Wireless Link Scheduling for Data Center Networks

Yong Cui
Tsinghua University
Beijing, 10084, P.R.China
cuiyong@tsinghua.edu.cn

Hongyi Wang
Tsinghua University
Beijing, 10084, P.R.China
wanghongyi09@mails.tsinghua.edu.cn

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

ABSTRACT

Data Center Networks (DCNs) suffer from the congestions caused by unbalanced traffic distributions. Nevertheless, it is difficult to address such a challenge with the existent Ethernet-based DCN architecture. Wireless networking has been considered as a viable approach with a high potential to tackle the problem due to its flexibility. This paper presents Wireless Link Scheduling for Data Center Networks (WLSDCN) to provide a feasible wireless DCN. More specifically, we propose a wireless DCN architecture as well as a scheduling mechanism. In our wireless scheduling, two different optimization objectives are considered, with one targeting the unbalanced traffic distribution and one maximizing the total network utility, under the constraints of limited wireless resources and co-channel interference. We propose two heuristics for the two problems and carry out extensive simulation study to validate the performance of our design. Our results demonstrate that WLSDCN can significantly improve the performance of DCN.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*

General Terms

Algorithms, Design

Keywords

Data center networks, DCN, wireless DCN, wireless networks, wireless scheduling.

1. INTRODUCTION

In recent years, many data centers have been established to provide services such as search, e-mail, Google File System [7], etc. Such a data center usually consists of thousands of servers forming a data center network (DCN). The key challenge of constructing a DCN is to provide high scalability and network capacity to accommodate a large number of servers as well as to meet the requirements of various applications. For example, in Map-Reduce [5], which is a typical cloud computing application, communications between master nodes and worker nodes usually generate a high volume of traffic throughout the whole network. The DCN to support this type of applications should be able to carry these transmissions efficiently to achieve a high performance.

To tackle this problem, researchers have make a lot of effort on the interconnection architectures and routing mechanisms. Some techniques [1, 15, 8] extend the current tree-based DCN topology by exploiting existent architectures such as the Clos network to achieve scalability and high network capacity. Additionally, new addressing and routing schemes are designed to utilize multiple transmission paths as well as to meet special application requirements such as the migration of the virtual machines. There also exists another category of designs [10, 13, 9] based on server-centric topologies. Instead of adopting hierarchical architectures based on the current DCN, these schemes employ recursive topologies by involving servers in data forwarding. With such a design, the bottleneck at the core layer switches is avoided, and multiple disjointed paths are available to the servers. Moreover, these schemes are optimized for the transmissions of multiple concurrent flows belonging to one server. The latest work [9] achieves load balancing for all-to-all traffic with a decentralized topology. Generally speaking, all the existing solutions improve the performance by providing more paths for data forwarding.

However, servers with high outburst traffic remain the bottleneck in DCNs. These servers usually cause losses on edge links [3] and have negative influence on the global performance. Since the traffic distribution is non-deterministic, it is impossible to address the problem by increasing the bandwidth of a certain group of servers with more links. On the other hand, adding links to all the servers is also inadvisable because of the high cost and the high difficulty in wiring.

Wireless networking has been mentioned as a possible approach [16] because wireless links can be easily established and adjusted among servers. This flexibility makes it much more convenient to extend network capacity for certain servers. Moreover, direct wireless links between servers can be taken as shortcuts to avoid the congestion at core switches.

In fact, the state-of-the-art development of wireless technologies has enabled high data rate transmissions suitable

ICUIMC '11 Seoul, Korea
Copyright 2011 ACM 978-1-4503-0571-6 ...$10.00.

for DCNs. Extremely highly frequency (EHF), which ranges from 30GHz to 300GHz, is a promising technology. In particular, the 60GHz communications has a 7GHz (57-64GHz) wide spectrum band and is able to provide a data rate that is more than 1Gbps [18]. The small wavelength of the radio signals also supports highly directional communications to increase the frequency reuse potential. Although the transmission range of the 60GHz frequency is relatively small (about 10m), it is adequate to support indoor wireless transmissions. In fact, a prototype device of 60GHz communications has already been manufactured [4].

In spite of the availability of the high data rate wireless technology, a number of other issues still need to be handled in order to provide a feasible wireless DCN. First, the requirements of scalability and network capacity should be considered in designing the network architecture. Second, the Ethernet DCN infrastructure and the overlay wireless network need to be carefully coordinated. Third, wireless scheduling is the key issue to determine when and where to establish wireless links.

To handle these problems, we propose Wireless Link Scheduling for Data Center Networks (WLSDCN), which considers various factors such as the traffic distribution, network topology, interference, etc. The major contributions of the paper are listed as follows.

- First, we design a hybrid architecture that integrates the existent Ethernet-based DCNs and wireless networks to take advantage of the high capacity of Ethernet and the high flexibility of wireless networking.

- Second, we present a distributed wireless scheduling mechanism that is able to adapt wireless links to the dynamic traffic demands of the servers. Furthermore, we introduce a novel method to organize the servers to effectively exchange traffic information through the network.

- Third, we formulate two wireless scheduling problems based on different optimization objectives. Both the traffic distributions and the contention of wireless resources are considered in our problem formulation. Additionally, we analyze the complexity and design a heuristic algorithm for each optimization problem.

The rest of the paper is organized as follows. The framework of WLSDCN is presented in Section 2. Section 3 elaborates the modeling of wireless scheduling as well as presents the optimization problems and the corresponding heuristics, whose performances are evaluated in Section 4. Section 5 describes important related work and Section 6 concludes the paper.

2. THE WLSDCN FRAMEWORK

In this section, we depict the framework of WLSDCN. We first introduce the design of our wireless DCN architecture. Then we discuss the requirements of implementing wireless scheduling and detail the procedure of our scheduling mechanism.

2.1 Wireless DCN Architecture

As mentioned before, wireless networks are introduced to alleviate the congestion of servers with high traffic demands. Nevertheless, the capacity of wireless links is limited due to interference and high transmission overhead. Therefore wireless networks should not be employed to substitute Ethernet entirely. Thus we take wireless communications as the supplement to wired transmissions in our wireless DCN architecture.

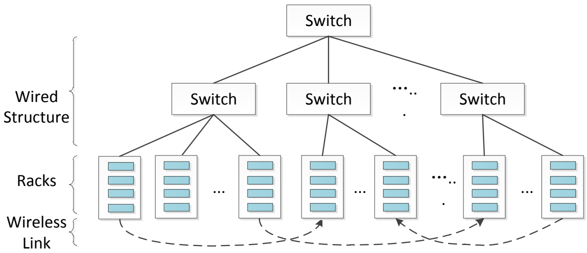


Figure 1: An example wireless DCN architecture, where dashed lines denote wireless links.

To exploit wireless transmissions, servers in a DCN should be equipped with radios. A possible approach is to assign radios to each server. However, this method requires a large number of radios, leading to a high cost and a severe waste of wireless devices since the very limited wireless channel resources can support the concurrent transmissions of only a small fraction of the radios. Therefore, it is more reasonable to allocate radios to groups of servers.

Based on these ideas, our wireless DCN is constructed as follows. First, the servers in the data center are organized into groups. Each group is called a *Wireless Transmission Unit (WTU)*[1]. Then we attach multiple radios to each server group. These radios will be shared by all servers in the same group and the original Ethernet architecture within a group is not changed. Figure 1 illustrates an example wireless DCN based on the current Ethernet data center architecture.

2.2 The Scheduling Mechanism

In addition to the architecture of the wireless DCN, another important module of WLSDCN is the wireless scheduling. Only appropriate arrangement of wireless links can improve the performance of the network effectively. In designing the scheduling mechanism, the following factors should be taken into consideration.

- **Distributed:** Centralized scheduling usually results in a high control overhead, which is prohibitive in wireless DCNs due to the large number of servers. For example, it is quite difficult to exchange information among all the servers, which is required by a centralized controller.

- **Traffic-oriented:** Compared with Ethernet, the capacity of wireless networks is very limited. Since our purpose is to employ wireless links for congestion alleviation, they should be properly scheduled based on the traffic distribution. Intuitively, the WTU with a larger traffic volume should have a higher priority to get extra wireless capacity.

- **Dynamic:** Since the traffic distribution of a wireless DCN is not static, wireless links should also be scheduled dynamically. In other words, the transmissions

[1] WTU is formally defined in Section 3.1.

of wireless links should adapt to the changing traffic distributions.

In order to meet these requirements, we design a distributed scheduling procedure that can be applied periodically to adjust wireless links based on the dynamic traffic distribution. The procedure mainly consists of two steps: first, the traffic demands are exchanged among servers; second, the servers execute our wireless scheduling algorithm, which is described in the Section 3, and perform wireless transmissions according to the output of the algorithm.

To realize the first step, each group takes one of its servers as the group head. The head server is equipped with a *control radio*, which is specialized in transmitting the information of traffic demands. It is also in charge of collecting the traffic demands of the servers inside the group. After that, all the head servers take turns to broadcast the traffic demands of their groups via their control radios on a specified *control channel.* When one head server is broadcasting, other head servers listen to the control channel for traffic information collection. Thus, the traffic distribution of the entire network is learned by all the head servers through the broadcasts over the control channel.

With the collected traffic information, the head servers can execute the scheduling algorithm independently to determine how to establish wireless links. Then they inform other servers in its group of the scheduling and wireless transmissions are carried out accordingly.

Note that each head sever should inform via broadcasts all the other head servers of the traffic demands it has collected. Therefore, traditional wireless technology rather than the EHF communications is suitable. We adopt IEEE 802.11 for exchanging the traffic information. In addition, we assume that the clocks of the servers in the network are synchronized such that the head servers can transmit traffic demands in a polling manner and servers of different groups can cooperate to transmit packets wirelessly.

3. THE SCHEDULING PROBLEM

In this section, we formulate the wireless scheduling problem and propose our solutions. First, we model the network with a digraph and study the constraints of wireless scheduling. Based on the modeling, we define two optimization problems with different objectives and propose heuristic algorithms to tackle them.

3.1 System Model

3.1.1 Wireless Transmissions

As mentioned in Section 2.1, servers in the DCN are organized into groups to perform wireless transmissions. We formalize this concept with Definition 1.

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

In fact, current data centers are mainly constructed based on racks. Therefore intuitively we can take a rack as a WTU. Additionally, many data centers that utilize other interconnection approaches share the feature that servers are organized into groups formed by basic architectures. For example, Fat-tree is constructed based on Pods [1]; the basic architecture of BCube is BCube0, which consists of a switch and several servers [9]. Thus it is reasonable to take the basic architecture as a WTU.

Based on Definition 1, transmissions in a DCN can be classified into two categories: intra-WTU transmissions and inter-WTU transmissions. As servers belonging to the same WTU are usually located in the Layer 2 domain (typically, they connects to the same switch), it is efficient to assign intra-WTU transmissions to Ethernet. Therefore, we focus on accelerating the inter-WTU transmissions with the assistance of wireless links. A directed graph is employed to formalize the distribution of these inter-WTU transmissions.

Definition 2. *A* wireless transmission graph *is a directed graph $G = (V, E)$ in which each node $v \in V$ denotes a WTU and the edge $e = (v_1, v_2) \in E$ denotes the transmission from v_1 to v_2.*

In a wireless transmission graph, each edge e is associated with a weight $t(e)$, which stands for the traffic demand of the corresponding transmission. Let $T(E) = \{t(e)|e \in E\}$.

Note that we employ the wireless transmission graph to model the potential wireless transmissions in a DCN. Ideally, all the possible inter-WTU transmissions should be included in the graph. However, the layout of a real data center makes it impossible as either the distance of two WTUs is too long or the existence of obstacles block the wireless connections between two WTUs. This problem is in particular likely to happen in a multi-story building. In our modeling, such transmissions are excluded from the graph. In other words, if direct wireless connections can not be established between v_1 and v_2, edge (v_1, v_2) is removed from the wireless transmission graph.

3.1.2 Wireless Links and Interference

In order to carry out wireless transmissions, wireless links are established between WTUs. In this work, we assume that the link from v_1 to v_2 only transmits the traffic from v_1 to v_2. In other words, wireless links are attached to the edges of the wireless transmission graph.

We also assume that it is possible for multiple wireless links to conduct cooperative transmissions for one edge so that the traffic between a pair a WTUs can be assigned to multiple links. Obviously, more wireless links for a transmission leads to a higher throughput. However, the number of wireless links is limited by the finite number of radios and channels.

A sender radio and a receiver radio can establish a wireless link and a radio cannot support multiple links simultaneously. Therefore, the number of wireless links belonging to a WTU v should not exceed the number of the radios of v, denoted by $r(v)$.

The transmission of a wireless link causes interference on other links and prevent the interfered links from utilizing the same channel. Whether a link interferes with another link is determined by the physical location of the endpoints of the corresponding transmission. We adopt a conflict-edge model to formalize the interference relationship among transmissions. In this model, each edge e is associated with a conflict edge set $I(e)$, in which all the edge interferes with e. Therefore, a wireless link on channel c is available to e if and only if no edge in $I(e)$ has an active link on c.

The following geometric models are commonly used to determine the conflict edge set of a network [2].

- **Node-exclusive model**: Edges sharing a common endpoint interfere with each other [14].

- **Unit disk model**: Each node has an interference range. Edge $e = (v_1, v_2)$ is in the conflict-edge set of $e' = (v'_1, v'_2)$ if v'_1 or v'_2 is in the interference range of v_1 or v_2 [11].
- **K-hop model**: Two edges interference with each other if the shortest path between their endpoints is equal to or less than K hops [17].

In this work, we employ EHF communications for the data transmissions in a wireless DCN. Since EHF communications are highly directional, the unit disk model is not appropriate. As we use wireless one-hop communications instead of multi-hop transmissions, the K-hop Model is unsuitable. Therefore, we adopt the node-exclusive model.

3.1.3 Channel Allocation

In this work, we assume that the wireless channels are orthogonal and thus wireless links on different channels do not interfere with each other. Let C be the set of available channels. In a wireless transmission graph, each edge e is associated with a subset of C, denoted by C_e, which is the set of channels assigned to e for wireless transmissions. In other words, if $e = (v_1, v_2)$ and $c \in C_e$, there is a wireless link from v_1 to v_2 on channel c.

We define the collection of C_e as a channel allocation scheme, which can be expressed by a two-dimension matrix S that can be expressed by (1), where $S(e, c)$ denotes whether or not a link on channel c is set up for e.

$$S(e,c) = \begin{cases} 1 & \text{if } c \in C_e, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

3.1.4 Transmission Utility

To improve the performance of the network, it is necessary to assign the limited channel resources to the edges that contribute more to the global performance. Thus, a metric is required to evaluate the contribution of each edge. Based on the design requirements of WLSDCN, the following factors are considered in the metric.

First, the traffic of a transmission should be taken into account. In our periodical scheduling mechanism, established wireless links occupy the channels during the whole period whether they are active or not. Therefore it is not reasonable to assign wireless links to the edge that carries a low traffic since the idle period caused by the early completion of the wireless transmissions leads to the waste of the channel resources.

Second, the distance between the source and the destination of a transmission is another significant factor. The flow with a longer wired path usually incurs a larger transmission latency and therefore aggravates the load of higher layer switches. Transmitting these flows over wireless links is obviously more beneficial to enhance the global performance.

Considering these issues, we employ Definition 3 to measure the contribution of a transmission to the global performance.

DEFINITION 3. *The* utility *of a transmission e is the product of the distance factor of e and the total traffic sent by the wireless links of e in a period.*

Let $u(e)$ be the utility of e, $d(e)$ be the distance factor, and $\Delta t(e)$ the total traffic sent by the wireless links of e in a period. The utility of e can be expressed as (2).

$$u(e) = d(e) \cdot \Delta t(e) \tag{2}$$

In this work, we take the hop count in the wired network between the source and the destination as the distance factor $d(e)$. $\Delta t(e)$ is determined by the traffic of e and the number of wireless links attached to e, which is equal to $|C_e|$. We also assume that all the wireless links have the same data rate and let Δt_0 denote the maximum traffic that a wireless link can transmit in a period. Thus, $\Delta t(e)$ can be computed based on (3).

$$\Delta t(e) = \min\{t(e), |C_e|\Delta t_0\} \tag{3}$$

Let E_v^s denote the set of transmissions that take v as the source and E_v^d denote the set of transmissions that take v as the destination. Based on Definition 3, we derive Definition 4.

DEFINITION 4. *The node utility of WTU v is the sum of the product of the traffic and the distance factor of all the transmissions in E_v^s, i.e.,*

$$\hat{u}(v) = \sum_{e \in E_v^s} t(e)d(e) \tag{4}$$

Definition 4 is introduced to estimate whether a WTU is hot. The WTUs with high node utility are considered as hot nodes.

3.2 Min-Max Scheduling

3.2.1 The Min-Max Optimization Problem

Based on Definition 4, we can design a wireless scheduling algorithm to resolve the congestions incurred by hot WTUs. A feasible approach is to always assign wireless links to the currently hottest node. In other words, the objective of the scheduling is to minimize the maximum total utility. This approach is defined as *Min-Max-Scheduling* (*MM-Scheduling* for short) and the corresponding optimization problem is formulated as (5).

$$\min(\max_{v \in V}(\hat{u}(v) - \sum_{e \in E_v^s} u(e))) \tag{5}$$

subject to

$$\sum_{e \in E_v} S(e,c) \le 1 \qquad \forall v \in V, \forall c \in C$$

$$\sum_{c \in C} \sum_{e \in E_v} S(e,c) \le r(v) \qquad \forall v \in V$$

In (5), the objective is to minimize the maximum remaining node utility after a transmission period. The first constraint ensures that no interference occurs and the second one makes the number of active links belonging to a WTU no more than the number of radios of the WTU. The wireless links fulfilling these constraints are free from interference and thus can transmit data simultaneously.

According to (2), there is a linear relationship between $u(e)$ and $\sum_{c \in C} S(e, c)$. Therefore, the problem can be converted to (6) by introducing an additional variable x. Obviously, problem (6) is an Integer Linear Programming, which is a well-known NP-hard problem.

$$\min x \tag{6}$$

subject to

$$\sum_{e\in E_v} S(e,c) \le 1 \qquad \forall v \in V, \forall c \in C$$

$$\sum_{c\in C}\sum_{e\in E_v} S(e,c) \le r(v) \qquad \forall v \in V$$

$$\hat{u}(v) - \sum_{e\in E_v^s} u(e) \le x \qquad \forall v \in V$$

3.2.2 The MM-Scheduling Algorithm

We design a greedy algorithm to tackle (5). To proceed, we need the following definitions.

Definition 5. *A node $v \in V$ is a* saturated node *(*SN*) if $\sum_{c\in C}\sum_{e\in E_v} S(e,c) \ge r(v)$.*

Definition 6. *An edge $e = (v_1, v_2) \in E$ is a* saturated edge *(*SE*) if $\sum_{e\in E_{v_1}\cup E_{v_2}} S(e,c) \ge 1$ for any $c \in C$.*

Algorithm 1 outlines our approach of Min-Max Scheduling. At each iteration, we pick up the hottest pending WTU v from V_p, which denotes the set of pending nodes. If either v turns out to be a SN or all its transmissions have become SEs, no more wireless links can be attached to v; if all its transmissions have no remaining traffic, there is no need to add wireless links to v. For both situations, v is considered as a scheduled node and is removed from V_p. Otherwise, if it is still possible and necessary to add wireless links to v, an appropriate edge of v and an available channel are taken to establish a new wireless link for v. The remaining traffic of the transmission is decreased accordingly. If there is no pending node, the algorithm terminates and the resultant matrix S is the channel allocation scheme of our MM-Scheduling.

Algorithm 1 (MM-Scheduling).

```
Input: $G = (V, E)$, $C$, $T(E)$
Output: $S$
$S \leftarrow 0$
$V_p \leftarrow V$
while $V_p \ne \varnothing$ do
    $v \leftarrow \arg\max_{\bar{v}\in V} \hat{u}(\bar{v})$
    if $v$ is a SN OR ($\forall e \in E_v^s$, $t(e) = 0$ OR $e$ is a SE) then
        $V_p \leftarrow V_p - v$
    else
        $\bar{e} = (v_1, v_2) \leftarrow$ a randomly selected element in $\{e | e \in E_v^s \wedge t(e) \ge 0 \wedge e$ is not a SE$\}$
        $\bar{c} \leftarrow$ a randomly selected element in $\{c | c \in C \wedge \sum_{e\in E_{v_1}\cup E_{v_2}} S(e,c) = 0\}$
        $S(\bar{e}, \bar{c}) \leftarrow 1$
        $t(\bar{e}) \leftarrow \max\{0, t(\bar{e}) - \Delta t_0\}$
    end if
end while
return $S$
```

The Complexity of MM-Scheduling. The complexity of Algorithm 1 depends on the number of iterations. Although the execution time of each iteration is not deterministic, we can estimate its upper bound. At each iteration, we add no more than one wireless link. Assume that each WTU is equipped with r radios. Therefore we can establish $|V| \cdot \min\{r/2, |C|\}$ links at most. If we do not set up a link during an iteration, a node is removed from the pending node set. This removal can be executed no more than $|V|$ times since we have at most $|V|$ pending WTUs. In summary, the loop is executed at most $O(|V|\min\{r/2, |C|\})$ times, and at each iteration, it takes $O(|V|)$ time to find the hottest node and determining whether a new link can be added requires $O(|V||C|)$ time to traverse all the nodes and channels.

Therefore the time complexity of Algorithm 1 is $O(|V|^2|C| \cdot \min\{r/2, |C|\})$. Since $|C|$ is usually much greater than r, the complexity can be simplified as $O(r|C||V|^2)$.

3.3 Best-Effort Scheduling

3.3.1 The Best-Effort Optimization Problem

In addition to MM Scheduling, we provide another possible approach to maximizing the total utility of all the edges. This approach is denoted as Best-Effort (BE) Scheduling and the corresponding optimization problem is defined in (7). The objective of (7) is to maximize the sum of the utilities of all the edges while the constraints are the same as those of (5).

$$\max \sum_{e\in E} u(e) \tag{7}$$

subject to

$$\sum_{e\in E_v} S(e,c) \le 1 \qquad \forall v \in V, \forall c \in C$$

$$\sum_{c\in C}\sum_{e\in E_v} S(e,c) \le r(v) \qquad \forall v \in V$$

3.3.2 Best-Effort Algorithm

It is obvious that (7) is also NP-hard. We design a heuristic algorithm to tackle (7) based on the Hungarian algorithm [12] for maximum weighted matching in Graph Theory.

Algorithm 2 (BE-Scheduling).

```
Input: $G = (V, E)$, $C$, $T(E)$
Output: $S$
$S \leftarrow 0$
Initialize $U$ based on (2)
while $U \ne 0$ do
    $E_p \leftarrow$ MaximumWeightedMatching($U$)
    for $e = (v_1, v_2) \in E_p$ do
        if $U(v_1, v_2) > 0$ then
            $c \leftarrow$ a randomly selected element in $\{c | c \in C \wedge \sum_{e\in E_{v_1}\cup E_{v_2}} S(e,c) = 0\}$
            $S(\bar{e}, \bar{c}) \leftarrow 1$
            $t(\bar{e}) \leftarrow \max\{0, t(\bar{e}) - \Delta t_0\}$
        end if
    end for
    Update $U$ based on (2)
    $U(v_1, v_2) = 0$ if $(v_1, v_2)$ is a SE, $\forall (v_1, v_2) \in E_p$
    Set the column and the row of $v$ in $U$ to 0 if $v$ is a SN, $\forall v \in V$
end while
return $S$
```

In BE-Scheduling, we define another two-dimension matrix U, in which an element $U(v_1, v_2)$ denotes the utility of the edge $e = (v_1, v_2)$. Initially, U is computed based on (2) (if $(v_1, v_2) \notin E$, $u(v_1, v_2)$ is set to 0). At each iteration, we perform the Hungarian algorithm on U to compute the maximum weighted matching. For each node pair (v_1, v_2) in the matching, if $U(v_1, v_2)$ is not 0, a wireless link on an available channel is attached to (v_1, v_2) and $t((v_1, v_2))$ is decreased accordingly. After establishing a link, U should be updated too. When updating U from one iteration to another, we have to consider the constraints in (5): if (v_1, v_2) becomes a SE, $U(v_1, v_2)$ should be set to 0; if a node v becomes a SN, all the elements relevant to v (i.e., the elements either in the same row or in the same column of $U(v, v)$) should be set to 0. The algorithm terminates if all the elements of U become 0, which indicates that either there is no remaining traffic or no wireless links can be established due to interference or the limit of radios.

The Complexity of SE-Scheduling. Similar to the analysis of MM-Scheduling, we can estimate the upper bound of the execution time of each iteration in Algorithm 2. As we can add no more than $|V|$ wireless links at each iteration, in total we can add $O(|V| \cdot \min\{r/2, |C|\})$ links. Furthermore, the weighted maximum matching with the Hungarian algorithm takes $O(|V|^3)$ time. Since setting up a link for an edge in E_p takes $O(|V||C|)$ time, it takes $O(|V|^2|C|)$ time to process the edges in E_p. Similarly, updating U requires $O(|V|^2|C|)$ time.

In conclusion, the time complexity of our SE-Scheduling is $O(\min\{r/2, |C|\}|V|^2 \max\{|V|, |C|\})$. Since $|C|$ is usually much larger than r and $|V|$ is much greater than $|C|$, the complexity can be simplified as $O(r|C||V|^3)$.

4. EVALUATION

In this section, we first introduce the methodologies of our evaluation. Then the simulation results are reported and analyzed.

4.1 Simulation Setup and Network Settings

In order to validate WLSDCN, we perform a series of simulations in a simulator implemented with C++. The simulated DCN has a rack-based tree topology, where servers are grouped into 64 racks and racks are connected together via two layers of switches as shown in Figure 2. Note that we adopt this topology because most current data centers are built based on the tree topology. The simulator takes two different traffic distributions as its input: one is a hotspot distribution in which 10% of the racks contribute 90% of the total traffic; the other is a uniform distribution where each server exchanges the same amount of data with other servers (all-to-all traffc).

In this simulation study, we evaluate the performance of both MM-Scheduling and BE-Scheduling. We also evaluate the effectiveness of our utility based optimizations by comparing the results with those of the scheduling that only considers throughput (In other words, the distance factors of all the edges are treated as 1). Furthermore, we take the performance of random scheduling as the comparison base.

The metric of the experiments is the transmission time of the given input traffic. However, it does not make sense to compare the transmission time of different traffic distributions. Therefore, we employ a *normalized transmission time* (NTT), which is the ratio of the transmission time of the network with scheduled wireless links and the one with wired architecture alone (i.e., no wireless links are established), to measure the improvement in performance for different traffic distributions.

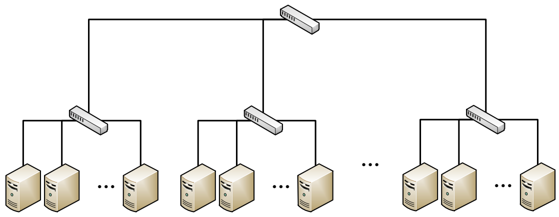


Figure 2: The DCN in our simulation study.

Table 1: Parameter Settings in the Simulations

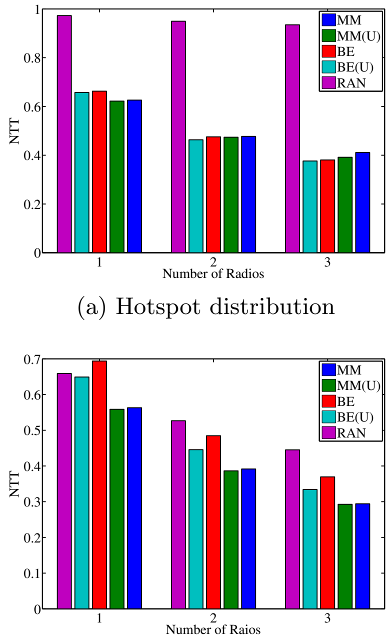
	Number of Radios	Number of Links	Bandwidth (Mbps)
Default	3	48	100
Range	$1 \sim 3$	$10 \sim 100$	$100 \sim 1000$

The impacts of multiple factors are considered in our simulation study, including the number of radios, the total number of wireless links, and the bandwidth of a wireless link. The number of radios determines the maximum number of wireless links that can be attached to a WTU. We also gradually add links based on our scheduling algorithms to investigate how many wireless links are adequate to achieve a considerable improvement in performance. On the other hand, the bandwidth of a wireless link, which has an impact on Δt_0, is also an important factor. We test wireless links with different bandwidths to evaluate whether WLSDCN is still effective if the data rate of wireless links is not as high as that of wired links. Note that we do not take the number of channels into consideration because we adopt the node-exclusive model to formulate the interference, in which the limit of radios is usually a more strict constraint than that of the channels. The default values and the range of the values of the parameters are listed in Table 1.

4.2 Simulation Results

Impact of the Number of Radios. In this experiment, we investigate the impact of the number of radios on each rack. Figure 3 shows the NTT of different algorithms, in which the mark U in the brackets indicates that the algorithm considers utility and RAN stands for random scheduling.

In general, our approach makes significant contributions in terms of transmission time reduction and this effect grows as the number of radios increases. For the hotspot distribution, our scheduling algorithms have a distinct advantage as random scheduling provides little help for the nodes with high volumes of traffic. On the other hand, the distinction among different approaches is not so obvious under the uniform distribution because all the racks bear a similar amount of traffic. In such a condition, randomly adding wireless links is also effective to some degree. Furthermore, the NTT of the uniform distribution is lower than that of the hotspot



(a) Hotspot distribution

(b) Uniform distribution

Figure 3: NTT *vs.* number of radios

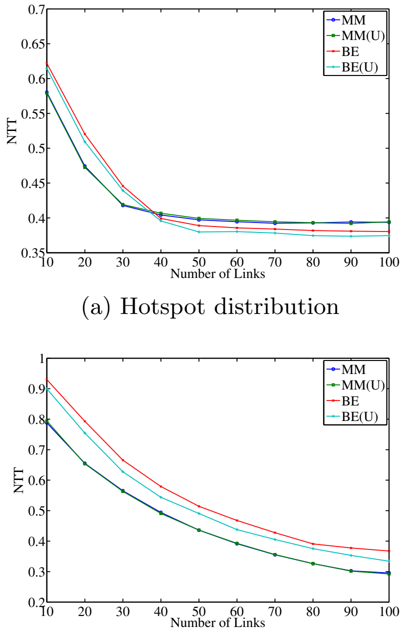
distribution because a lot of radios become idle after the corresponding racks complete all their transmissions in the hotspot distribution, which leads to a lower utilization of the radios.

Impact of the Number of Wireless Links. The NTT over different numbers of wireless links is illustrated in Figure 4. We no longer report the results of random scheduling because they are much higher than those of our scheduling algorithms as shown in Figure 3, especially under the hotspot distribution. It is obvious that the improvement in performance increases with the increase of the number of wireless links.

According to the results, the network with only 30-40 additional wireless links acquires almost the same performance as the one with 100 wireless links in hotspot distribution. This is because 30-40 wireless links are enough to support all the saturated hot nodes. On the other hand, uniform distribution does not demonstrate this feature.

This experiment is different from the previous one, which focuses on the impact of the number of radios. The maximum number of wireless links that can be added to a single rack has an impact on the performance, especially when the total number of links is limited. As a result, MM-Scheduling obtains a shorter NTT than BE-Scheduling if the total number of links is below 30 as shown in Figure 4(a), which is different from the results shown in Figure 3.

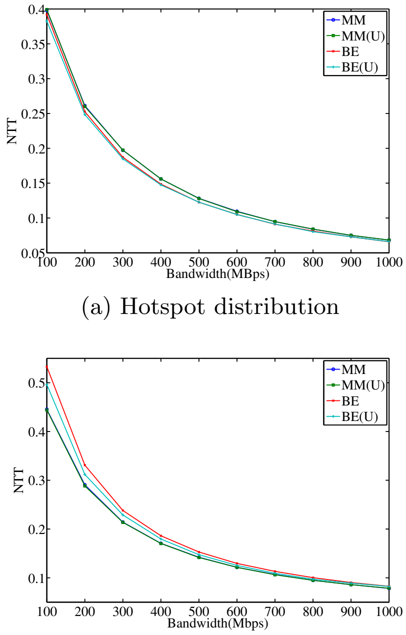
Impact of the Bandwidth of Wireless Links. The NTT over different bandwidths is shown in Figure 5, in which the bandwidth of wireless links ranges from 100Mbps to 1000Mbps while the bandwidth of wired links is a constant 1000Mbps. It can be seen that the transmission time decreases with the increase of the bandwidth of wireless links.



(a) Hotspot distribution

(b) Uniform distribution

Figure 4: NTT *vs.* number of links



(a) Hotspot distribution

(b) Uniform distribution

Figure 5: NTT *vs.* bandwidth

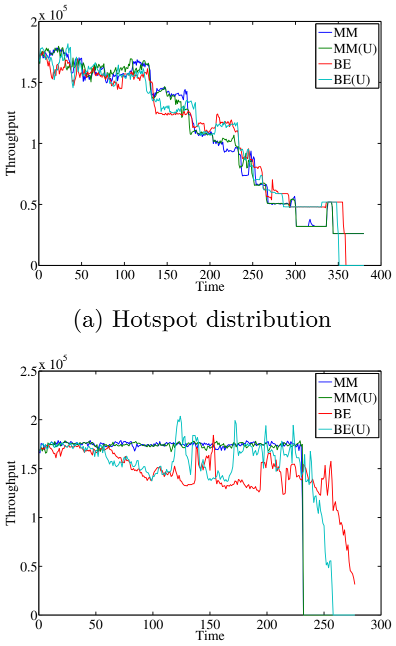
However, the relationship between the transmission time and the bandwidth of the wireless links is not linear. Instead, the network enjoys a significant improvement by adding wireless links with a limited bandwidth.

Impact of Utility. It seems that considering utility rather than the throughput makes only a little contribution to

NTT. There are several reasons for this phenomenon. First, in the hotspot distribution, the traffic of the hot nodes is much larger than that of other racks. Therefore, the difference between the hop counts of different transmissions is negligible. Second, the topology only has three layers and the majority of the racks are far from each other (i.e., the hop count between two racks is 4). Therefore the distance factors of most edges are the same.

Comparison of the Two Approaches. For the hotspot distribution, the two algorithms achieve similar performance, which contradicts to the intuition that MM-Scheduling is optimized for hotspot distributions. In fact, under this distribution, both scheduling approaches would attach wireless links to the currently hottest nodes after other nodes finish their transmissions. Thus, the transmission time mainly depends on the traffic of the hot nodes. This observation is demonstrated by Figure 6(a), in which the throughputs of different approaches all drop gradually except for the few steep declines caused by the completion of the transmissions belonging to the hot nodes. Because the maximum number of wireless links attached to the hot nodes is independent of the scheduling mechanism, MM-Scheduling and BE-Scheduling result in similar transmission times.

For the uniform distribution, although BE-Scheduling maximizes the utilization of wireless links, it cannot balance the remaining traffic among the racks. As a result, a few racks accumulate a large volume of traffic and turn out to be hot nodes. These hot racks become the bottleneck as the number of wireless links attached to a node is limited. This phenomenon is illustrated by Figure 6(b), in which the throughput of BE-Scheduling experiences a series of steep declines which are similar to that of Figure 6(a). On the other hand, MM-Scheduling does not suffer from this problem. Thus it outperforms BE-Scheduling.



(a) Hotspot distribution

(b) Uniform distribution

Figure 6: Throughput *vs.* Time

5. RELATED WORK

Researchers have made a lot of effort on the interconnection architectures and routing of DCNs. Major results can be divided into two categories. One is to extend the existing tree-based architecture. Fat-tree [1], for example, groups servers into pods and establishes multiple paths between the core layer and the aggregation layer. Leveraging on the Fat-tree architecture, Portland [15] provides a scalable fault-tolerant DCN that supports virtual machine migration. VL2 [8], on the other hand, constructs a Clos Network, based on which new addressing and routing mechanisms are proposed to provide high capacity and performance isolation among different services. In these architectures, additional paths are provided for the transmissions between different branches of the tree such that there is sufficient bandwidth to forward the incoming traffic. The performance of DCNs is improved at the cost of switch upgrades and more hardwares, including more switches and wires.

The other category is to construct a recursive topology by involving servers in data forwarding. DCell [10] takes a switch and several servers as a basic unit and constructs high level topologies recursively by directly connecting servers of different units together. FiConn [13] is an extension of DCell for servers with only two ports. BCube [9] is proposed for modular DCN. Different from DCell, the basic units of BCube are linked together via a switch rather than direct links between servers, thus avoiding the bottleneck in transmitting all-to-all traffic and achieving a graceful performance degradation. In these approaches, servers play a dominant role and therefore, bottleneck at higher layer switches does not occur. Moreover, distributing traffic to several available paths prevents the bottleneck on the path with a heavy load. However, these approaches are also questionable owing to the complexity of wiring and the fact that the data forwarding efficiency of the servers is not as high as that of switches.

In addition to the work on Ethernet based architectures, research on utilizing other technologies is also carried out. Flyway [16] proposes to introduce wireless communications to DCN. It is motivated by the following observation: at any instant of time, only a few racks in a DCN have a large amount of data to transmit; therefore providing extra links to increase the capacity of these racks can enhance the performance considerably. In order to set up the extra links between any pair of racks, ToRs in [16] are equipped with 60GHz antennae so that extra wireless links can be arbitrarily established between different racks. Besides, [16] also considers how to place flyways appropriately. According to the simulation results, a simple flyway based tree topology can yield a great improvement without constructing a complex topology consisting of a lot of switches and wires. Nevertheless, as an initial study of utilizing wireless in DCN, [16] does not consider the wireless interference and the resource constraints such as the limited number of radios. Helios [6], on the other hand, is a hybrid electrical/optical solution for DCNs. With the help of optical switches, Helios adds optical circuits between servers to provide a high network capacity. Although employing different techniques, Flyway and Helios share the same feature that both require the underlying Ethernet architecture and other transmission technologies integrate seamlessly for better performance.

6. CONCLUSIONS

In this paper, we propose WLSDCN to utilize wireless transmissions in DCNs. Both the network architecture and the scheduling mechanism are designed to provide an effective wireless DCN. Based on the modeling of the transmissions in wireless DCNs, we investigate two different scheduling objectives to obtain two different problem formulations and provide a scheduling heuristic for each of them. More specifically, the Min-Max-Scheduling focuses on the challenge of unbalanced traffic distributions and our heuristic intends to serve the hot WTUs in a greedy manner. On the other hand, the Best-Effort-Scheduling aims at maximizing the utilization of wireless resources and the proposed heuristic is based on the maximum weight matching over a utility matrix. Simulation study is performed to evaluate the two scheduling algorithms. Compared with random scheduling, our approaches achieve a significant improvement. Moreover, we notice that wireless links make great contributions even if their bandwidths are not as high as those of wired ones; a limited number of wireless links is enough to maximize the performance for a typical unbalanced traffic distribution. Our simulation results also indicate that both scheduling mechanisms are effective even if serious interference limits the number of concurrent wireless transmissions. As part of our future work, we plan to study the joint optimization of wireless scheduling and multi-path routing to achieve a better utilization of wireless resources in wireless DCNs. Furthermore, small-scale experiments will be carried out to investigate various challenges in real systems.

7. ACKNOWLEDGMENTS

This work is supported by the US National Science Foundation grant CNS-0831852, the 973 Program of China (No. 2009CB320503 and 2011CB302800), and the National Natural Science Foundation of China (No. 60873252 and No. 60911130511).

8. REFERENCES

[1] 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*, pages 63–74, New York, NY, USA, 2008. ACM.

[2] M. M. S. P. B. Han, V.S. Kumar and A. Srinivasan. Distributed strategies for channel allocation and scheduling in software-defined radio networks. In *INFOCOM '09: Proceedings of the 28th IEEE International Conference on Computer Communications*, 2009.

[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*, pages 65–72, New York, NY, USA, 2009. ACM.

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

[5] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008.

[6] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat. Helios: a hybrid electrical/optical switch architecture for modular data centers. In *SIGCOMM '10: Proceedings of the ACM SIGCOMM 2010 conference on SIGCOMM*, pages 339–350, New York, NY, USA, 2010. ACM.

[7] S. Ghemawat, H. Gobioff, and S.-T. Leung. The google file system. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 29–43, New York, NY, USA, 2003. ACM.

[8] 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*, pages 51–62, New York, NY, USA, 2009. ACM.

[9] 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*, 39(4):63–74, 2009.

[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*, pages 75–86, New York, NY, USA, 2008. ACM.

[11] P. Gupta and P. R. Kumar. The capacity of wireless network. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[12] H. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[13] 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.

[14] X. Lin and S. Rasool. A distributed and provably efficient joint channel assignment, scheduling and routing algorithm for multi-channel multi-radio wireless mesh network. In *INFOCOM '07: Proceedings of the 26th IEEE International Conference on Computer Communications*, 2007.

[15] 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*, pages 39–50, New York, NY, USA, 2009. ACM.

[16] 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.

[17] G. Sharma, R. R. Mazumdar, and N. B. Shroff. On the complexity of scheduling in wireless networks. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 227–238, New York, NY, USA, 2006. ACM.

[18] P. Smulders. Exploiting the 60ghz band for local wireless multimedia access: Prospects and future directions. *IEEE Communications Magazine*, 40(1):140–147, 2002.