility is computed. As long as each edge of the wireless transmission graph is assigned a weight, our scheduling approach can be applied to generate the corresponding channel allocation scheme.

B. Allocating Channels

After constructing the wireless transmission graph, channel can be assigned based on the graph. In this subsection, we first formulate the channel allocation problem and then propose a genetic algorithm to handle the problem.

1) Formulation of the Channel Allocation Problem: We formalize channel allocation as an optimization problem and the channel allocation scheme is taken as the variable of the problem. As for the objective of channel allocation, we propose Definition 5 to estimate the impact of a wireless transmission on the global performance based on the definition of utility. The objective function of the optimization problem is the total weighted throughput of all the wireless transmissions.

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

Several constraints should be considered in channel allocation. First, the number of active edges belonging to a node should not be more than the number of antennas of that node. Second, the assigned channels should be in the available channel set C. Third, for each active edge, its SINR should be higher than the threshold as shown in (1).

Let $E_s(v)$ denote the set of edges whose source node is v and $E_d(v)$ be the set of edges whose destination node is v. Based on the above analysis, the channel allocation problem can be expressed with (9). The optimal solution of the problem is the channel allocation scheme that meets the constraints in (9) and maximizes the total weighted throughput.

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

subject to

$$|\{e | e \in E_s(v) \cup E_d(v) \wedge c(e) > 0\}| \leq \omega(v), \forall v \in V$$
$$c(e) \in \{0, 1, 2, \cdots, |C|\}, \forall e \in E$$
$$\mathrm{SINR}(e) \geq T_{\mathrm{SINR}}, \forall e \in \{\bar{e} | \bar{e} \in E \wedge c(\bar{e}) > 0\}$$

2) Genetic Algorithm: In this work, we tackle the channel allocation problem with a GA-based scheduling algorithm. The concept of genetic algorithm is to simulate the process of natural evolution, in which the individuals with higher fitnesses are more likely to survive.

GA is advantageous in solving the channel allocation problem. First, the delicate design of GA enables it to achieve a better performance in handling NP-hard problems than simple heuristics, such as naive greedy search. Second, the channel assignment problem has inherent local optimization property [17]. An allocation scheme for a subnetwork with less interference locally is more likely to be part of the global allocation scheme because the interference range of wireless transmissions is limited. The property fits well into GA because the selection operator and the crossover operator of GA can reserve optimal local allocation schemes. Third, GA does well in handling the traffic demand evolution. The traffic distribution of a period is strongly correlated to that of 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 scheduling algorithm, we first describe the problem mapping and our design of the main operators (selection, crossover and mutation) of GA.

a) DNA, Individual and Generation: In the 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 form 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 preserve 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 arranged 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 [17]. The DNAs of the scheme can be encoded in the ascending order of the number.

b) Selection: The basic idea of selection is to evaluate the fitness of all the candidate individuals, which is done by computing the fitness function of each individual. In our channel allocation problem, the fitter individual stands for the scheme that achieves a higher total weighted throughput. Therefore we simply take the total weighted throughput as the fitness function f.

We adopt the roulette wheel selection as the selection operator, where the selection probability $p_s(X)$ of an individual X in a generation $\mathbb{X}$ is calculated based on (10). The interval $[0, 1]$ is divided into subintervals in such a way that each individual corresponds to a subinterval with the length proportional to its selection probability.

$$p_s(X) = \frac{f(X)}{\sum_{\bar{X} \in \mathbb{X}} f(\bar{X})} \tag{10}$$

When selection is executed, random numbers ranging from 0 to 1 are generated to select individuals. For each random number, the individual that corresponds to the interval including the random number is selected. Each individual can be selected multiple times. Thus, candidate individuals with lower fitness are more likely to be eliminated. The selection operator is detailed in Figure 2.

Input: m individuals $\mathbb{X} = \{X_1, X_2, \cdots, X_m\}$
Output: m selected individuals $\mathbb{Y} = \{Y_1, Y_2, \cdots, Y_m\}$
1: $\mathbb{Y} \leftarrow \varnothing$
2: $p_s(X_i) \leftarrow \frac{f(X_i)}{\sum_{X \in \mathbb{X}} f(X)}$, for $i = 1, 2, \cdots, m$
3: $b_i \leftarrow \sum_{j=1}^{i} p_s(X_j)$, for $i = 0, 1, \cdots, m$
4: **for** $j = 1$ to m **do**
5: Generate a random number in $[0, 1)$, denoted as δ
6: Find i such that $b_{i-1} \leq \delta < b_i$
7: $\mathbb{Y} \leftarrow \mathbb{Y} + X_i$
8: **end for**
9: **return** $\mathbb{Y}$

Fig. 2. Selection algorithm

c) 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 admissible only if both offsprings are feasible solutions. Figure 3 details the procedure of crossover. For each pair of parent individuals, it takes $O(n)$ time to find the best crossover point.

Input: two parent individuals X_1, X_2
Output: two offspring individuals Y_1, Y_2
1: $f_m \leftarrow 0$
2: $Y_1 \leftarrow 0$, $Y_2 \leftarrow 0$
3: **for** $i = 0$ to n **do**
4: $(Y_1', Y_2') \leftarrow$ single-point crossover of X_1 and X_2 at i
5: **if** Y_1', Y_2' are feasible **and** $\text{Max}\{f(Y_1'), f(Y_2')\} > f_m$ **then**
6: $f_m \leftarrow \text{Max}\{f(Y_1'), f(Y_2')\}$
7: $Y_1 \leftarrow Y_1'$, $Y_2 \leftarrow Y_2'$
8: **end if**
9: **end for**
10: **return** Y_1, Y_2

Fig. 3. Crossover algorithm

d) Mutation: In GA, each generated offspring mutates at a certain probability to turn into a new individual. The mutation usually changes 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 concept of neighborhood is given in Definition 6.

Definition 6: Given a wireless transmission graph $G = (V, E)$, the *k-neighborhood* ($k \in \{1, 2, \cdots, n\}$) of a solution scheme X is the set of solutions in which each solution has at most k elements that are unequal to the corresponding elements in X

Let $N(X, k)$ denote the k-neighborhood of X. Assuming X is optimal in $N(X, k)$, the larger the k, the higher the possibility of X being the global optimal solution; if $k = n$, X is definitely the global optimal solution. It is obvious that it takes a huge cost to find the optimal solution in a large neighborhood. However, we only need to search in a relatively small neighborhood (typically $k = 1$ or 2) in mutation. Therefore the time cost is tolerable.

Input: origin individual X; mutation probability p_m; neighborhood size k
Output: new individual Y
1: $Y \leftarrow X$
2: Generate a random number in $[0, 1)$, denoted as δ
3: **if** $\delta < p_m$ **then**
4: **for all** Y' in $N(X, k) - X$ **do**
5: **if** Y' is feasible **and** $f(Y') > f(Y)$ **then**
6: $Y \leftarrow Y'$
7: **end if**
8: **end for**
9: **end if**
10: **return** Y

Fig. 4. Mutation algorithm

Figure 4 details the procedure of mutation. We 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.

e) GA-based scheduling algorithm: Based on the problem mapping and the designs of selection, crossover, and mutation, we depict the GA-based scheduling algorithm in Figure 5.

Input: m initial 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
1: **repeat**
2: $\mathbb{X}_1 \leftarrow Selection(\mathbb{X})$
3: Divide the individuals in $\mathbb{X}_1$ into pairs randomly; denote the set of pairs as $\mathbb{X}_p$
4: $\mathbb{X}_2 \leftarrow \{Crossover(X_i, X_j) | (X_i, X_j) \in \mathbb{X}_p\}$
5: $\mathbb{X}_3 \leftarrow \{Mutation(X, p_m, k) | X \in \mathbb{X}_2\}$
6: **until** No evolution occurs for l generations
7: $Y \leftarrow \arg\max_{X \in \mathbb{X}} f(X)$
8: **return** Y

Fig. 5. GA-based scheduling algorithm

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

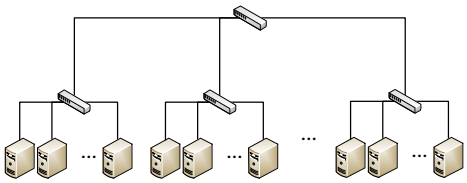


Fig. 6. Experiment DCN architecture

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.

V. 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.

A. Evaluation Setup and Methodologies

1) Experiment Setup: The experiments are performed in a simulating data center composed of 64 racks. Typically, each rack is equipped with 20 servers so there are in total more than 1000 servers. The racks are connected to 8-port switches and form a 3-layer tree structure. Figure 6 illustrates the DCN architecture used in the simulation study.

The data rate of wired links is set to 1Gbps and the propagation delay is set to 2μs. As for wireless networks, since no standard has been published to specify the parameters of 60GHz communications, we just follow the specifications of existing prototype devices. According to [5], the channel bandwidth of 60GHz is 2.5GHz and the running frequency ranges from 57GHz to 66GHz. Thus, we assume $|C| = 4$.

2) Experiment Design: Our experiment consists of two parts. In the first part, we use the QualNet simulator to establish the experiment DCN structure illustrated in Figure 6 and simulate the data transmissions in a real DCN.

With regard to the input to the network, we generate inter-rack TCP traffic, whose distribution follows the property that a few racks account for the majority of the traffic, 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 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 slightly more balanced, with 20 racks generating 70% of the total traffic. By employing different traffic matrices, we aim to investigate the impact of the traffic distributions.

For each input matrix, we prepare different cases as shown in Table I. Test Case WIRE is taken as comparison to demonstrate the impact of wireless transmissions; besides, the utility of the transmissions are computed according to their delay in Test Case WIRE. For the cases that enable wireless networks, we run the GA-based scheduling algorithm and set up wireless links based on the channel allocation scheme. By comparing the cases with different ω, we inspect the performance improvement as the number of wireless links increases. We also test the case that only throughput is considered in scheduling (the utilities of all the transmissions are treated as 1) to examine the effectiveness of considering utility.

As for the second part of the experiments, we study the performance of the GA-based scheduling algorithm, which involves the optimality of the solution and the convergence speed. The former corresponds to the fitness of the solution while the latter is measured by counting the bred generations when terminating the search. By running the algorithm 20 times for each scenario, we obtain the average values of the two metrics. Note that it does not make sense to compare fitness of the schemes for different traffic demand matrices directly. Therefore we turn to *normalized fitness*, which is computed by dividing the fitness by the largest fitness we have obtained for the same matrix.

Another problem about GA is the impact of the mutation operator. We investigate the problem by comparing the performance of the algorithm with different mutation probabilities.

Moreover, we also test the ability to handle the traffic demand evolution. A traffic demand matrix sequence with a strong time correlation is taken as the input. More specifically, a new traffic matrix is generated by adjusting the previous matrix in a small range. In addition to the basic matrix sequence, we also randomly select new hot nodes in generating new demand matrices to mimic the condition of outburst traffic. By comparing the performance of the inheriting GA search and the normal GA search, we reveal the advantages of GA in traffic-based scheduling.

B. Simulation Results

1) Impact of Wireless Transmissions: Figure 7a and Figure 7b illustrate the job completion time under different input traffic demands. All the nodes are arranged in the descending order of the individual completion time and that of the fist node is considered as the global job completion time. For the sake of visual clarity, we only involve the top 20 nodes in Figure 7a as the completion time of the remaining nodes is quite small.

TABLE I
Parameters of the Test Cases

Test Case	Wireless Enabled	No. of Antennas	Utility Considered
WIRE	No	/	/
W4U	Yes	4	Yes
W2U	Yes	2	Yes
W4	Yes	4	No

As shown in these figures, utilizing wireless transmissions reduces the job completion time significantly; the increase of the number of available antennas can further shorten the time as more antennas lead to more wireless links.

For different traffic distributions, the effect raises as the distribution gets more unbalanced. Yet, wireless networks still decrease the completion time by up to 30% for a relatively balanced distribution (M_2). Especially, the racks other than hot nodes can also benefit from wireless transmissions even if there is no wireless links attached to them, which is because wireless transmissions decrease the traffic load on the Ethernet.

It is noteworthy that the global job completion time decreases for M_1 if we take the utilities of different transmissions into consideration. More specifically, the completion time of the hottest node in the Test Case W4U is shorter than that in the Test Case W4 while other nodes are likely to experience longer transmission times in the Test Case W4U. The results indicate that optimizing weighted throughput outperforms merely optimizing the throughput in terms of allocating the limited wireless channel resources to alleviate the congestion of the hottest nodes. As a side effect, other nodes would get fewer opportunities to utilize wireless transmissions, which lengthens their completion time. On the other hand, the two test cases achieve similar completion times for M_2 because the utilities of different transmissions are close to each other under a relatively balanced traffic distribution.

In addition to the job completion time, we also take throughput as another metric of the global performance. Since different transmissions have different completion times, measuring the throughput of a node is meaningless. Therefore, we pay attention to the throughput of each transmission. Figure 7c and Figure 7d illustrate the distributions of the throughput of all the test cases, where transmissions are arranged in the descending order of the throughput. We only involve the top 100 transmissions in the figures as the throughput of other transmissions is negligible. Table II records the total throughput of all the test cases.

Generally speaking, the total throughput benefits a lot from wireless transmissions. Increasing the number of antennas improves the throughput considerably as the raise of the throughput mainly results from the additional capacity provided by the wireless links

Figure 7c and Figure 7d indicate that optimizing the weighted throughput based on utilities has a significant impact on the network throughput, especially for an unbalanced traffic distribution. By considering the delay of different transmissions, our approach alleviates the congestion of hot nodes effectively. Consequently, all the transmissions benefit and maximizing weighted throughput does better in improving the overall throughput than merely maximizing the throughput.

2) Scheduling Algorithm Performance: Table III shows the performance of our GA-based scheduling algorithm over different mutation probabilities. As the probability increases, the convergence speed also increases while the fitness of the solutions falls. This is caused by 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 result in unsatisfactory solutions. As an extreme example, if p_m is 1, the algorithm turns 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.

Figure 8 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 as those solutions provide optimized channel allocation schemes for evolving traffic demands. As a result, inheriting GA search usually leads to a higher fitness and a shorter convergence time than the normal GA search.

VI. Conclusion

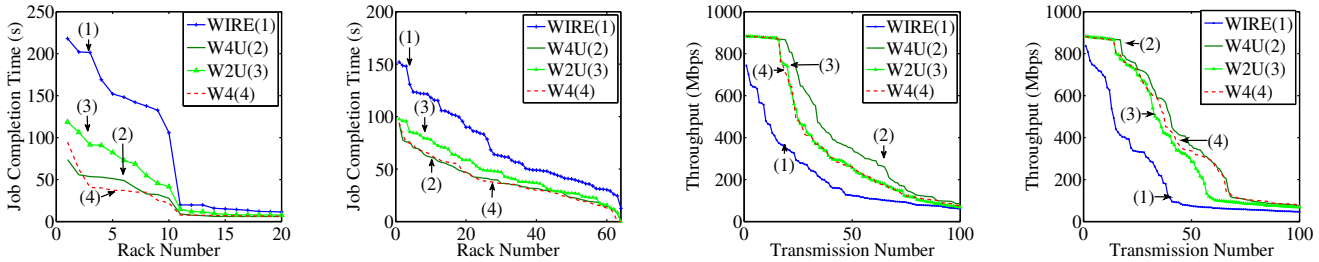
In this paper, we present an exploratory investigation on utilizing wireless networks in DCNs. Different from existing works, we take wireless interference and SINR-based date rate into consideration to build a generic model for wireless DCNs. Besides, we take into account the coordination of the throughput of wireless networks and the global performance. A new metric is proposed to measure the contributions of wireless transmissions. Based on these considerations, we study the channel allocation problem and design a GA-based scheduling algorithm by implementing the procedures of selection, crossover, and mutation. We perform elaborate simulations to evaluate the effectiveness of wireless transmissions in a DCN. According to the simulation results, the global performance of a wireless DCN is improved considerably in terms of both throughput and job completion time. Moreover, we analyze the performance of the GA-based algorithm based on a series of experiments and demonstrate that it is an excellent approach to tackle the channel allocation problem in wireless DCNs.

TABLE II
TOTAL THROUGHPUT OF ALL THE TEST CASES

Test Case	WIRE	W4U	W2U	W4
M_1	26.2Gbps	47,7Gbps	40.6Gbps	40.7Gbps
M_2	27.7Gbps	48.4Gbps	42.8Gbps	46.2Gbps

TABLE III
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



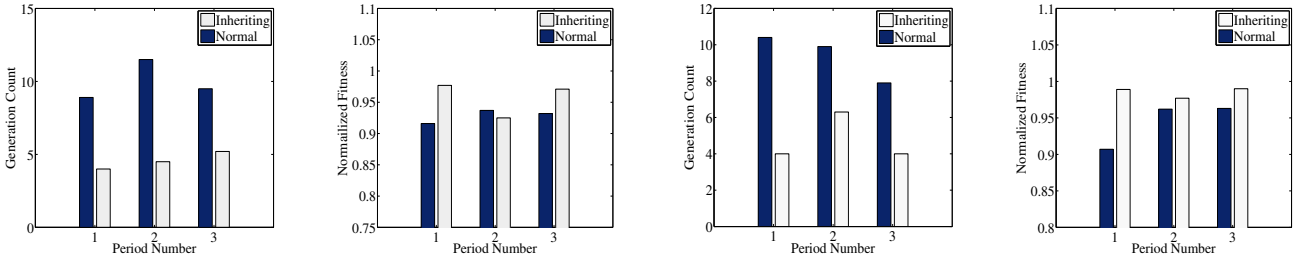
(a) Job completion time of the top 20 racks under input traffic M_1

(b) Job completion time of all the racks under input traffic M_2

(c) Throughput of the top 100 transmissions under input traffic M_1

(d) Throughput of the top 100 transmissions under input traffic M_2

Fig. 7. Performance of all the test cases



(a) Convergence Time

(b) Fitness

(c) Convergence Time (with outburst traffic)

(d) Fitness (with outburst traffic)

Fig. 8. Performance of GA in handling traffic demand evolution

ACKNOWLEDGMENT

This work is supported by NSF of China (60911130511, 60873252), 973 Program of China (2007CB307105, 2011CB302800), and the US NSF grant CNS-0831852.

REFERENCES

[1] S. Arnold, *Google Version 2.0: The Calculating Predator*. Infontorics Ltd, 2007.

[2] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[3] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," in *WREN '09: Proceedings of the 1st ACM workshop on Research on enterprise networking*. New York, NY, USA: ACM, 2009, pp. 65–72.

[4] 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.

[5] L. Caetano and S. Li, *Sibeam Whitepaper: Benefits of 60 GHz*, 2005.

[6] J. P. S. Kandula and P.Bahl, "Flyways to de-congest data center networks," in *HotNets 09: the 8th ACM Workshop on Hot Topics in Networks*, 2009.

[7] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *SIGCOMM '08: Proceedings of the ACM SIGCOMM 2008 conference on Data communication*. New York, NY, USA: 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: Proceedings of the ACM SIGCOMM 2009 conference on Data communication*. New York, NY, USA: 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: Proceedings of the ACM SIGCOMM 2009 conference on Data communication*. New York, NY, USA: 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: Proceedings of the ACM SIGCOMM 2008 conference on Data communication*. New York, NY, USA: 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: Proceedings of the 28th IEEE International Conference on Computer Communications*, 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] J. P. S. Kandula and P.Bahl, "Your data center is a router: The case for reconfigurable optical circuit switched paths," in *HotNets 09: the 8th ACM Workshop on Hot Topics in Networks*, 2009.

[15] A. Zomaya and M. Wright, "Observations on using genetic-algorithms for channel allocation in mobile computing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 13, no. 9, pp. 948–962, 2002.

[16] 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.

[17] Y. Ding, Y. Huang, G. Zeng, and L. Xiao, "Channel assignment with partially overlapping channels in wireless mesh networks," in *WICON '08: Proceedings of the 4th Annual International Conference on Wireless Internet*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 1–9.

[18] G. Brar, D. M. Blough, and P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," in *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2006, pp. 2–13.