OpenFlow in Wireless-Associated Data Center Network

Jiahao Zhang
Department of Electronic Engineering
Shanghai Jiaotong University
Email: zjh08177@126.com

Qinqi Huang
Department of Electronic Engineering
Shanghai Jiaotong University
Email: hqq0126@hotmail.com

***Abstract*—Today's data centers offer tremendous aggregate bandwidth to clusters of tens of thousands of machines. The network architecture typically consists of a tree of routing and switching elements with progressively more specialized and expensive equipment moving up the network hierarchy. However, since the rapid development of information technology, the demand of information exchange and storage is also increased in an amazing speed, bring new challenges to current data centers.**

In this paper, we analyze the current infrastructure of data center network and point out several fatal drawbacks according to the nowadays's increasing transmission demands and more complicated situations. To resolve these problem. We modify the traditional data center network with some additional wireless links called flyways to enlarge the transmission capacity. We use Openflow methods to realize the control of wireless-associated DCN. A control protocol is provided to find out best strategies to handle every corresponding conditions.

I. INTRODUCTION

Nowadays, Data Center is becoming increasingly essential in today's world, providing services such as network search, e-mail, and Cloud services like Google File System[1], etc. Hundreds of racks each contain several servers form a data center network(DCN). The goal of a DCN design is to achieve high data rate with satisfactory scalability and a huge capacity to meet the requirements of various applications. For example, WeChat, which is a popular live chat application in smartphones, allowing millions of users to send messages, voice or even video records to others. The DCN constructed to support this type of application should be able to transmit these huge amounts of packets fast with lowest errors. How to avoid blocks in hot nodes is the main problem in today's DCN. Solutions must be taken to get a high performance.

To overcome these challenges, researchers have made many efforts on both the topological structures as well as the scheduling protocol of DCN. In some works, researchers tackles this problem by combining many more links and switches with variants of multipath routing so that the core of the network is no longer oversubscribed. [6], [7], [8]. The network performance of these designs is undoubtedly improved in some extent, but this benefit comes with large material cost and implementation complexity[9]. The method of maintaining and upgrading these data centers is very hard since some designs require so many wires that cabling becomes a challenge. There exists several techniques which extend the traditional tree-based DCN topology by exploiting existent architectures to achieve higher scalability and capacity. In reference [2], [3], [4], several some tree-based structures are introduced to enlarge the channel capacity of the edge layers.

In reference [5], the author provide new-concept switches which can be reconstructed according to various situations to avoid congestions. Bottleneck at the core layer switches is avoided with this structure. However, the switches need to be handled every time the hot spots' distributions are changed so it can't be applied well in business.

The common problem that the above researches faced is that the uncertain conditions of various transmissions. Severs with high outburst will become hot nodes in DCN. They may cause blocks and data losses on the links connected to this node, which has a negative influence on the performance of application. It is impossible and unrealistic to lighten the situation by increasing the number of severs and capacity.

Wireless connections such as Flyways can release the hot nodes [10], [11]. In normal circumstance, hot node is always isolated, and other nodes still have spare ability to receive or transmit packets. So wireless connections can be added between hot nodes and other nodes. It is simple to set up wireless connections in current DCN structure. The increase of bandwidth and capacity becomes convenient.

However, there still exists some problems about wireless-associated DCN. How to schedule the wireless communication flexibly is complex in current network structure. The coordination between wireless and wired links in DCN also need to be carefully considered. Soft Designed Network(SDN)[12], [13] has been mentioned as a possible approach to meet the various demand of applications without modification of devices. [14], [15], [16], [17], [18]In this paper, we propose a Wireless-Associated Data Center Network controlled by Openflow. We use wireless communications based on 60GHz technology to lighten the hot nodes in DCN and use Openflow to control. We also provide a schedule which considers various factors such as the traffic load distribution, radio range and interference, etc. The major contributions of the paper are as follows.

- First, we redesign the DCN architecture with a hybrid structure of wired- and wireless network. It is based on the current Ethernet-based DCN. This hybrid DCN is able to work with both the high capacity and data rate of wired network and the high flexibility of wireless network.
- Second, we present a schedule which is carried on Openflow controller to coordinate the wired- and wireless network with the consideration of factors mentioned above.

It is able to organize a normal DCN to exchange traffic information efficiently.

- Third, we simulate the model and optimize the schedule in various situations to achieve a better performance. We are still doing this part of work.

The rest of the paper is organized as follows. The frame of Wireless-Associated DCN is presented in Section II. Section III elaborates how Openflow is added to Wireless-Associated DCN. Section IV introduces our schedule and presents an algorithm to achieve better performance. Section V describes the optimization and simulation in future work. Section VI concludes the paper.

II. WIRELESS-ASSOCIATED DCN FRAMEWORK

In this section, we depict the framework of Wireless-Associated DCN. First we introduce the structure and the object of DCN. Then we introduce the design of our Wireless-Associated DCN architecture.

A. Basic cognation of data center network

As we mentioned before, Data Center plays an important role in current network structure. DCN is based on computer systems and other associated components such as telecommunications and storage systems. The basic unit in DCN is a rack. In a rack, several servers are set and connected with each other. Among racks, communications can be established and through switches DCN is able to be connected to other network such as Internet.

With the demand of applications, DCN must have high capacity and wide bandwidth. Thus current DCNs are usually built based on the gigabit Ethernet, whose capacity is highly beyond that of wireless connections. Our design is based on the current DCN structure. Thus we only add the wireless devices and Openflow controllers. The switches should be replaced as Openflow switches correspondingly, which we will introduce in detail as follow.

B. Wireless DCN architecture

With the state-of-art development of wireless technology, wireless communications can reach a high data rate which can meet the demand of DCN. 60GHz technology is a possible way. 60GHz devices work on extremely highly frequency(EHF), which gives it 7GHz(57-64GHz) wide bandwidth and the data rate could be more than 1 Gbps. Due to the small wavelength, 60GHz wireless communication can support highly directional communications, which means in the area of DCN facility house the frequency reuse ratio can be increased. Although the range of 60GHz wireless communication is relatively small(almost 10m) and may cause high interferences, it can be addressed by modification of device locations, and with only several 60GHz devices, the whole area of DCN facility house can be included in the wireless communication range.

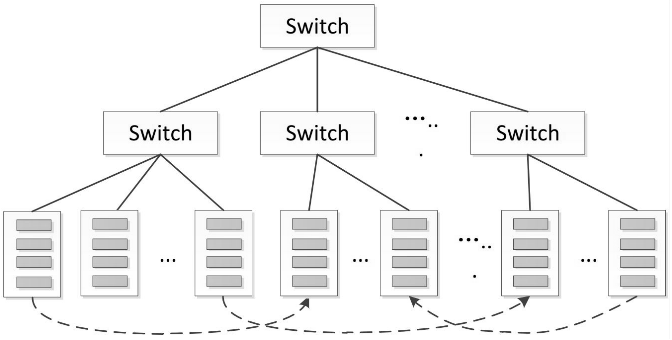

Fig. 1: Wireless-Associated DCN architecture

In our design, wireless networks are used to alleviate the heavy traffic load in DCN, i.g. hot node issue. Due to the limited capacity caused by interference, wireless networks should not be the only way to transmit information. On the other hand, in normal situation, hot nodes always occur randomly, which means at any time only a few nodes will be the hot nodes. Therefore the amount of wireless devices is limited and it must be enough for the worst case DCN will face.

Based on these ideas, we can construct our DCN as follow. First, several racks are organized into a group as a unit. Each unit is called $Wireless\ Transmission\ Unit(WTU)$. We add a set of wireless devices on each WTU, which can be shared by all racks in this WTU. In WTU, racks could communicate with each other by wired network. And among WTUs, racks in different WTUs are able to make connections by both wired- and wireless links. Fig. 1 illustrates our Wireless-Associated DCN

C. Scheduling Mechanism

In spite of the increasing capacity, we still faces serval issues to provide a feasible wireless-associated DCN. First, situation of all nodes must be sensed and carefully considered when controller make decision to add wireless links. Second, the range and interference should be taken into consideration to coordinate wireless network and wired network. Third, how to realize the schedule and make the performance better.

To handle these problems, we present our schedule mechanism. Only with the appropriate arrangement of wireless links, the hot node issue will be solved. In generally, the key to lighten the traffic load is to set up more paths at right places. In designing the scheduling mechanism, factors as follows must be taken into consideration.

- **Distributed:** Centralized scheduling may result a high control overhead. In Wireless-Associated DCN this issue is more serious. Thus the model must have multiple controllers structure.
- **Oriented:** Because of the limited capacity of wireless network, wireless links must be set over the hottest nodes, which requires a sensor layer and a controller layer.
- **Dynamic:** Since the traffic distribution of a DCN is not static, the wireless schedule should be dynamic to meet the demands, which means, with the help of sensor layer, controllers must make decisions to set up wireless links dynamically.

To meet requirements above, we design a distributed scheduling procedure to adjust wireless links allocation based on the dynamic traffic distribution. First, racks in one WTU exchange the traffic information and inform it to the controller. Second, controllers collect the information and execute our schedule, which is described in Section IV. Then wireless links will be set according to the result of the algorithm.

To realize the first step, each WTU must have a top of rack(ToR). It takes the responsibility to collect traffic information of all racks and all ToRs must take turns to inform the controller about the traffic demands of WTU. The wireless devices are also set on this ToR. When a wireless link is going to be set up in this WTU, the ToR will receive the information from controller and then arrange the wireless link available to the destination rack.

Controllers are essential in wireless schedule. After the traffic demands collection, controllers judge which nodes are hot nodes and arrange wireless links to set. Because of Openflow, the algorithm can be arbitrarily modified without changing the hardware of controllers and switches. In addition, we adopt IEEE 802.11 for exchanging the traffic information.

To simplify the model, we assume that all clocks of the racks in a DCN are synchronized. Thus all racks know when to send traffic demand without collisions.

III. OPENFLOW IN WIRELESS-ASSOCIATED DCN

In this section, we introduce the Openflow in soft-defined network and how to modify Openflow controllers to our Wireless-Associated DCN.

A. Basic cognations of Openflow

How to schedule the wireless communication flexibly is very complex in current network structure. In Ethernet-based DCN structure, the network highly depends on equipments and protocols to make transmission. Innovations are hard to have a piratical way to experiment. Protocols are deeply binded with hardware devices such as routers and switches. Having recognized the problem, the networking community is hard at work developing programmable networks, such as Open Networking Foundation, which is a user-led organization dedicated to promotion and adoption of Soft Designed Network(SDN) and SDN protocols, manages the Openflow standard[19]. Soft Designed Network(SDN) has been mentioned as a possible approach to meet the various demands of applications without modification of devices. DCN with Openflow has been realized by Google[20].

Openflow is a SDN protocol, which allows direct access to and manipulation of the forwarding plane network devices such as switches and routers. In Openflow controlled network, hardware devices such as routers and switches never make decisions where to transmit information. Network control is moved out of the networking switches to logically centralized control software in controller. Openflow provides an open protocol to program the Flowtable in various routers and switches. Traffic flows can be partitioned into different routers by network administrator. In this way, new routing protocols could be simply to experiment.

With Openflow, we can schedule traffic flows and set wireless links according to the algorithm. Openflow controllers are applied to execute our schedule, which will be discussed in detail in next subsection.

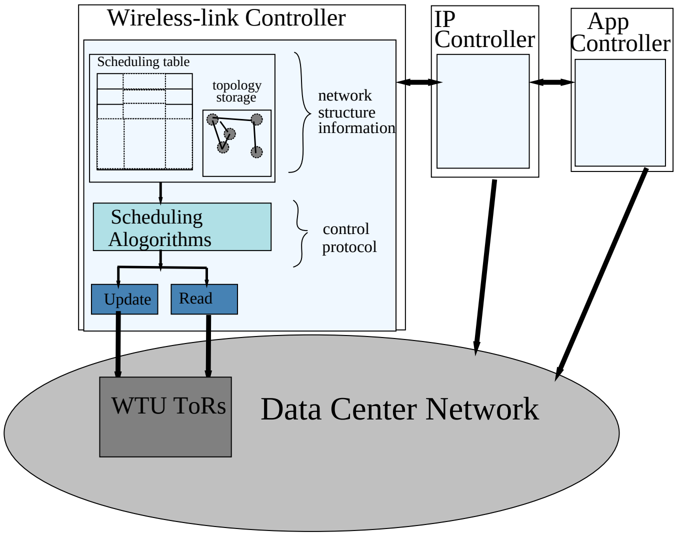

Fig. 2: The Topological structure of multiple controller controlling wireless-associated data center network.

B. Modification of Openflow Controller

Openflow controller can add and remove flow entries from the Flowtable. With Openflow controllers, we can set up the wireless links at any time and any where. In our model, each Openflow controller controls serval WTU through Openflow switches. These switches are different from current switches and contain Flowtable, with an action associated with each flow entry, to tell the switch how to process the flow. All constraints can be modified in controllers' software.

A static controller is an application running on a computer. It establishes flows to connect a set of racks. And through Openflow switches the controller can send instructions to ToRs, which will set up wireless links among WTU.

Therefore we can adjust our algorithm to meet various demands of applications. For example, when at midnight, the schedule could choose only one hot node, i.g. the hottest node, to set up wireless link because the traffic demands are usually low at that time. With this schedule power can be saved, and at the daytime we can set more wireless links to meet the increasing demand.

C. Multiple Openflow Controller

Racks and their demands of traffic are distributed in DCN, which means communication can be built between any two of racks. Thus centralized scheduling may cause high overhead. We take the multiple Openflow controllers model instead of the centralized model to achieve better performance. The structure is shown in Fig 2. There are many controllers besides these three shown in the figure.

There are three layers in our multiple controllers model. The IP controllers are distributed with interconnectiong among racks. They monitor and analyze the flows by receiving the status of data center resources from top of racks(ToR). Wireless-link controllers take the responsibility to execute the algorithm. In the wireless-link controller, the network structure information is stored and updated. The scheduling algorithm is based on the stored information and is used to decide the best arrangement of wireless links usage. The controller instructions are sent to the ToR of each WTU to realize the control protocol. In application controllers, we can change the algorithm with ease. They inform the wireless-link controllers what the schedule is. With the coordination of these three types of controllers, the procedure is distributed and each module has its own function. When a modification is necessary, we just need to change the certain part to expand the model.

IV. OPENFLOW CONTROL PROTOCOL IN WIRELESS-ASSOCIATED DCN

In this section, we provide a general protocol to solve the wireless scheduling problems mentioned in Sec. II, using the openflow mechanism . The general flow chart of our protocol is provided in Fig. 3. First, we model the network using a mathematical graph with three main constraints of wireless scheduling. Based on the modeling, we define an optimization problem and propose heuristic algorithms to tackle them. Unfortunately, because of the limited experimental equipments, we only provide some estimated results, without quantified data.

A. Model Constraints

Since the real wireless transmission system is usually randomly distributed according to various situations and particular demands, a mathematical graph is used to represent the architecture of a wireless DCN.

DEFINITION 1. A wireless transmission graph *is a directed graph $G = (V, E)$ with each node $v \in V$ indicating to a WTU, and each edge $e = (v_1, v_2) \in E$ corresponding to the transmission from v_1 to v_2.*

Notice that since only inter-unit transmissions are assigned to wireless links, we do not consider the condition of self-loop in G.

We use $T(E) = \{t(e)|e \in E\}$ to denotes the traffic demand of the edges. In some cases, the traffic demand might not be satisfied because of the limited capacity of wireless links.

Before we analyze our protocol for details, we assume that all the wireless links have the same data rate to simplify our calculations . Also, we assume that the environment is ideally stable that will not affect our system.

Now we describe the three restrictions of our optimization problem.

(i) Let $C(E) = \{c(e)|e \in E\}$ be the set of channels that are available to the edges. No wireless link would be set up if C_e is $\varnothing$. If C_e has multiple elements, this means that several links are used to accelerate the transmission. Notice that $C(E)$ is the first constraint of our optimization problem since the number of wireless transmission links are always limited.

(ii) The second constraint is the interference problem. Interference is caused by the transmission of one wireless link on other links, preventing the interfered links from utilizing the same channel. Wether a link interferes with another link is determined by the physical location of the endpoints of the corresponding transmission. Here a conflict-edge model is used to formalize the interference relationship among transmissions is provided. We use $I(E) = \{i(e)|e \in E\}$ to represent the conflict edge set of the edges, with $i(e)$ containing all the edges interfering with e. It is clear that 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.

(ii) We also should consider the limited distance that a wireless link can be used to transmit information, since the frequency of wireless links are very high, about $60GHz$. Similarly, we use $D(E) = \{d(e)|e \in E\}$ to indicate the distance between a sender and a receiver. For each $d(e)$, $d(e) \leq d_0$ should be satisfied where d_0 is a pre-defined constant representing the effective transmission distance of a wireless link.

To simplify the problem, we modify the channel allocation matrix $C(E)$ that it can include all the above three restrictions. Any links that is available to e must also not in any interference regions in any others' links, and is effective in distance. Then we define matrix $S(e, c)$ that will later be the constraint of the optimization problem.

DEFINITION. 2 R is a $e \times c$ matrix with

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

Here e^* denotes any other edges in E expect e.

B. Transmission Utility

To find out the optimized arrangement of wireless links, we need to calculate the transmission utility of all wireless links as a whole.

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 and $\delta t(e)$ the total traffic sent by the wireless links of e in a typical period. The utility of e can be expressed as (2).

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

Here, $d(e)$ is the distance factor of the wireless link which is described in pervious subsections. $\delta t(e)$ is determined by the traffic of e and the number of wireless links attached to e, which is equal to $c(e)$. Thus, $\delta t(e)$ can be computed as follow,

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

where Δt_0 denote the maximum traffic of a wireless link's transmission in a period.

We denote E_v^s as the set of transmissions that take v as the source and E_v^d as that take v as the destination. Based on definition 3, we introduce the following definition:

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.,

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

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

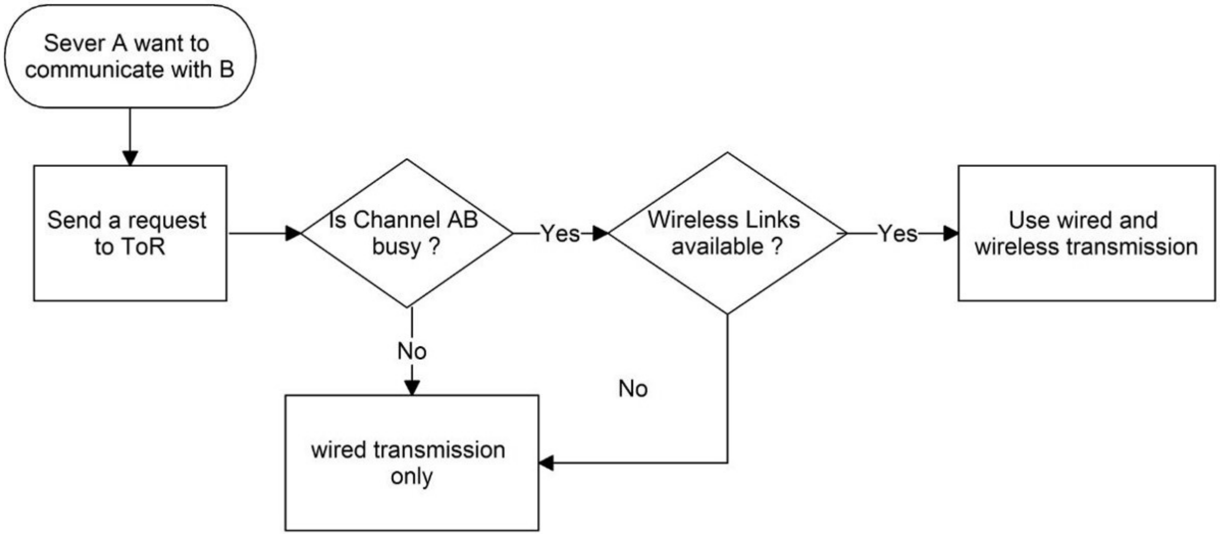

Fig. 3: Flow chart of openflow control protocol in Wireless-Associated DCN

C. The Min-Max Optimization Problem

By using Definition 4, a wireless scheduling algorithm can be formulated to resolve the set of problems described in previous sections, the most important of which, is the congestions incurred by hot WTUs. One feasible approach is to always assign wireless links to the currently hottest node. That is to say, the final goal of our scheduling protocol is to minimize the maximum total utility. That's why we call our protocol as *Min-Max Scheduling*. The corresponding optimization problem is formulated as follow

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

$$s.t. \begin{cases} \sum_{e \in E_v^s} S(e,c) \leq 1 & \forall v \in V, \forall c \in C \\ \sum_{c \in C} \sum_{e \in E_v} S(e,c) \leq r(v) & \forall v \in V \end{cases} \tag{6}$$

Now we explain the above equations in details. In (5), the objective is to minimize the maximum remaining node utility after a transmission period. The first constraint means that no interference occurs and the link is available. The second one describes the restrictions of limited radios of the WTU. Since the wireless links should be transferred and received according to the radios, the second equation ensures that the number of active links belonging to a WTU is no more than the number of radios of the WTU. Any links limited-source and distance problems, which means that it can transmit data simultaneously with other working links. Now we convert Eq. (5) as follow:

$$\min(x) \tag{7}$$

$$s.t. \begin{cases} \sum_{e \in E_v^s} S(e,c) \leq 1 & \forall v \in V, \forall c \in C \\ \sum_{c \in C} \sum_{e \in E_v} S(e,c) \leq r(v) & \forall v \in V \\ \max(\hat{u}(v) - \sum_{e \in E_v^s} u(e)) \leq x & \forall v \in V \end{cases} \tag{8}$$

Obviously, by introducing an additional variable x, problem (7) becomes an Integer Linear Programming, which is a well-known NP-hard problem.

D. Scheduling Algorithm

We design a greedy algorithm to tackle (7). First we introduce two more definition which we will use them in the algorithm.

DEFINITION 5. A node $v \in V$ is a overloaded node (ON) if $\sum_{c \in C} \sum_{e \in E_v} S(e,c) \geq r(v)$.

DEFINITION 6. An edge $e = (v_1, v_2) \in E$ is a overloaded edge (OE) if $\sum_{e \in E_{v1} \cap E_{v2}} S(e,c) \leq 1$ for any $c \in C$.

The following Algorithm outlines our approach of Min-Max Scheduling. we choose the hottest pending WTU v from V_p at every iteration. V_p denotes the set of pending nodes. No more links can be set up to v if either v becomes a ON or all its working links become OEs; there is no need to add wireless links to v if all its transmissions have no remaining traffic. For both conditions, v is regarded 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 traffic of transmission remained decreases correspondingly. If no pending node is left, the algorithm terminates and the resultant matrix S is the best channel allocation scheme of our Min-Max Scheduling. The flow chart of the MIN-MAX SCHEDULING is shown in Fig. 4.

```
ALGORITHM. (MIN-MAX SCHEDULING)
INPUT: $G = (V,E), C(E), T(E)$;
OUTPUT: $S$;
$S \leftarrow 0$;
$V_p \leftarrow V$;
do
    $v \leftarrow$ arg $\max_{\bar{v} \in V} \hat{u}(\bar{v})$;
    if($v$ is a ON)
```

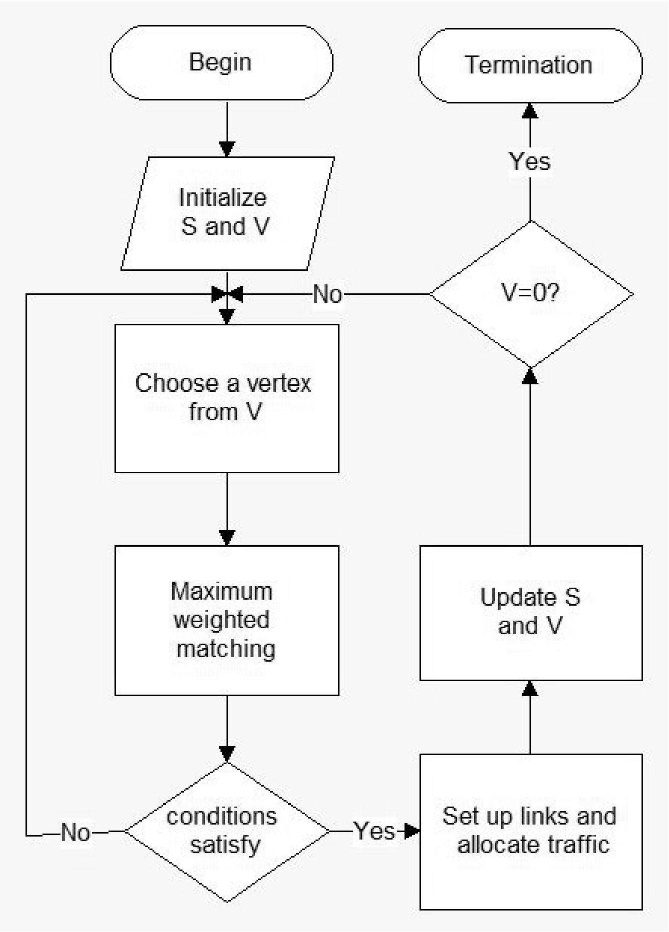

Fig. 4: Flow chart of MIN-MAX SCHEDULING algorithm

```
    {
        $V_p \leftarrow V_p - v$;
    }
    else
    {
        $\bar{e} = (v1, v2) \leftarrow$a random element in;
        $\{e|e \in E_v^s \wedge t(e) \geq 0 \wedge e$ is not a OE$\}$;
        $\{c|c \in c(e) \wedge \sum_{e \in E_{v1} \bigcup E_{v2}} S(e,c) = 0\}$;
        $S(\bar{e}, \bar{c}) \leftarrow 1$;
        $t(\bar{e}) \leftarrow \max\{0, t(\bar{e}) - \Delta t_0\}$;
    }
while($V_p \neq \emptyset$)
return S
```

The time complexity of the Algorithm depends on its number of iterations. We can evaluate its upper bound of execution time though the time of each iteration is not deterministic. We assume that we add no more than one wireless link in each iteration. Also we assume that r radios is available for each WTU. Therefore we can establish $|V|.\min\{r/2, |C|\}$ links at most. One node is removed from the pending node set if no link are set up during an iteration. 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 $O(|V|)$ time to find the hottest node and determining if a new link can be added requires $O(|V||C|)$ time to traverse all the nodes and channels.

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

E. Performance Analysis

The scheduling algorithm we use is much similar to the BEST-EFFORT Scheduling Algorithm used in [21]. Thus we can evaluate our performance using the results given in BE Scheduling. The simulation results are reported in Fig. 5. In the legend *unbalanced* and *uniform* stand for the two traffic distributions, respectively; while *high* means that the bandwidth of wireless links is the same as that of wired links, and *low* indicates that the bandwidth of wireless links is only 10 percent that of the wired links.

Number of radios—— The impact of the number of radios of each WTU is investigated. The results show that wireless links lead to a great improvement for both types of distributions and the improvement increases as the number of radios grows.

It is clearly that the maximum number of links that can be added to a node is limited by the number of radios. In the unbalanced traffic distribution, when most nodes complete their transmissions, wireless links can only be added to a few nodes with a high volume of traffic. Thus, some radios would become idle. On the other hand, in the uniform traffic distribution, the utilization ratio of the radios can stay at a higher level for a longer time because there are still a lot of nodes having traffic to transmit. Therefore, the normalized transmission time of unbalanced traffic distribution is higher than that of the uniform traffic distribution. As the bandwidth of wireless links becomes higher, wireless transmissions play an increasingly important role, and thus the effect has remarkable growth. Notice that the completion time is shortened considerably when the bandwidth of wireless links is the same as that of wired links. This is because the wireless scheduling provides dozens of wireless links equivalent to wired links, which can significantly improve the net-work capacity.

Number of Wireless Links—— We study the impact of the number of wireless links by removing some of the wireless links obtained with our algorithm. For the unbalanced traffic distribution, we observe that the network with only 30C40 additional wireless links achieves almost the same result as that obtained by the network with 100 wireless links. This is because 30C40 wireless links are enough to maintain high utilization of the radios belonging to the nodes with high traffic. Moreover, the addition of a few wireless links to the wired network can make great contributions even if the wireless bandwidth is low. This is due to the fact that these additional links are employed to resolve the bottleneck of the network by our scheduling algorithm. As a result, the completion time of the unbalanced traffic distribution is lower than that of the uniform traffic distribution. As for the uniform traffic distribution, the performance keeps increasing as the number of wireless links rises because there are always nodes that need extra bandwidth to accelerate transmissions.

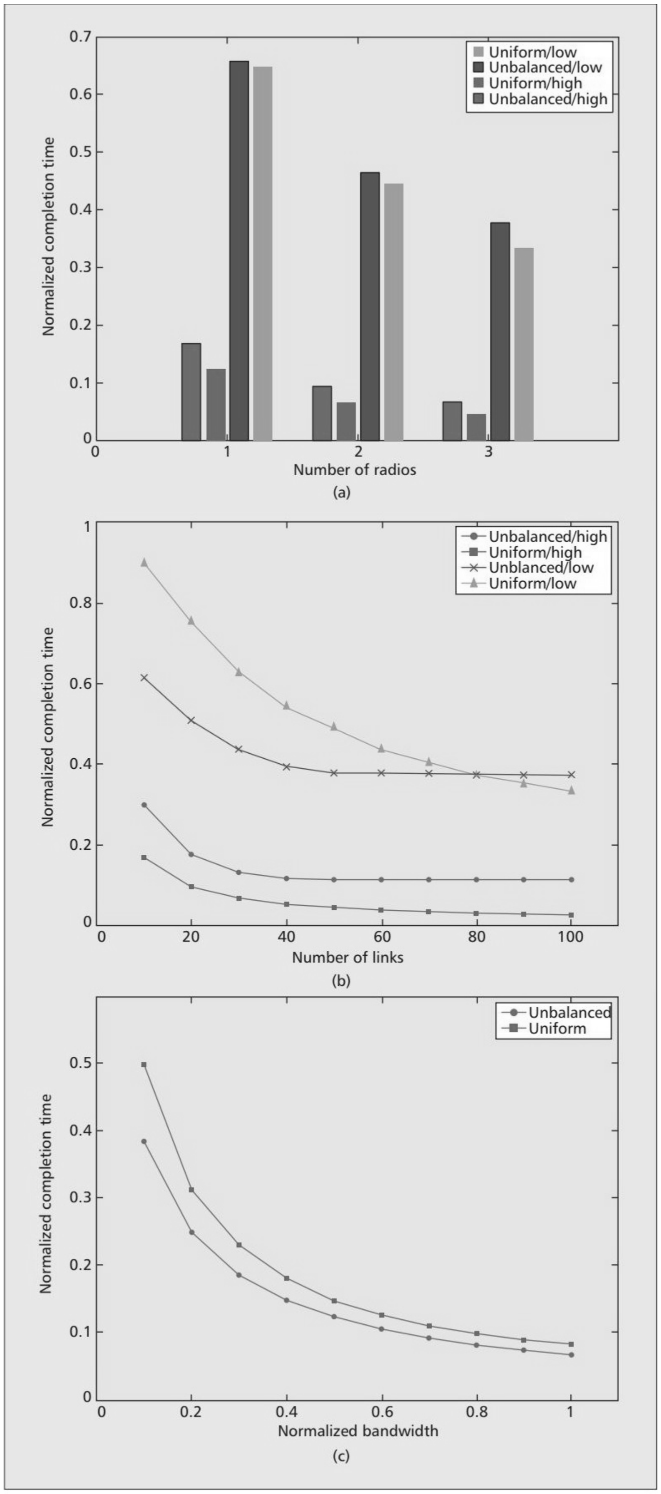

Fig. 5: Simulation results: a) impact of the number of radios; b) impact of the number of wireless links; c) impact of the bandwidths of wireless links.

Bandwidth of a Wireless Link−− Simulations are also performed to investigate the impact of the bandwidth of a wireless link. The results are reported in Fig. 5c, where normalized band-width stands for the ratio between the bandwidth of a wireless link and that of a wired link. The trends of both traffic distributions are similar: the completion time drops at a fast speed when the normalized bandwidth increases from 0.1 to 0.5; the decrement is small when the normalized bandwidth becomes higher than 0.8. These phenomena result from the fact that if the bandwidth of wireless links is very low, the transmissions assigned to wire-less links would suffer from long delay and become a bottleneck. Solving this problem by increasing the bandwidth can produce a much better result.

V. CONCLUSION

Wireless networking is proposed as a feasible approach to handle the limitations of Ethernet-based DCN architectures. In this paper, in order to make perfect control of wireless links in DCN, we provide an Openflow Control Mechanism to establish a Openflow controlled wireless-associated data center network. This design can greatly improve the traffic capacity of DCN as well as solve the problems of traditional methods of wireless networking in DCN. Although we do not provide our own simulations according to limited experimental equipments, we compare our scheduling algorithm with that of similar researches and conclude in very optimistic results. In future works, we will purchase some flyway products and simulate a simple DCN based on Openflow via NS-2. More research on this topic will be done in experiments.

ACKNOWLEDGMENT

We thank Xihua Tian, Huangchi Liu and Wanjuan Chen for some helpful discussions. This work is supported by the course-*Wireless Communication and Networking* in Shanghai Jiao Tong University(2013).

REFERENCES

[1] Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. ”The Google file system.” ACM SIGOPS Operating Systems Review. Vol. 37. No. 5. ACM, 2003.

[2] Heller, Brandon, et al. ”ElasticTree: saving energy in data center networks.” Proceedings of the 7th USENIX conference on Networked systems design and implementation. 2010.

[3] Webb, Kevin C., Alex C. Snoeren, and Kenneth Yocum. ”Topology switching for data center networks.” USENIX Hot-ICE 11 (2011).

[4] Um, Ki Sung, et al. ”Development of an HL7 interface engine, based on tree structure and streaming algorithm, for large-size messages which include image data.” Computer methods and programs in biomedicine 80.2 (2005): 126-140.

[5] Gyarmati, Lszl, and Tuan Anh Trinh. ”Scafida: A scale-free network inspired data center architecture.” ACM SIGCOMM Computer Communication Review 40.5 (2010): 4-12.

[6] M. Al-Fares et al. ”A scalable, commodity data center network architecture.” InSIGCOMM, 2008

[7] A. Greenberg et al. ”VL2: A scalable and flexible data center network”. InSIGCOMM, 2009.

[8] C. Guo et al. BCube, ”High performance, server-centric network architecture for data centers”. InSIGCOMM, 2009

[9] S. Kandula, J. Padhye, and P. Bahl. ”Flyways to de-congest data center networks”. InHotNets, Nov. 2009.

[10] Kandula, Srikanth, Jitendra Padhye, and Paramvir Bahl. ”Flyways to de-congest data center networks.” (2009).

[11] Halperin, Daniel, et al. ”Augmenting data center networks with multi-gigabit wireless links.” ACM SIGCOMM Computer Communication Review. Vol. 41. No. 4. ACM, 2011.

[12] McKeown, Nick. ”Software-defined networking.” INFOCOM keynote talk, Apr (2009).

[13] Drr, Frank. ”Software-defined Networking.” (2012).

[14] McKeown, Nick, et al. ”OpenFlow: enabling innovation in campus networks.” ACM SIGCOMM Computer Communication Review 38.2 (2008): 69-74.

[15] Heller, Brandon, et al. ”ElasticTree: saving energy in data center networks.” Proceedings of the 7th USENIX conference on Networked systems design and implementation. 2010.

[16] Matias, Jon, et al. ”An OpenFlow based network virtualization framework for the cloud.” Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on. IEEE, 2011.

[17] Cole Schlesinger, Matt Caesar, et al. ”Programming Data Centers Declaratively.”

[18] Pries, Rastin, Michael Jarschel, and Sebastian Goll. ”On the usability of OpenFlow in data center environments.” Communications (ICC), 2012 IEEE International Conference on. IEEE, 2012.

[19] Open Networking Foundation.”Open Networking Foundation: SDN Defined.”

[20] Google Inc. ”Inter-Datacenter WAN with centralized TE using SDN and OpenFlow.”

[21] Cui, Yong, et al. ”Wireless data center networking.” Wireless Communications, IEEE 18.6 (2011): 46-53.

[22] Cui, Yong, Hongyi Wang, and Xiuzhen Cheng. ”Wireless link scheduling for data center networks.” Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication. ACM, 2011.