Sandwich Tree: A new datacenter network based on passive optical devices

Pei Jin[a,*], Changcheng Huang[a]

[a]*Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, K1S 5B6, Canada*

Abstract

Most datacenter network designs overwhelmingly use expensive and power-consuming electronic switches or expensive active optical switches with long reconfiguration time. In this paper, we explore architectural solutions to leverage the design elements of Passive Optical Cross-Connection Networks with Multiple Planes (POXN/MPs) and Passive Optical Cross-Connection Networks with Multiple Planes and Bundled Ports (POXN/MP-BPs), both of which consist primarily of passive optical fabrics and optical transceivers that replace groups of switches in hierarchical networks. Through simple physical interconnections, our proposed architectures allow datacenter network (DCN) to incrementally scale out in network capacity. From developed formulas for calculating cost and power consumption, we demonstrate that POXN/MP-BPs can significantly reduce the cost and power consumption of datacenter networks compared to the traditional DCNs. To lower overhead and adapt to the types of real datacenter scenarios that are possible with POXN/MP-BPs, we propose the new Multiple Channels with Bundled Ports Distributed Access Protocol (MCBDAP), which outperforms the Multiple Channels Distributed Access Protocol (MCDAP) for POXN/MP in terms of bandwidth efficiency, especially for those applications involving higher proportions of inter-rack traffic than intra-rack traffic.

Keywords: datacenter networks, optical datacenter networks, passive optical devices, bundled ports, power consumption

1. Introduction

Datacenter networks (DCNs) are important for delivering web services, online services, social networks, and modern data storage infrastructures. They also play a key role in cloud computing [1]. At present, most DCN designs involve the massive usage of active and expensive devices, such as electronic switches and active optical switches. However, given the increasing demands for cloud infrastructure, the semiconductor industry has reached the physical limits of voltage scaling [2, 3]. Bandwidth and power consumption requirements for future systems reach 400 PB/s and 20 MW [4], and the need to reduce these usages has necessitated changes to future datacenter architectures. At the same time, some studies [5, 6] have shown that datacenters should be able to handle highly dynamic and unpredictable traffic patterns, which change constantly at a granularity of 15 ms. Furthermore, many applications involving large and distributed computations are spread across many racks, meaning that datacenters must be capable of providing high bandwidth to entire networks [7].

Current DCNs use a variety of traffic patterns. This creates a significant burden when the DCNs are in their communication stages. When facing the challenge of massive amounts of transferred data and rapid changes in traffic patterns, high network performance often cannot be achieved due to oversubscription, which results in bandwidth bottlenecks that exceed the percentage of the oversubscribed factor when a number of hosts in a pod would like to communicate with hosts in other pods [11]. Consequently, more and more new non-blocking network designs based on electrical packet switches, such as BCube [8], VL2 [9], and fat tree [10], have been proposed. VL2 and fat tree utilize Clos topologies to build more flexible DCNs with greater functionality, including extensive path diversity among servers, non-blocking performance, robustness-facing link failures, etc [11]. BCube, a recursive defined structure, offers the novel idea that the server plays a role in forwarding packets on behalf of other servers. These designs reduce costs by using commodity switches instead of non-commodity switches; however, the high power consumption of electronic switches has not yet been considered.

Motivated by the advantages of large bandwidth over packet switching, some fully optical and optical burst switch-based DCN designs have been proposed. This field has been attracting more and more attention. Many papers have discussed the issue of how to use Optical Burst Switching (OBS)s in a network. A variety of contention resolution and avoidance schemes have been presented for OBS networks [12]. However, an OBS is an active device, which must set up lightpaths before sending packets. This step occupies network resources. If traffic is not significant, it will take a long time to buffer, thus increasing packet delay. For this reason, most OBS-based DCNs, such as Helios [13] and C-through [14], are used only as bypaths while electrical networks are retained. Helios utilizes the MEMS optical switch, and WDM technology provides higher bandwidth with less power consumption than the electrical packet switch-based topology. C-through is a hybrid packet and circuit-switched datacenter network architecture (called HyPac), which adds optical circuit switches between switches

*Corresponding author. Tel.: +1 613 262 9001.
Email addresses: peijin@sce.carleton.ca (Pei Jin), huang@sce.carleton.ca (Changcheng Huang)

across different racks in order to match their maximum aggregated bandwidth demands following configuration, while maintaining the traditional packet switches of a tree topology.

Different from these two approaches, an all-optical solution based on AWGR has been proposed in [15]. Due to the elimination of the electrical loopback buffer, to avoid contention, the denied packets will be blocked in the buffer at the host. The resulting packet retransmission will lead to more latency. Another drawback is the high hardware cost caused by the wide usage of expensive active elements, such as Field-Programmable Gate Array (FPGA) and Reflective Semiconductor Optical Amplifier (RSOA). Another all-optical solution [16] utilizes the Wavelength-Divison Multiplexing (WDM), Wavelength Selective Switch (WSS), Micro-Electro-Mechanical-Systems Switches (MEMS), optical circulators, and optical transceivers to build a flexible topology. However, massive usage of expensive optical components, such as WSS and MEMS optical switches, substantially increases the cost of building Proteus-based datacenters. In addition, due to both the limitations in maximum channels of Dense Wavelength Division Multiplexing (DWDM) technology and the port density of the ToR switches, Proteus' low scalability hinders its performance in DCNs.

Due to the complex control plane, slow switching issue, and expensive hardware cost, it undermines the capacity advantage of optical devices in DCNs. From this point, two passive optical devices, POXN and POXN/MP, both of which are power-efficient and cost-efficient, and can provide broadcast transmission medium, attract our attention because of their potential to be leveraged as core elements to build DCNs. POXN [17] proposed passive optical coupler fabric constructed by multiple stages of couplers in Banyan topology as the switch fabric for traffic transmission in networks. POXN can scale up to 81-port coupler fabrics using existing 2×2 or 3×3 couplers and enables collision-free frame transmission based on a distributed polling protocol. The drawback of the POXN is the low average transmission rate per port for unicast traffic. Based on POXN, POXN/MP [18] adds an additional plane for unicast traffic transmission. Under the new protocol, POXN/MP coordinates dynamic traffic distribution in two planes. However, the potential of employing POXN and POXN/MP in a large-scale DCN has not been explored. Based on the limitations of existing devices, the problems we are facing can be summarized as follows:

• A single POXN or POXN/MP coupler fabric can accommodate up to 81 ports. To further scale up the number of ports as required by large DCNs, we cannot simply connect multiple POXNs or POXN/MPs in a cascade due to insertion losses and power splits. A new network architecture that can scale up to support arbitrary number of servers in a DCN while taking the low-power and low-cost advantages of POXN or POXN/MP is required.

• POXN/MPs must have the same port types. This limits their ability to engage in traffic aggregation, which is typically desired in a DCN, especially at the access level. For example, a typical top-of-rack (ToR) switch has twenty 1 Gbps downlinks and two 10 Gbps uplinks. POXN/MPs do not support this kind of configuration.

In this paper, we propose a Passive Optical Cross-connection Network with Multiple Planes and Bundled Ports (POXN/MP-BP) based on the POXN/MP. Compared to the old POXN/MP, the POXN/MP-BP exhibits better performance across many key indicators, including lower hardware cost, lower power consumption, and higher bandwidth efficiency, especially in cases requiring traffic aggregation. We will present POXN/MP-BP, highlighting the advantages in physical transmission system, power consumption, and algorithm compared to POXN/MP. Leveraging proposed POXN/MP-BP and existing POXN/MP, we propose a new DCN based on these passive optical devices, called Sandwich Tree. Certain levels of existing electronic switches in current DCNs (e.g. ToR switch and core switch) can be replaced by POXN/MPs or POXN/MP-BPs, thus building a tree topology through placing electronic switches and POXN/MPs or POXN/MP-BPs alternatively. The introduction of this alternative placement rather than massive usage of electronic switches will significantly lower the total cost to build and power consumption to maintain large DCNs. Furthermore, as optical signal is transmitted in a broadcast-and-select fashion, it is ideal for dynamic traffic patterns such as multicast and broadcast traffic to be routed with less duplicate packets in intermediate electronic switches.

The remainder of the paper is organized as follows. In section 2, we present POXN/MP-BPs in terms of physical interconnections, facility cost and power consumption savings, the new protocol–MCBDAP–for collision-free transmission, and the new algorithm for MCBDAP. In section 3, we present our proposed Sandwich Tree structure by exploring the approach of integrating POXN/MP-BP as a design element into a multi-tier DCN step-by-step. In addition, we analyze its benefits in terms of hardware cost and power consumption. In section 4, we analyze the network performance of the proposed structure–the Sandwich Tree–based on POXN/MP-BPs and/or POXN/MPs from device-level to network-level. In section 5, we highlight our contributions by comparing with the existing works. In section 6, we conclude the paper.

2. POXN/MP-BP

Before introducing POXN/MP-BP, we will discuss previous works on POXN and POXN/MP in terms of physical transmission system, wavelengths assignment in different planes, and merits and drawbacks.

Ni et al. [17] proposed POXN where a passive optical coupler fabric built by integrating multi-stage of 2×2 or 3×3 couplers in Banyan structure is used as the core device. Each server port has an array of fix-tuned transmitters and an array of fix-tuned receivers, all working at different wavelengths. Leaving a transmitting port of a server, the wavelengths are combined by a multiplexer before they reach an ingress port of the coupler fabric in the middle. Within the fabric, the wavelengths are carried to all egress ports due to the broadcast nature of the fabric. Out of an egress port of the coupler fabric, all wavelengths are split by a demultiplexer before reaching the receiving port.

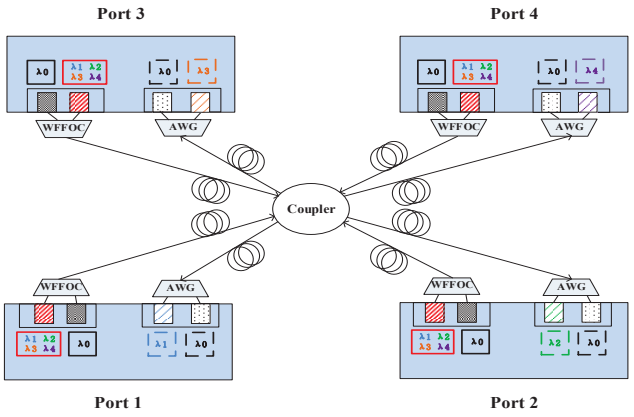

Figure 1: Physical interconnections of a four-port POXN/MP.

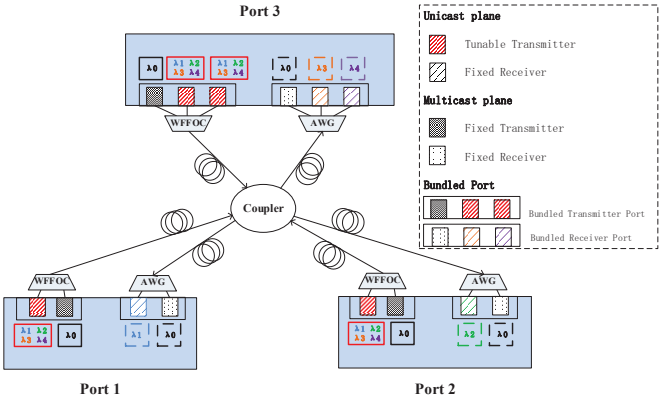

Figure 2: Physical interconnections of a three-port POXN/MP-BP.

The power budget of a transmitter determines how many ports a passive coupler fabric can have.

An et al. [18] proposed a POXN/MP, which is an upgrade version of the POXN that introduces a new plane for transmitting unicast traffic. A four-port POXN/MP transmission system is shown in Fig. 1. In this architecture, four ports are connected through a coupler fabrics as well. The difference is that each port is now equipped with two pairs of transmitters and receivers, one of which uses a fixed wavelength shared by all ports for multicast traffic, and the other of which uses a tunable transmitter and a fixed receiver with wavelength different from port to port for unicast traffic. The multicast traffic should be sent in a packet-by-packet manner, while unicast traffic can be sent in parallel if the receiving ports are different.

We find that POXN/MPs have the potential to be leveraged as core DCN devices, since they are characterized by low power consumption, low cost, high transmission capacity, and the broadcast transmission of multicast traffic. These merits make POXN/MPs more competitive devices for DCNs than electronic and active optical switches. However, if a POXN/MP is integrated into a multi-tier DCN, it cannot address traffic aggregation at a certain level of the DCN, since all POXN/MP ports are homogeneous. Thus, a new device is needed that can satisfy the following three requirements: First, this device should retain the characteristics of POXN/MPs. Second, it should provide support functionality for traffic aggregation at a certain level of DCN. Third, it should not degrade performance, such as channel efficiency. Thus, we design a new device, called the POXN/MP-BP, based on the POXN/MP.

In this section, the physical interconnections in POXN/MP-BPs will be discussed, and an example of a three-port POXN/MP-BP will be provided. Then, we will explore how much the power budget can be saved for POXN/MP-BP compared to the POXN/MP. Based on the physical transmission system, we will develop how the new algorithm works in the new MCBDAP, with the assistance of pseudo code and a transmission scheduling example. In addition to satisfying the requirements of building a DCN with a passive optical device, POXN/MP-BPs can also bring more flexibility to transmission scheduling through their physical interconnection changes. Thus, we will discuss how POXN/MP-BPs achieve this objective and how their efficiency algorithms can be improved by reducing constant tuning times in certain scenarios.

2.1. Physical Transmission System

To support the traffic aggregation function, POXN/MP-BPs must modify the physical interconnections of previous POXN/MPs. These modifications should be implemented without reducing channel efficiency, increasing hardware cost, or increasing power consumption. A three-port POXN/MP-BP transmission system is shown in Fig. 2.

Like POXN/MPs, and without introducing other devices, POXN/MP-BPs employ a passive cross-connection coupler fabric that acts as their core component, connecting all ports through its interfaces. Additionally, POXN/MP-BPs use two optical planes that enable dynamic traffic patterns in DCNs.

The main difference between POXN/MPs and POXN/MP-BPs is the introduction of port bundling on designated ports. For instance, in Fig. 2, ports 3 and 4 in Fig. 1 are bundled together as a logical port. Accordingly, port 3 is equipped with one more transmitter and one more receiver. As a result, the 2×1 WFFOC is changed to a 3×1 WFFOC and the 1×2 AWG is changed to a 1×3 AWG. Bundling can also save the number of transponders required. As shown in Fig. 2, the bundling port only needs one transponder instead of the two transponders in Ports 3 and 4 in Fig. 1 for multicast plane. The two optical planes can still work effectively, as long as the unicast transmitters use non-overlapping wavelengths to deliver unicast traffic simultaneously and the multicast transmitter uses its exclusive wavelength to broadcast multicast traffic sequentially. More importantly, these changes do not affect the scheduling performance at all and, instead, bring much more scheduling flexibility, thus improving efficiency, which will be discussed in section 2.3.1.

2.2. Power Consumption Advantage

In this section, we will discuss how bundled ports can save power consumption for transmitting ports by reducing power loss caused by coupler.

The problem with the mechanism for bundled ports is that more ports are multiplexed/demultiplexed by WFFOC/AWG, which may increase the power loss for WFFOC/AWG. However, considering the existing optical technology, the power loss caused by a multiplexer/de-multiplexer is 2.5 dB for Coarse Wavelength Division Multiplexing (CWDM) based on thin-film filers [17]. Therefore, without causing extra power loss, the number of bundled ports can be multiplexed/de-multiplexed into a single fiber which connects to an $N \times N$ coupler fabrics.

We now discuss how the POXN/MP-BP saves power budget for transmitters. Compared to the same system capacity of a POXN/MP, we assume that in a POXN/MP-BP, half of the ports are individual ports and the other half are bundled ports (each of which is B-in-1 bundled port). Now we can know that both of B-in-1 bundled port and individual port need one input/output port for coupler fabrics and there are $N/2B$ B-in-1 bundled ports and $N/2$ individual ports. This indicates we only need a $(N/2B + N/2)$-port coupler fabrics. Recall from the section 2, power loss caused by 3×3 coupler and 2×2 coupler can be reduced from $(5.47 \times \lceil \log_3 N \rceil - 0.2)$ dB to $[5.47 \times \lceil \log_3(N/2B + N/2) \rceil - 0.2]$ dB and from $(3.71 \times \lceil \log_2 N \rceil - 0.2)$ dB to $[3.71 \times \lceil \log_2(N/2B + N/2) \rceil - 0.2]$ dB. With such decrease of power loss, we can lower the required power budget for transmitters while retaining the same system capacity.

In Fig. 3, it is clear to see that through saving the number of input/output ports for coupler fabrics, we can use the combination of 2×2 coupler and 3×3 couplers to build a coupler fabrics with a relatively smaller port count and thus reducing the power loss at the coupler fabrics. For example, the POXN/MP needs 5 stages of 2×2 couplers to build a 32×32 coupler fabrics, which leads to $3.71 \times 5 - 0.2 = 18.3$ dB. By contrast, the POXN/MP-BP with 2-in-1 bundled ports ($B = 2$) only needs a 24×24 coupler fabrics ($N/2B + N/2 = 24$) that can be built by 3 stages of 2×2 couplers and 1 stage of 3×3 coupler, which leads to $3.713 + 5.47 \times 1 - 0.2 = 16.4$ dB. It is interesting to see that two lines indicating the POXN/MP-BP with 2-in-1 and 4-in-1 bundled ports overlap. The reason is that though the POXN/MP-BP with 4-in-1 bundled ports ($B = 4$) needs a 20×20 coupler fabrics ($N/2B + N/2 = 20$), we can only build a 24×24 coupler fabrics with 4 unused ports since manufacturing a unit coupler with more than 3×3 ports is not trivial with the existing technology [17].

Fig. 4 shows that through saving the number of input/output ports for coupler fabrics for POXN/MP constructed by 3×3 coupler, we can still use the combination of 2×2 and 3×3 couplers to build relatively smaller port count of coupler fabrics for POXN/MP-BP and thus reducing the power loss at the coupler fabrics. Different from Fig. 3, POXN/MP and POXN/MP-BP with 2-in-1 bundled ports overlap. We now explain why this happens. For example, POXN/MP needs 3 stages of 3×3 couplers to build a 27×27 coupler fabrics. POXN/MP-BP with 2-in-1 bundled ports ($B = 2$) needs a 20×20 coupler fabrics ($(N + 1)/2B + (N - 1)/2 = 20$) that can be built by 3 stages of 2×2 couplers and 1 stage of 3×3 couplers, which leads to $3.71 \times 3 + 5.47 \times 1 - 0.2 = 16.4$ dB. Evidently, using 3 stages of 3×3 couplers is better than this approach. Thus, POXN/MP-BP with 2-in-1 bundled ports will choose 27×27 coupler fabrics.

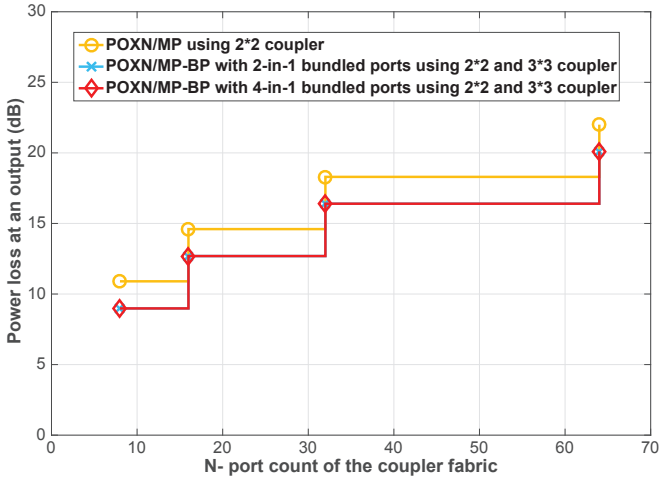

Figure 3: Compare power loss at a coupler output for POXN/MP using 2×2 coupler and POXN/MP-BP with 2-in-1 bundled ports and 4-in-1 bundled ports using 2×2 and 3×3 couplers.

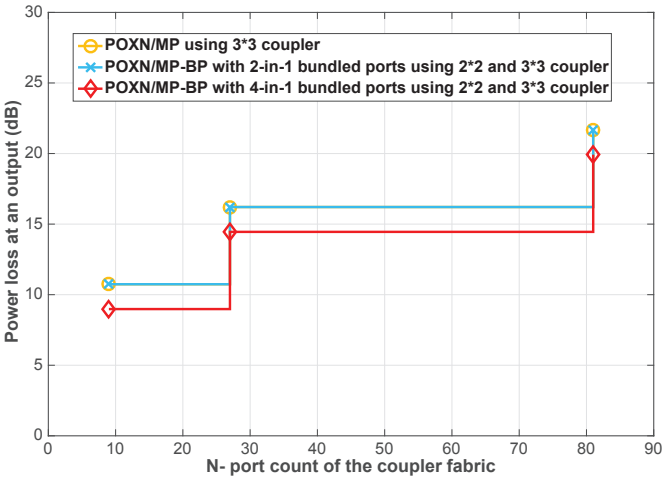

Figure 4: Compare power loss at a coupler output for POXN/MP using 3×3 coupler and POXN/MP-BP with 2-in-1 bundled ports and 4-in-1 bundled ports using 2×2 and 3×3 couplers.

By contrast, POXN/MP-BP with 4-in-1 bundled ports (B=4) needs a 18×18 coupler fabrics ($(N + 1)/2B + (N - 1)/2 = 18$) that can be built by 2 stages of 3×3 couplers and 1 stage of 2×2 couplers, which leads to $3.71 \times 1 + 5.47 \times 2 - 0.2 = 14.45$ dB. Thus, in this scenario, POXN/MP-BP should have a larger bundled port count to save power consumption.

2.3. MCBDAP

The introduction of bundled ports changes the unicast plane. The best time to send and receive unicast traffic will differ from those for POXN/MPs. To adapt to these changes in the physical interconnection, a new protocol–MCBDAP–is proposed.

The MCBDAP is still divided into two phases: the discovery phase and the data transfer phase. Like the MCDAP, the MCBDAP begins with a discovery phase, which is followed by a data transfer phase. The POXN/MP-BP comprises several groups of bundled ports that are physically combined in each switch. The

changes to the physical interconnections do not make any difference for the multicast plane. However, the introduction of bundled ports creates more flexibility in the data transfer phase for the unicast plane by reducing the tuning time for the tunable transmitter and the variable idle time for available receiving ports.

In this section, we will explain how the new algorithm works using pseudo code. Furthermore, the new algorithm's improvements in efficiency, as compared to the old MCDAP algorithm, will be discussed under different scenarios.

2.3.1. Benefits and trade-offs

The difference between the MCBDAP and the MCDAP is that the MCBDAP provides much more flexibility in certain scenarios, thus improving algorithm efficiency by reducing type 2 mismatches and constant tuning time. To explore these benefits, we must analyze which kinds of scenarios trigger them. This is helpful when we simulate the POXN/MP-BP in a more complex environment (e.g., a network-level environment), since we can leverage this flexibility and avoid unnecessary overhead as much as possible.

We explore four scenarios that can trigger the MCBDAPs benefits, as follows:

Scenario 1:

At the beginning of each cycle, transmitter i has a much larger traffic volume $T^Q_{ij'}$ to receiver j' than T^Q_{ij} receiver j. Considering the LQF algorithm for MCDAP, transmitter i is not allowed to send to receiver j, since receiver j is being used by the transmission $T^Q_{i'j}$. The new algorithm also cannot send to receiver j due to the potential for a wavelength collision. However, with the introduction of bundled ports, transmitter i is able to send the traffic $T^Q_{ij'}$ which originally was designated to be sent from transmitter i to receiver j' to receiver j at the current stage. Based on the largest queue first principle, we know that if a larger queue size can be selected, all transmitters tend to finish at roughly same time, which reduces the potential for type 2 mismatches.

Scenario 2:

Transmitter i has just finished sending its traffic to receiver j (which is one of receivers of the bundled port). Assuming j' is the other receiver of the same bundled port, if $T^Q_{ij'}$ has not been sent yet, transmitter i can keep sending unicast traffic $T^Q_{ij'}$ to receiver j without tuning to another wavelength. Unlike in the old algorithm for the POXN/MP, in the new algorithm, transmitter i does not need to wait for another available receiver and then tune its wavelength to this receiver. More specifically, the transmitter i will not experience a constant tuning time T^T during the two continuous data bursts.

Scenario 3:

Transmitter i, which is a bundled port, compares its traffic size with each other bundled transmitters on the same node that is sending to the available receiver j. Then, it picks up the largest traffic size to send to receiver j.

Scenario 4:

Transmitter i, which is a transmitter of a bundled port, pairs with receiver j. If the receiver j is also a receiver of a bundled port, the traffic volume to be sent from the transmitter i to the receiver j will be compared with all other combinations of the other transmitters of the same bundled port (including i) and the other receivers of the same bundled port (including j), making the selection range much wider. As a result, larger traffic volume can be picked up and sent out.

Trade-off:

If these scenarios happen frequently during the data transfer phase, this will lead to a situation in which each transmitter tends to send more traffic to receivers of the bundled ports than to receivers of the normal ports during the early stages. As a result, receivers of the bundled ports may become available, while available transmitters may have no traffic to send. This situation, which is most likely to happen close to the cycle end, may increase the potential for type 2 mismatches.

The following Fig. 5 presents the pseudo code for the core of the MCBDAP algorithm, where Q_{ij} indicates the traffic to be sent from transmitter i to receiver j, $Q_{ij_{left}}$ refers to the traffic to be sent from transmitter i to each remaining receiver j, T_i and R_j refer to the normal individual transmitter and receiver, T_b and R_b refer to the transmitter and receiver of the bundled port, t_i^{Idle} represents transmitter i's idle time, t^T represents the constant tuning time, $t_{current}$ represents the current time, t^F_{ij} represents the completion time for the transmission from transmitter i to receiver j, $t_{ij'}$ represents the transmission time for $Q_{ij'}$ (i.e., scenario 2), $t_{i'j}$ represents the transmission time for $Q_{i'j}$ (i.e., scenario 3), and $t_{i'j'}$ represents the transmission time for $Q_{i'j'}$ (i.e., scenario 4).

3. Sandwich Tree Structure

Previous calculations in [17] have shown that a POXN/MP can accommodate up to 81 ports. However, to build a large-scale DCN, tens of thousands of servers must be accommodated. To accomplish this, first, one could consider cascading the POXN/MP to achieve a high port count to connect more servers. Since current DWDM technology can support only 160 channels, the maximum number of ports on all servers is 160 [19]. If more servers join, the overlapping wavelengths will be recycled, resulting in transmission collisions. Furthermore, insertion losses and power splits in the passive optical fabric consume too much power, suggesting that the power budget may not be sufficient to transmit a signal through two POXN/MPs. Another one may attempt to connect two POXN/MP-BPs through a switch between them, rather than to cascade POXN/MP-BPs directly. Based on Googles multi-tier DCN architecture shown in Fig. 6, we propose the Sandwich Tree structure, where we can alternately place switches and POXN/MP-BPs or POXN/MPs. The motivation behind this approach is to increase uplink bandwidth from servers to switches, reduce the number of network tiers, and lower power consumption and total expenditure.

In this section, we will discuss how to build a Sandwich Tree. It is similar to fat tree topology; however, compared with the traditional fat tree topology, the Sandwich Tree introduces POXN/MP-BPs and POXN/MPs to replace portions of switch-

New Largest Queue First for POXN/MP-BP

for all transmitting port i **do**
  select $\max\{Q_{ij}\}$ to send from transmitting port i to receiving port j
  while $j, j \in R_i$ is not available or $j, j \in R_b$ has 0 bundled ports **do**
    select $\max\{Q_{ij_{left}}\}$ of the remaining $j, j \in R_i \bigcup R_b$
  end while
  disable paired transmitting port i and receiving port j and let $Q_{ij} = 0$
end for
repeat
  a pair of i and j finishes its data burst first, enable corresponding i and j and then traverse
  for all receiving port i **do**
    search for an available receiving port j
    for all transmitting port i **do**
      search for an available transmitting port i
      if an available $i, i \in T_i$ pairs an available $j, j \in R_i$ and $Q_{ij} \neq 0$ or an available $i, i \in T_b$ pairs an available $j, j \in R_i$ and $\sum Q_{i'j} \neq 0, i' \in T_b$ **then**
        pick up Q_{ij} or $max\{Q_{i'j}\}$ and disable just paired T_i and R_j and let $Q_{ij} = 0$ or $Q_{i'j} = 0$
        if $t_i^{Idle} > t^T$ **then**
          $t_{ij}^F = t_{current} + t_{ij}$ or $t_{i'j}$
        else if $t_i^{Idle} > 0$ and $t_i^{Idle} < t^T$ **then**
          $t_{ij}^F = t_{current} + (t^T - t_i^{Idle}) + t_{ij}$ or $t_{i'j}$
        else $t_i^{Idle} = 0$
          $t_{ij}^F = t_{current} + t^T + t_{ij}$ or $t_{i'j}$
        end if
      end if
      if an available $i, i \in T_i$ pairs an available $j, j \in R_b$ and $\sum Q_{ij'} \neq 0, j' \in R_b$ or an available $i, i \in T_b$ pairs an available $j, j \in R_b$ and $\sum Q_{i'j'} \neq 0, i' \in T_b$ and $j' \in R_b$ **then**
        pick up $max\{Q_{ij'}\}$ or $max\{Q_{i'j'}\}$ and disable just paired T_i and R_j and let $Q_{ij'} = 0$ or $Q_{i'j'} = 0$
        if $t_i^{Idle} > t^T$ **then**
          $t_{ij}^F = t_{current} + t_{ij'}$ or $t_{i'j'}$
        else if $t_i^{Idle} > 0$ and $t_i^{Idle} < t^T$ **then**
          if j_{last} of $i, i \in T_i \bigcup T_b == j$ **then**
            $t_{ij}^F = t_{current} + t_{ij'}$ or $t_{i'j'}$
          else
            $t_{ij}^F = t_{current} + (t^T - t_i^{Idle}) + t_{ij'}$ or $t_{i'j'}$
          end if
        else
          if j_{last} of $i, i \in T_i \bigcup T_b == j$ **then**
            $t_{ij}^F = t_{current} + t_{ij'}$ or $t_{i'j'}$
          else
            $t_{ij}^F = t_{current} + t^T + t_{ij'}$ or $t_{i'j'}$
          end if
        end if
      end if
    end for
  end for
until all queues in each transmitting port have been sent

Figure 5: New algorithm for POXN/MP-BPs.

es and we assume that there is traffic aggregation from servers to switches through first-level POXN/MP-BPs.

The first subsection explores how POXN/MP-BPs replace different layers of switches in detail. The second subsection investigates how the Sandwich Tree is constructed. The third subsection discusses how routing mechanisms work for unicast, multicast, and broadcast traffic. The last subsection illustrates the advantages of the Sandwich Tree over the traditional fat tree structure in terms of cost and power savings through detailed calculation.

3.1. POXN/MP-BPs as design elements to replace switches

To build a Sandwich Tree, we just address the issue of how to replace certain levels of switches in a multi-tier tree structure. For instance, the ToR switch aggregates traffic from source

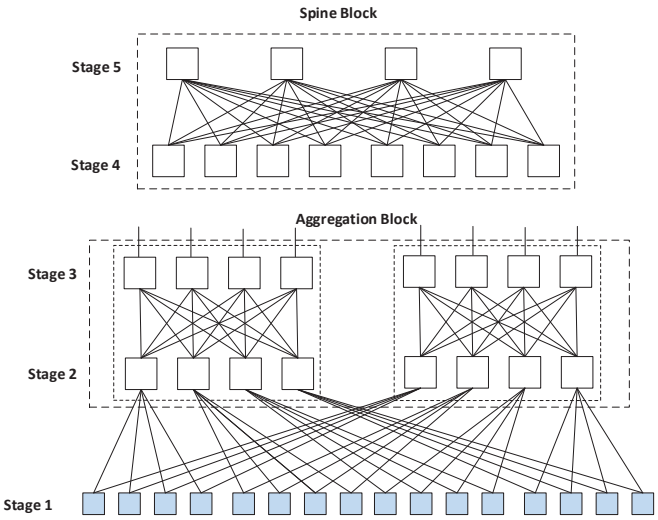

Figure 6: One example of Google's DCN with five stages of switches.

servers and is asymmetric in structure, with uplinks and downlinks that differ in speed and number. The homogeneous structure of the POXN/MP is not capable of addressing this kind of traffic aggregation; thus, in this case, the POXN/MP-BP becomes the only choice. So, in this section, we will discuss how to replace certain levels of switches in a multi-tier structure with POXN/MPs or POXN/MP-BPs.

Multi-stage Clos topologies built from massive silicon commodity switches can support the building of large-scale DCNs [1]. For example, Google's DCN has been developed and upgraded to new generations every one to two years since 2004. The foundation of all of such architectures is a multi-stage structure, which can be divided into three aspects: ToR switches, aggregation blocks, and spine blocks. One instance of such a structure is shown in Fig. 6. The extensive use of switches in a multi-stage network will significantly increase the network latency and the total cost. Thus, our goal is to deliver a network with a low hop count and a low hardware cost by replacing groups of switches using POXN/MPs or POXN/MP-BPs.

Now, we explain how POXN/MP-BPs or POXN/MPs can be leveraged to replace groups of switches. First, we will discuss the replacement of individual switch layers; then, we will explore the simultaneous replacement of multiple layers of switches.

First, we examine the ToR switch at stage 1. Assuming that 20 servers are connecting with ToR switches using twenty 1 Gbps uplinks and that the top ports of the ToR switch are connecting with stage 2 switches using two 10 Gbps uplinks, the old physical interconnections of POXN/MPs will not work to replace the ToR switches, since all POXN/MP ports are homogeneous. Therefore, a POXN/MP-BP will perform its function here to replace the ToR switch with two groups of 10 bundled ports on the two switches and 20 individual ports on each server. More specifically, the POXN/MP-BP will consist of a 48×48 coupler; 20 individual ports on the server side, each of which will be equipped with a 1×2 AWG, a 2×1 WFFOC, and a pair

of tunable and non-tunable transceivers; and two groups of 10 ports on each switch side, each of which will be equipped with a 1×16 AWG, a 16×1 WFFOC, 10 tunable transceivers, and one non-tunable transceiver. Though a 40×40 coupler is sufficient, unit couplers can be only 2×2 or/and 3×3; thus, input/output numbers can be any integer numbers composed solely of prime factors of 3 and/or 2 [17]. The case is the same for the AWG and the WFFOC.

Second, switches at stages 2, 3, and 4 have same numbers of uplinks and downlinks. Either POXN/MP-BPs or POXN/MPs can be utilized to individually replace all switches in one of these stages. For instance, a stage-2 switch has four 10Gbps uplinks and four 10Gbps downlinks, which can be easily replaced by an 8×8 POXN/MP.

Third, like switches at stages 2, 3, and 4, switches at stage 5 can also be replaced by POXN/MPs because of their symmetric characteristics. The port counts for the POXN/MPs can differ, and transceivers in this stage are normally equipped with faster speed (e.g., 40/100 Gbps). In addition to individual replacements within certain stages, POXN/MPs and POXN/MP-BPs can realize multiple replacements at different stages simultaneously; however, no two replacement stages can be adjacent. For instance, stage 1, stage 3, and stage 5 can be replaced simultaneously by POXN/MPs or POXN/MP-BPs. However, stage 1 and stage 2 or stage 4 and stage 5 cannot be replaced at the same time because of problems caused by POXN/MP interconnections, as already explained in section 3. Therefore, we can conclude that POXN/MPs or POXN/MP-BPs can be leveraged to replace, at most, $k/2$ and $(k+1)/2$ stages, respectively, of a Clos datacenter network with k even and odd stages. Furthermore, since it enables the bundled ports with random numbers, the introduction of POXN/MP-BPs can even benefit random topology networks, as long as the total number of ports does not exceed the port count limitation of the POXN/MP-BPs.

3.2. Sandwich Tree Structure

There are three main reasons for proposing Sandwich Tree structure with POXN/MP-BPs. First, POXN/MP-BPs can provide greater uplink bandwidth on certain switches because of the mechanism for bundled ports. Second, the introduction of POXN/MP-BPs will lead to lower hardware costs and higher energy efficiency. Third, POXN/MP-BPs can further save power budget for transmitters by reducing the power loss for the coupler fabric.

Fig. 7 shows the construction of an eight-pod Sandwich Tree structure. The Sandwich Tree structure replaces two levels of electronic switches–the ToR switch and the core switch–with POXN/MP-BPs and POXN/MPs, respectively.

It is important to note that we apply the fat tree addressing methodology. In a k-pod network, there are two forms of addressing, which correspond to two types of devices: servers and switches. The address form of the server can be denoted as follows: $10.pod.coupler.ID$, where pod refers to the pod number ($pod \in [0, k-1]$), coupler represents the position of the coupler to which servers are connecting ($coupler \in [0, k/2b-1]$ (where b is the number of bundled ports), and ID represents the host position ($ID \in [2, k/2+1]$). For each switch, the address can be

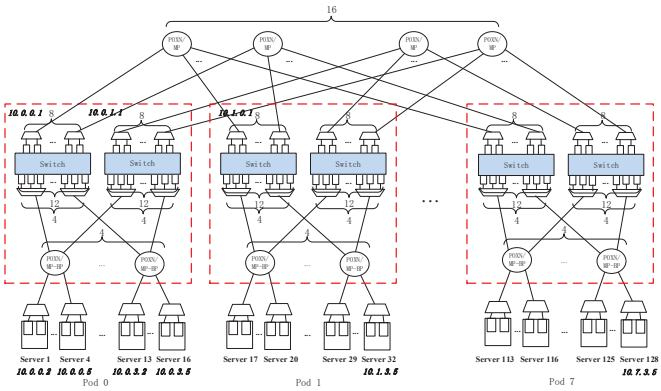

Figure 7: An example of an eight-pod Sandwich Tree with POXN/MP-BPs and POXN/MPs.

represented by $10.pod.switch.1$, where pod has the same meaning as in the server address, switch refers to the position of the switch (from left to right) in a pod, and ID number is replaced by a constant 1, which can be used to differentiate switch from server.

3.3. Routing in Sandwich Tree

In this section, we will explain how unicast and multicast traffic are delivered from the source server to the destination server(s) in a Sandwich Tree topology.

First, we will discuss how unicast traffic is sent from server to server. Before this protocol can work, two questions must be solved. First, how can a switch determine the packet destination and insert the packet into the correct output port when it receives a packet from a server? Second, how do the ports on the upper level of switches, which are connected by the top level of POXN/MPs, know which receiving port(s) to send their packets to?

Our solution to the first question leverages a new two-level routing table, which can be implemented in hardware using CAM [20]. This offers a fast approach to finding a match. The approach draws on the fat tree topology, which also implements a two-level lookup table to forward packets. Before the routing table is introduced, the transponder number of switch 10.0.0.1 is clearly presented in Fig. 8. The lower level of the switch comprises four groups of bundled ports, each of which has one transponder (i.e., 0, 3, 6 and 9) for multicast traffic and four 2-in-1 bundled ports, each of which has two transponders (i.e., 1, 2, 4, 5, 7, 8, 10 and 11) for unicast traffic. On the top side, there are eight normal ports, each of which has one transponder (i.e., 12, 14, 16, 18, 20, 22, 24, and 26) for multicast traffic and the other transponder (i.e., 13, 15, 17, 19 , 21, 23, 25, and 27) for unicast traffic.

Fig. 9 illustrates the two-level routing table for switch 10.0.0.1, which draws inspiration from [10]. First, packets generated at each server can be classified into intra-rack traffic and inter-rack traffic. Since first-level ToR switches are replaced by POXN/MP-BPs in the Sandwich Tree, servers must make a decision regarding which switch to send their packets to. Source servers can solve this problem using many ap-

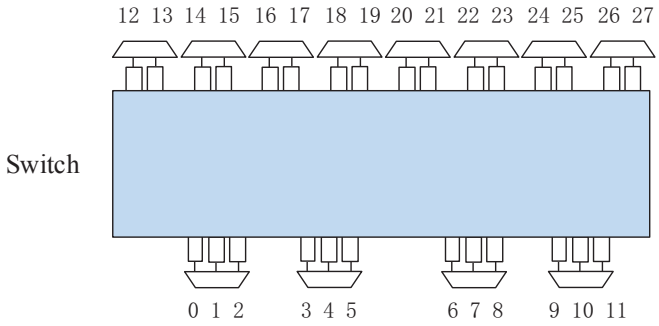

Figure 8: Transponder numbers 0 to 27 for switch 10.0.0.1 in a POXN/MP-BP with eight pods.

Table 1: Transponder number for switch 10.0.0.1 corresponding to different kinds of traffic

Switch side	Port number	Traffic type
Down	0,3,6,9	multicast traffic
	2,4,5,7,8,10, and 11	unicast traffic
Up	12,14,16,18,20,22,24, and 26	multicast traffic
	13,15,17,19,21,23,25, and 27	unicast traffic

Prefix	Transponder
10.0.0.4/30	1
10.0.0.0/30	2
10.0.1.4/30	4
10.0.1.0/30	5
10.0.2.4/30	7
10.0.2.0/30	8
10.0.3.4/30	10
10.0.3.0/31	11
0.0.0.0/0	

Suffix	Transponder
0.0.0.2/16	13
0.0.0.2/17	15
0.0.0.3/16	17
0.0.0.3/17	19
0.0.0.4/16	21
0.0.0.4/17	23
0.0.0.5/16	25
0.0.0.5/17	27

Figure 9: Two-level routing table for switch 10.0.0.1. It will deliver an incoming packet to a specific transponder depending on the destination IP address of the packet.

proaches, as long as the switch that receives the packet can forward the packet to the correct transponder. Here, we consider a simple approach. Let us assume the upper layer application is based on TCP. A hash function based on a tuple (*src IP*, *dst IP*, *src port*, *dst port*) is implemented in each source server. Subsequent packets will follow the same path, thus avoiding packet reordering. For example, if server 16 10.0.3.5 sends a packet to server 1 10.0.0.2, according to the hash function, the packet will first be forwarded to transponder 10 on switch 10.0.0.1. Once the switch receives the incoming packet, it determines the next hop based on the two-level routing table. Each entry in the first-level prefix will be searched first, until the second entry, 10.0.0.0, is found. Then, the packet will be forwarded to transponder 2. From there, the packet will be delivered to destination 10.0.0.2.

For the inter-rack traffic, the only difference is that the packet will traverse one more level of POXN/MPs. For instance, let us assume that server 128 10.7.3.5 in pod 7 sends a packet to the server 1 10.0.0.2 in pod 0. In the first stage, according

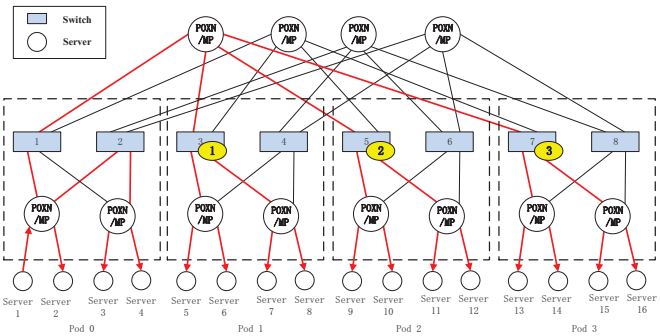

Figure 10: Sandwich Tree broadcast.

to the hash function, the packet will be forwarded to transponder 10 on switch 10.7.1.1. Then, the longest-matching prefix search does not yield a terminating prefix, so the second-level suffixes will be searched. Based on this search, transponder 13 is found, and the packet is forwarded to it. Through the upper-level POXN/MPs, the corresponding transponder 13 on switch 10.0.1.1 will receive the packet and forward it to transponder 2. From there, the packet will be delivered to destination 10.0.0.2.

Second, we will discuss how multicast and broadcast traffic is transferred in the Sandwich Tree.

As we can see from Fig. 10, server 1 wishes to send broadcast traffic to all other servers in the Sandwich Tree. In order to fully utilize the characteristics of the multicast plane for POXN/MPs or POXN/MP-BPs, we must use other switches to relay broadcast traffic to local servers within a pod because this will reduce the number of duplicate packets on the switch side. For instance, in Fig. 10, both switch 1 and switch 2 receive the packet. Then, switch 1 forwards the packet through its upper transponders, while switch 2 forwards the packet through its lower transponders. If switch 1 is selected to forward the packet to servers 3 and 4, it must duplicate one more packet and then forward it through its lower transponder. The original packet will continue traversing another POXN/MP on the top and arrive at switches 3, 5, and 7. Here, each switch will make a duplicate packet, which will be sent through the switch's lower transponder. In this way, only three duplicate packets are generated in the switches. Regarding multicast traffic, we use the same approach proposed in [8] . A centralized manager is added to collect IGMP join requests forwarded by each switch and then to assign a forwarding state to each switch, which finally delivers the multicast traffic to the interested servers.

Compared to the same scale of a traditional fat tree, server 1 sending a broadcast packet requires 14 duplicate packets, since each switch requires n-1 duplicate packets to send a packet through n ports. In Fig. 11, we can see all 14 packet duplicates are denoted by circles. In terms of broadcast and multicast traffic, the Sandwich Tree saves more bandwidth than the traditional fat tree structure by reducing the number of duplicate packets. The bigger the scale of the DCN, the more bandwidth will be saved.

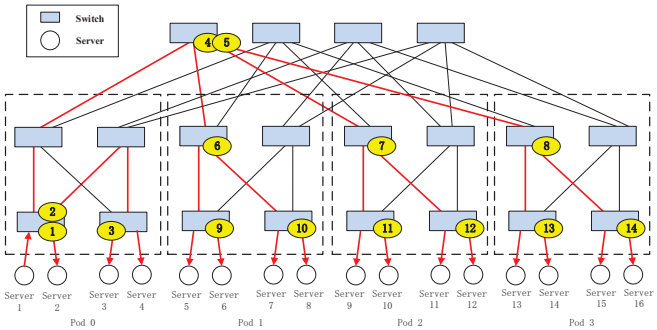

Figure 11: Traditional fat tree broadcast.

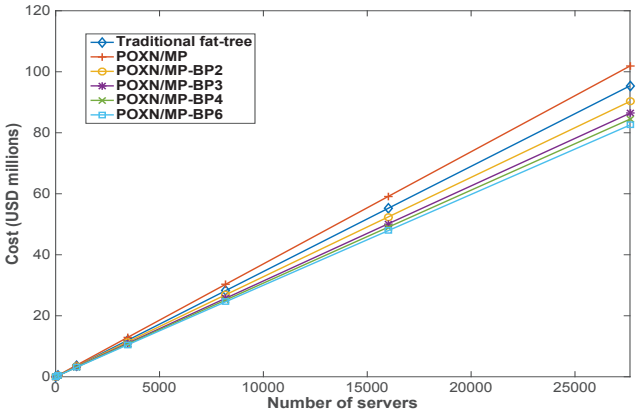

Figure 12: Cost comparison among the traditional fat tree topology, the POXN/MP, and the POXN/MP-BP, with different numbers of bundled ports.

3.4. Benefits in terms of cost and power consumption

In building current DCNs, minimizing cost and power consumption is critical. Thus, in this section, we will discuss the cost and power consumption of the Sandwich Tree. Six general formulas will be developed to represent the cost and power consumption of all of three topologies: the Sandwich Tree using POXN/MPs only, the Sandwich Tree using POXN/MPs and POXN/MP-BPs, and the traditional fat tree topology. Finally, the advantages of the Sandwich Tree with POXN/MP-BPs over the other two topologies are clearly shown in tables and graphs.

3.4.1. Cost advantage

In order to illustrate the huge cost savings made possible by the Sandwich Tree topology, we will develop a general formula for all three topologies. In this model, the Capital Expenditure (CAPEX) per link is not adopted because of the asymmetry of the aggregation switch. Instead, the CAPEX is calculated for the whole system worth being investigated for the same number of servers. First, we examine the main components in the fat tree topology, which comprises $(k/4)^3$ servers and $(k/2)^2$ k-port core switches in total, $k/2$ k-port aggregation switches, and $k/2$ k-port access switches existing in k pods [10]. We also calculate the main components for the other two topologies. Based on the main component quantities outlined in Table 2, plus their unit costs[16, 24], we can develop three CAPEX formulas for three kinds of topologies. For the fat tree topology, the total cost $C_{fattree}$ can be represented by:

$$C_{fat\ tree} = \frac{5k^3}{4} \times 450 + \frac{3k^3}{2} \times 200 = 862.5k^3 \qquad (1)$$

For the POXN/MP topology, the total cost $C_{POXN/MP}$ is as follows:

$$\begin{aligned} C_{POXN/MP} &= \frac{k^3}{2} \times 450 + \frac{3k^3}{4} \times 350 + \frac{3k^3}{4} \times 525 \\ &\quad + k^2 \times 480 + \frac{3k^3}{4} \times 40 \\ &= 911.25k^3 + 480k^2 \end{aligned} \qquad (2)$$

For the POXN/MP-BP replacement in the fat tree topology, the total cost $C_{POXN/MP-BP}$ is as follows:

$$\begin{aligned} C_{POXN/MP-BP} &= \frac{k^3}{4} + \frac{k^3}{4b} \times 450 + \frac{k^3}{2} + \frac{k^3}{4b} \times 350 \\ &\quad + \frac{3k^3}{4} \times 525 + k^2 \times 480 + \frac{k^3}{4b} + \frac{k^3}{2} \times 40 \\ &= 701.25k^3 + 210\frac{k^3}{b} + 480k^2 \end{aligned} \qquad (3)$$

Based on the developed formulas, we develop the above Fig. 12. We choose different pod values (k = 4, 8, 16, 24, 32, 40, and 48) and then calculate the corresponding number of servers which is $k^3/4$. The x-axis represents the number of servers in the datacenter, which ranges from only 16 to 27,648, while y-axis represents the total expenditure (in million USD) for building DCNs of different scales. POXN/MP-BPb is used to denote different cases of the number of bundled ports, and b here represents the same thing as b in the above formulas. For example, if k is 48 and b is 4, this means that four ports will be bundled together on each switchs NIC, the number of switches in each pod will be $k/2b = 6$, and each switch will have $4 \times k/2 = 96$ ports connected to the 24 first-level couplers through 24 different links.

The line marked by plus sign representing the POXN/MP increases sharply with the increasing number of servers, reaching approximate 102 million USD. The fat-tree topology, denoted by the line marked by diamond, illustrates a total cost that increases slightly less than that of the POXN/MP, reaching around 95 million USD (thus saving about 6.5 million USD) when the datacenter accommodates 27,648 servers (48 pods). In accordance with expectations, the POXN/MP-BP2 achieves significant cost savings, costing as much as 5.1 million USD less than the fat tree topology. With the increasing number of bundled ports on the switch side, the POXN/MP-BP performs better and better. More specifically, expenditure savings can reach 12.9 million USD and 13.8 million USD for POXN/MP-BP4 and POXN/MP-BP6, respectively. Furthermore, when b increases from 2 to 6, total expenditures decrease from 90.3 million USD to 81.6 million USD. These results match expectations, since,

Table 2: Comparison of main component quantities among three topologies

Topology	Quantity of Components					
	Line-card&switch fabric	10G LR (1310nm)	10G LR (1550nm)	10G tunable LR (1550nm)	Coupler	AWG and WFFOC
Fat tree	$5k^3/4$	$3k^3/2$	0	0	0	0
POXN/MP	$k^3/2$	0	$3k^3/4$	$3k^3/4$	k^2	$3k^3/4$
POXN/MP-BP	$k^3/4 + k^3/4b$	0	$k^3/2 + k^3/4b$	$3k^3/4$	k^2	$k^3/2 + k^3/4b$

Table 3: Unit power consumption of each element

Device	Unit power consumption(W)	Unit price (USD)
Switch port	12.5	21600/48= 450
LR(1310 nm) transceiver	1	200
LR(1550 nm) transceiver	1.5	350
LR(1550 nm) tunable transceiver	1.5	525
Coupler	0	480
AWG or WFFOC	0	40

when the number of bundled ports increases, fewer total elements are required (except for the cases of tunable LR (1550 nm) transceivers and couplers).

3.4.2. Power consumption advantage

In terms of the power consumption, the benefits of the Sandwich Tree are also clear. One big advantage of using a passive optical coupler is that it consumes zero power by nature. Furthermore, neither the AWG not the WFFOC uses any power. The only element that increases the power usage is the long-range LR (1550 nm), which uses more power than the long-range LR (1310 nm). However, the unit difference between the two is very minor, at only 0.5 W. By contrast, the fat tree topology, with its wide deployment of electronic switches, consumes far more power than the POXN/MP-BP topology.

We develop three general formulas to determine the power consumption of all three topologies based on the number of elements in the last section and the unit power consumption of each component in Table 3. One thing should be noticed is that the POXN/MP-BP topology can consume less power by reducing the power budget for coupler fabrics, which is demonstrated in the section 3.2. This is why the formula of the POXN/MP-BP topology contains an extra coefficient 0.67.

The power consumption of the fat tree topology $P_{fat\ tree}$ can be calculated using the following formula:

$$P_{fat\ tree} = \frac{5k^3}{4} \times 12.5 + \frac{3k^3}{2} \times 1 = 17.125k^3 \quad (4)$$

For the POXN/MP, the power consumption $P_{POXN/MP}$ can be calculated as follows:

$$P_{POXN/MP} = \frac{k^3}{2} \times 12.5 + \frac{3k^3}{4} \times 1.5 + \frac{3k^3}{4} \times 1.5 = 8.5k^3 \quad (5)$$

For the POXN/MP-BP, the power consumption $P_{POXN/MP-BP}$ can be calculated as follows:

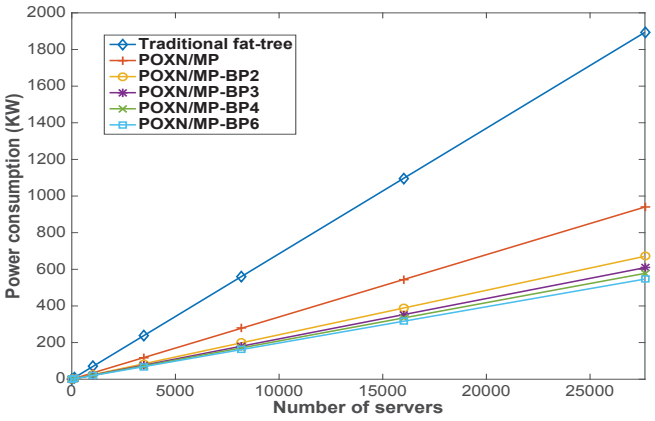

Figure 13: Power consumption comparison of the traditional fat tree topology, the POXN/MP, and the POXN/MP-BP, with different numbers of bundled ports.

$$\begin{aligned} P_{POXN/MP-BP} &= \frac{k^3}{4} + \frac{k^3}{4b} \times 12.5 + \frac{k^3}{2} + \frac{k^3}{4b} \times 1.5 \times 0.67 \\ &\quad + \frac{3k^3}{4} \times 1.5 \times 0.67 \\ &= 4.38k^3 + 3.38\frac{k^3}{b} \end{aligned} \quad (6)$$

As can be seen from Fig. 13, the results show that the POXN/MP-BP has a significant advantage over the traditional fat tree topology and the POXN/MP with respect to power consumption. For the fat tree topology, power consumption increases dramatically with the increase in servers, reaching approximately $1.89 \times 10^3 KW$ when the datacenter has 27,648 servers. The POXN/MP uses significantly less power, consuming only $0.94 \times 10^3 KW$ for a datacenter with 27,648 servers. Undoubtedly, the POXN/MP-BP has the best performance of all three topologies. It exhibits a much slower increase, when the number of servers increases by four orders of magnitude (from 16 servers to 27,648 servers). More specifically, for a large-scale datacenter containing 27,648 servers, the POXN/MP-BP6 replacement topology results in power savings up to $1.34 \times 10^3 KW$, representing a 71% reduction in the total power consumption of the fat tree topology.

Compared with the savings in the expenditure, it is clear that the power savings of the POXN/MP-BP topology over the POXN/MP and fat tree topologies are more substantial. Thus, the POXN/MP-BP replacement methodology not only saves money, but also relieves the burden of power usage within the

datacenter. Furthermore, since less power is used, more funds will be saved. At the price of 12.73 cents/KW hour [21], the power usage difference between the fat tree topology and the POXN/MP-BP6 topology results in a savings of 1.42 million USD per year, under the condition of 27,648 servers. Finally, cooling costs represent another major factor in total electricity usage, representing nearly 30% of this usage in large-scale datacenters [22]. Thus, it should be noted that, once cooling costs are taken into consideration, total saving will be greater than 1.46 million USD.

4. Numerical Results

Thus far, we have explored the general concepts of upgrading POXN/MPs to the new POXN/MP-BP design element and building Sandwich Trees using POXN/MPs and POXN/MP-BPs in a fair amount of detail. We have also investigated the routing schemes, hardware costs, and power consumptions of proposed architectures.

In this section, we will set up experiments at the device and network levels to theoretically assess our simulation models. We will first examine a single POXN/MP-BP works at the device level in a real network scenario using an OPNET modeler. We will simulate the complete protocol, including the discovery phase and the data transfer phase, in order to demonstrate the practicability and correctness of the POXN/MP-BP. Finally, we will simulate the proposed topologythe Sandwich Treeat the network level.

In order to prove the practicability and validity of the POXN/MP-BP, a six-port POXN/MP-BP is first simulated at the device level. Then, we evaluate the network performance of a Sandwich Tree with a combination of POXN/MP-BPs and POXN/MPs and the network performance of a Sandwich Tree with POXN/MPs only at the network level over two steps. The first step is to build a two-level network model with one level of POXN/MPs and one level of switches. The second step is to build a three-level network model with one level of POXN/MP-BPs, one level of switches, and one level of POXN/MPs. All simulations are simulated in the OPNET modeler and all the results are shown with a 95% confidence interval.

4.1. Device-level simulation

As with the simulation setup parameters in [18], we assume that all transmitters (including the fixed transmitter and the tunable transmitter) work at the speed of 10 Gbps. All control messages are 128 bytes, packet sizes follow an exponential distribution with a mean size of 1024 bytes, the data transfer phase lasts 144.96 μs, and a discovery phase is triggered every 20000 cycles. The simulation topology is depicted in Fig. 14.

The overhead consists of control message overhead and hardware limitations, which are shown in Fig. 4[18]. The upper bounds of the MCBDAP can be calculated as follows: 1-12/144.96 = 0.917.

As we can see from Fig. 15, the MCBDAP exhibits better performance than the MCDAP. The systems maximum bandwidth efficiency is about 0.81, which is very close to the value of the upper bound for the MCBDAP. For instance, when the load ρ is 0.81, the throughput for the MCBDAP can reach 8.09 Gbps, compared with 7.56 Gbps for the MCDAP, representing a relative improvement of approximately 7%. As a result, a six-port POXN/MP-BP can achieve 64.72 Gbps aggregated throughput, compared with 60.48 Gbps throughput for an eight-port POXN/MP. This benefit will increase as more ports become involved in the system. This result also matches expectations, since the efficiency improvement algorithm in section 2.3.1 suggested that, at some point, bundled ports transmitting and receiving packets for one another greatly reduces the mean packet queueing delay, allowing more packets to be delivered during the data transfer phase.

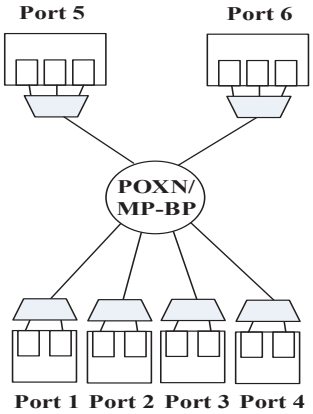

Figure 14: Simulation topology of an six-port POXN/MP-BP at the device-level.

Table 4: Time periods for different kinds of overhead

Operation	Time period
Inter-port guard interval (including laser off and on, automatic gain control [AGC], clock data recovery [CDR], and code-group alignment intervals)	2 μs
Transmission time for control messages T^D	0.1014 μs
Inter-port processing time	10 ns
Inter-frame gap time	9.6 ns
Worst-case propagation delay from port to coupler	5 μs
Tunable transmitter tuning time	4.916 μs
Per-frame service time	0.5092 μs

In addition, Fig. 16 illustrates how the increase in load to a transmitting port affects the mean packet delay in a single POXN/MP-BP. The mean packet delay shows a very steady and slow increase when the value of the load varies between 0 and 0.7, after which it experiences a substantial increase, especially when it reaches the load limit of 0.81. Compared to the POXN/MP, the POXN/MP-BP exhibits a flatter increasing trend regarding the mean packet delay. This is because a certain amount of overhead can be avoided through the benefits of the new protocol.

4.2. Network-level simulation

First, a two-tier network model with POXN/MP-BPs is simulated in order to prove the network performance (e.g., aggregate

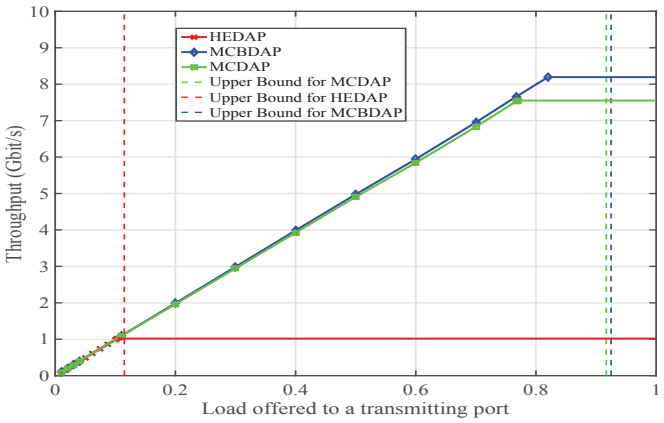

Figure 15: Throughput performance of the MCBDAP with an increase in load.

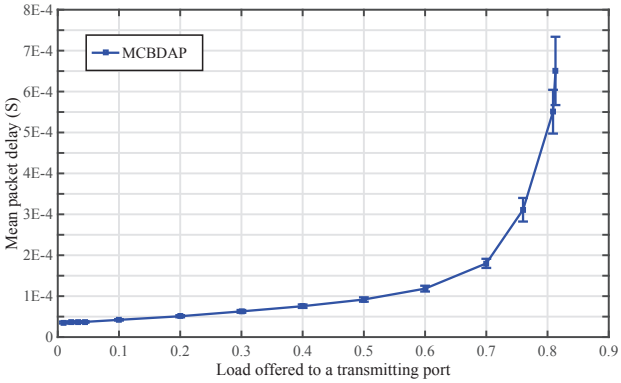

Figure 16: Mean packet delays for the MCBDAP with different offered loads ρ.

throughput) of the MCBDAP at the network level. This performance is compared to that of the POXN/MP structure. Second, a three-tier network model with a combination of POXN/MP-BPs and POXN/MPs is simulated to prove that POXN/MP-BPs and POXN/MPs are able to operate well in a multiple-tier network and to cooperate efficiently with traditional electronic switches.

4.2.1. Two-tier Sandwich Tree Modeling

We will illustrate how to integrate POXN/MPs and POXN/MP-BPs into Sandwich Trees in detail. In this subsection, we will simulate a POXN/MP-BP in a two-tier Sandwich Tree. The next subsection will show how to scale up continuously with one more level of POXN/MPs in a three-tier Sandwich Tree. The simulation topology is shown in Fig. 17. To compare performance, the same scale of POXN/MPs is simulated, as shown in Fig. 18. Since it is too complicated and impractical to simulate a 48-port POXN/MP-BP on a packet-by-packet basis, we choose a simulation topology in which two POXN/MP-BPs form subnets 1 (including server 1 to server 4) and 2 (including server 5 to server 8), respectively. Each of the two POXN/MP-BPs connects a group of bundled ports on each switch, while connecting to four normal ports on four servers. Packets arriving at eight servers still follow a Poisson distribution. According to the configuration of the Cisco Nexus

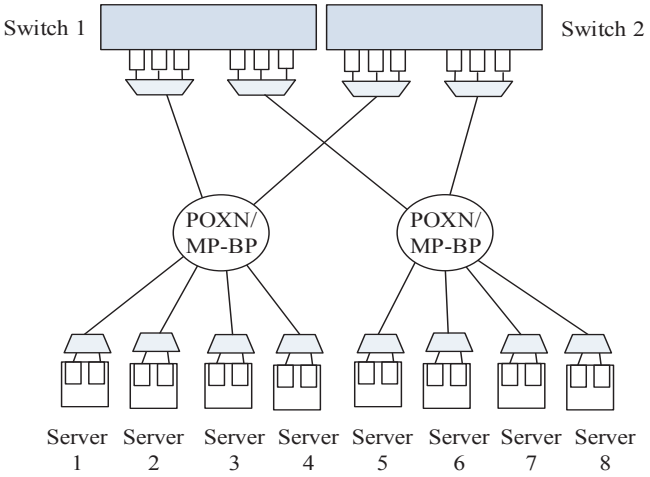

Figure 17: Simulation topology of POXN/MP-BP with eight servers.

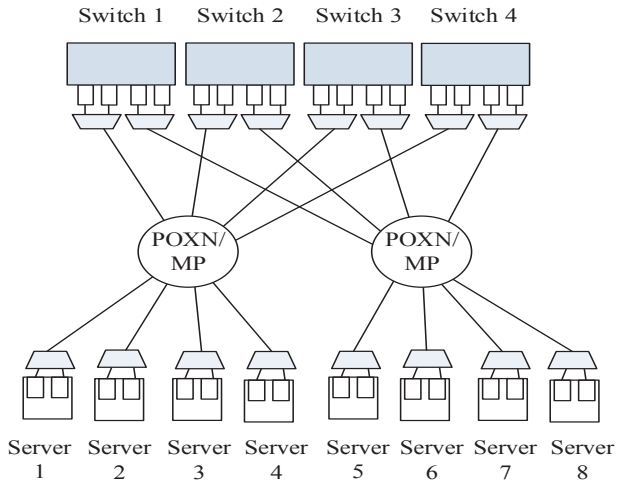

Figure 18: Simulation topology of a POXN/MP with eight servers.

3000 series switch [23], the switching time is proportional to the packet size and is less than 4×10^{-6} s.

We use an all-to-all traffic pattern, where a server distributes half of its traffic to servers connected to a different POXN/MP-BP. Further, this portion of traffic is uniformly distributed among these servers. In this setting, packets are evenly distributed over links based on a 4-tuple. This achieves the equivalent effect of employing Equal Cost Multi-Path (ECMP) at a flow or even finer packet granularity. Thus, we believe that the results here also hold for general ECMP-like routing, such as VLB [11, 26], which is widely used in DCNs. Similar to the fat tree addressing form [10], the server follows the IP address format 10.*pod*.*coupler*.*ID*, and the switch follows the format 10.*pod*.*switch*.1. The last two quad-dotted pieces of information−coupler, ID, switch, and 1are added into the control messages for each server and switch, as relevant. Since, in this scenario, no inter-pod traffic is introduced, the *pod* information is not needed.

End-to-end (ETE) delay performance must be analyzed, s-

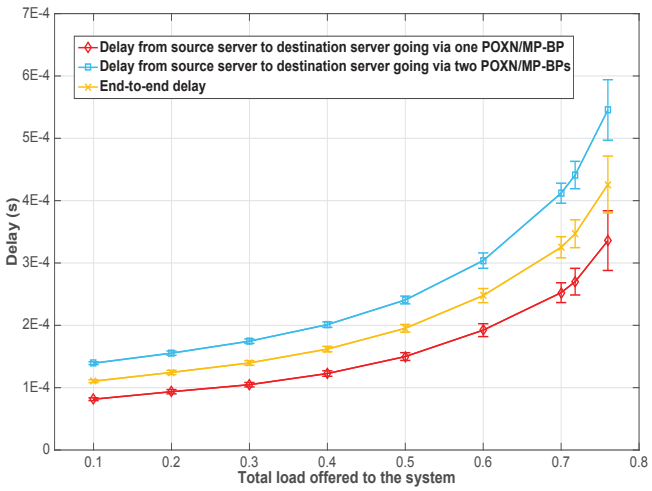

Figure 19: Delay from source server to destination server going via one POXN/MP-BP; delay from source server to destination server going via two POXN/MP-BPs; and end-to-end delay for a two-tier Sandwich Tree topology (shown in Fig. 17) with different loads offered to the system.

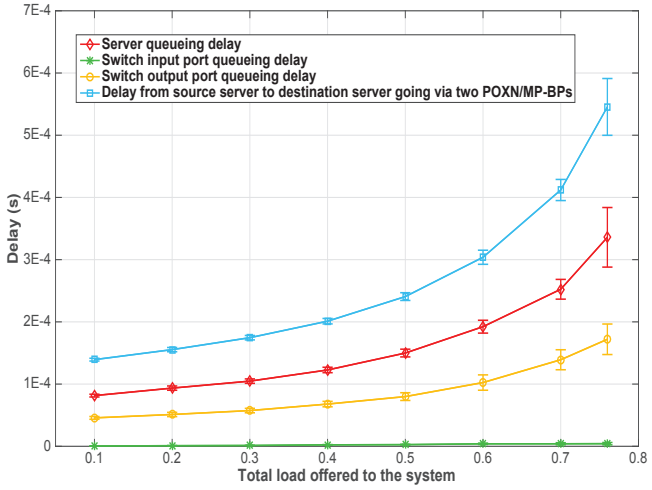

Figure 20: Delay from source server to destination server going via two POXN/MP-BPs include three main kinds of delays, which are server queueing delay, switch input port queueing delay, and switch output port queueing delay.

ince this is an important factor that can measure the quality of service (QoS) of a network. We measure only packet queueing delay, switching delay, and ETE delay.

Fig. 19 shows three categories of delays: internal traffic delay, external traffic delay, and total traffic ETE delay. Internal traffic refer to traffic transmitted among servers 1, 2, 3, and 4 or among servers 5, 6, 7, and 8. External traffic refers to traffic relayed by the switch. For example, server 1 might send traffic to server 5, 6, 7, or 8, which would need to be relayed by switch 1 or 2 (shown in Fig. 17). ETE traffic refers to all traffic (including internal and external traffic) transmitted across the entire network.

All three kinds of delays show an increasing trend with the increase in loads offered to each server. As can be seen in Fig. 19, the closer they are to the load limit of 0.761, the faster their corresponding delays increase. As the load changes from 0.1 to 0.761, internal traffic always experiences at least average delay, while external traffic suffers the longest average delay. This result can be expected, since internal traffic does not need to undergo a switch before reaching its destination servers (e.g., server 1 sends traffic to server 2, 3, or 4 in Fig. 17). By contrast, external traffic will experience more delay, including queueing delay, switching delay, and propagation delay due to switching to other POXNs (e.g., server 1 sends traffic to servers 5, 6, 7, or 8 in Fig. 17). In addition, ETE delay illustrate the overall performance of all internal and external traffic in the network. 4.8×10^{-4} s.

To verify the simulation results, we measure the three kinds of delays that dominate ETE delay: server queueing delay, switch queueing delay, and switch delay (Fig. 20). Propagation delay are not drawn in the figure, since these are represented by a constant value. Furthermore, twice the propagation delay plus the server queueing delay in Fig. 20 is equal to the internal traffic delay in Fig. 19. It is interesting to note that, near the load limit in Fig. 20, both the server queueing delay and the ETE delay show drastic increases, while the switch queueing delay still retains its slow, increasing trend. This is because of our traffic pattern setup, in which half of the traffic generated at each server will traverse the switch to reach its destination, resulting in the switch having a lower load than the source server.

To demonstrate our thinking, we also measure the average link utilization for all kinds of links, as shown in Fig. 21, including two uplinks and two downlinks. The uplink from the source server to the POXN/MP-BP carries external traffic sent from the source server to the bundled port on the switch and carries internal traffic from source server to the other servers connected by the same POXN/MP-BP. These two kinds of traffic travel through the POXN/MP-BP, and then external traffic is carried by the uplink from the POXN/MP-BP to the bundled port. The link from the bundled port to the POXN/MP-BP carries the external traffic already relayed by the switch, which will reach its destination after traveling through another POXN/MP-BP. The link from the POXN/MP-BP to the normal port carries two kinds of traffic, including internal traffic sent from source servers connected by the same POXN/MP-BP and external traffic sent from other source servers connected by other POXN/MP-BPs. This is why the lines marked by cross and square have nearly double the link utilization of the lines marked by diamond and asterisk.

Based on the measurements and analyses of the packet delay for these three different kinds of traffic, the main components of end-to-end delay, and the link utilization for the four different kinds of links, we can conclude that the introduction of one more tier in the new two-tier Sandwich Tree structure does not become the bottleneck, while the source server reaches its system limit with the increasing load.

Per our expectations, the switch side link utilization is nearly half of that of the source server utilization. This may explain why queueing delay on the switch maintain a slow increase while the ETE and queueing delay on the server show dras-

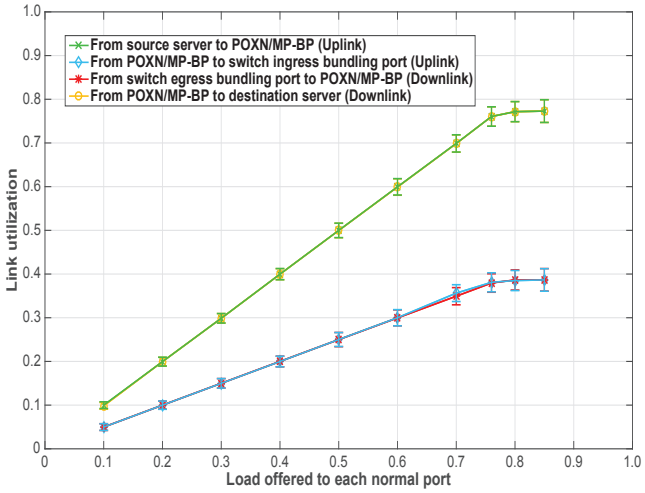

Figure 21: All four kinds of link utilizations with different loads offered to each source server.

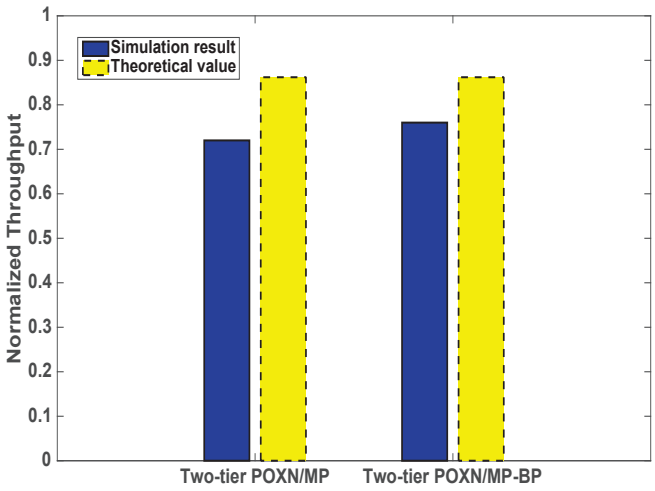

Figure 22: Normalized aggregate throughput for a two-tier POXN/MP and a two-tier POXN/MP-BP.

tic increases in Fig. 20. Compared with the queueing delay for internal traffic, external traffic experiences less packet delay when it traverses another POXN/MP-BP. This further explains why, in Fig. 19, the external traffic delay is not double that of the internal traffic with the introduction of another tier of POXN/MPs.

Finally, we measure aggregated throughput as the sum of the traffic transfer rate through servers 1 through 8, which reaches about 60.88 Gbps for the unicast plane of the POXN/MP-BP. For the internal traffic, the per-port maximum efficiency for the unicast plane is still 0.917. By contrast, for the external traffic, the per-port maximum efficiency for the unicast plane is $1 - (12 + 12 + 4)/144.96 = 0.807$ ($12\ \mu$s is the overhead for the POXN/MP-BP, and $4\ \mu$s is the switching latency). Since each server generates traffic that has an equal likelihood of being internal or external traffic, the overall per-port maximum efficiency for the unicast plane is 0.862, according to a weighted arithmetic mean.

Fig. 22 depicts the difference between the Sandwich Tree built with POXN/MP-BPs and that built with POXN/MPs in terms of aggregate throughput as a percentage of ideal aggregate throughput (i.e., normalized aggregate throughput). If all terminals are capable of sending traffic at a full rate, the value will be 1. From the simulation results, we see that the two-tier POXN/MP-BP topology achieves a higher normalized throughput of 0.76.

4.2.2. Three-tier Sandwich Tree Modeling

Based on the last subsection, we add a core layer on the top of two-tier Sandwich Tree to create a three-tier Sandwich Tree architecture in order to demonstrate that POXN/MP-BP and POXN/MP can alternately substitute switches and operate efficiently and independently in a scaled-down network.

Accordingly, to build this two-pod, three-tier Sandwich Tree structure, we introduce another pod containing eight more servers. The simulation topology is depicted in Fig. 23. In each pod, each switch uses its lower ports to connect to eight servers

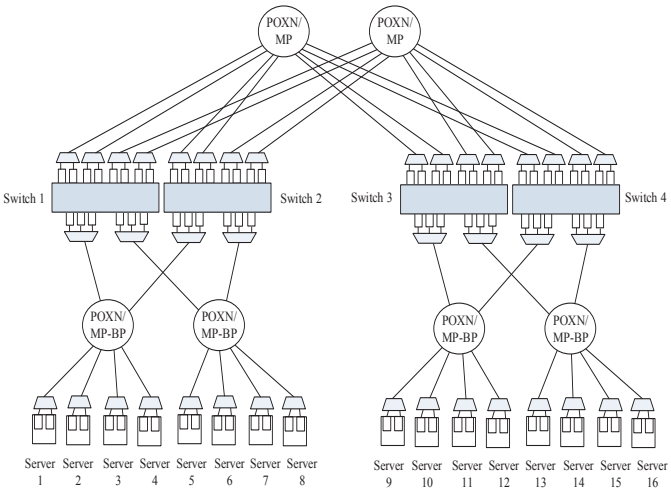

Figure 23: The three-tier simulation topology is a two-pod network composed of four POXN/MP-BPs, two POXN/MPs, and four switches.

through the POXN/MP-BP interconnection, while connecting with the other three switches through the POXN/MP interconnection. We compare this model with the three-tier Sandwich Tree structure built using switches and POXN/MPs only (without POXN/MP-BPs), which is shown in Fig. 24.

All the settings are the same as those in the two-tier simulation. We still measure average intra-pod traffic delay, inter-pod traffic delay, and total traffic ETE delay. In the first scenario, we evaluate unbiased traffic, which means that each server has a 1:1 proportion of inter-pod and intra-pod traffic to send.

Fig. 25 reveals that the three-tier network built through the combination of POXN/MP-BPs and POXN/MPs exhibits nearly the same increasing trend for all three kinds of delays in the network. It is good to know that the introduction of another layer of POXN/MP does not have a great impact on network latency. At the same load of 0.72, the average inter-pod traffic delay only increases by approximately 1.2×10^{-4} s compared to that of the two-tier Sandwich Tree. The path length

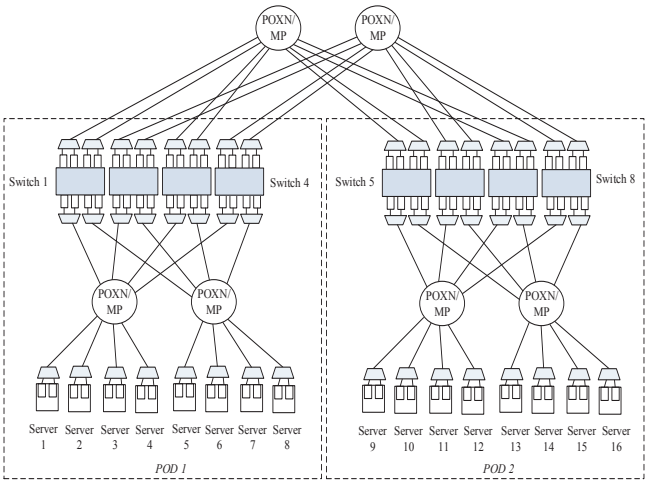

Figure 24: The three-tier simulation topology is a two-pod network composed of six POXN/MPs and eight switches.

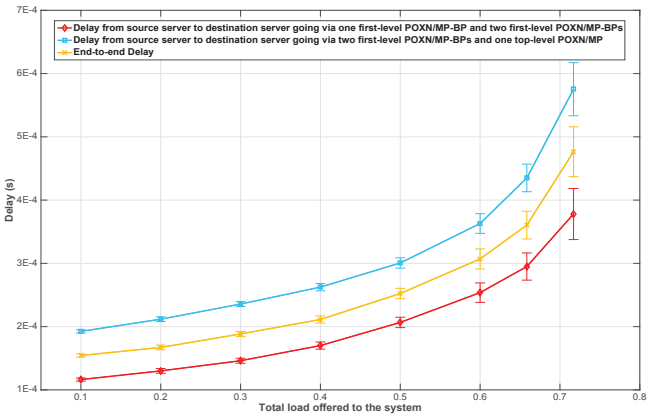

Figure 25: Delay from source server to destination server going via one first-level POXN/MP and two first-level POXN/MP-BPs; delay from source server to destination server going via two first-level POXN/MP-BPs and one top-level POXN/MP; and end-to-end delay in a three-tier Sandwich Tree topology with different loads offered to the system.

difference shows that this increase is composed of three components: one queueing delay in the switch, one switch latency, and one propagation delay. From the simulation results, we can determine that top-level POXN/MPs connected by upper switch ports yield an average throughput of 3.57 Gbps and a queueing delay of around 7.5×10^{-5} s. First, the average throughput of 3.57 Gbps means that half of the traffic is external. This agrees with our traffic pattern setup. Furthermore, the 7.5×10^{-5} s queueing delay agrees with the simulation results for the mean packet delay for the POXN/MP when the load is 0.36 at the device level [9]. Since upper switch ports run another, independent protocol for the POXN/MP, matching values are expected. Compared to the external traffic in the two-tier Sandwich Tree, in addition to the extra queueing delay at the top-level POXN/MPs, inter-pod traffic will experience two more propagation delays between the two pods through the POXN/MP and one more switching latency in the switch within the destination pod. These propagation and average switching delays are 10^{-5} s and 4.45×10^{-5} s, respectively. The sum of the three delays agrees with the increase in inter-pod traffic delays 1.2×10^{-4} s, further demonstrating the validity of our results.

In terms of normalized throughput, the addition of the top tier only decreases the performance by about only 3.2%. In comparison with the POXN/MP-only architecture, the Sandwich Tree containing both POXN/MP-BPs and POXN/MPs results in a higher normalized throughput, achieving a throughput of about 0.717, with a relative percentage difference of around 5.7.

For the internal traffic, the per-port maximum efficiency for the unicast plane is still 0.917. By contrast, for the inter-pod traffic, the per-port maximum efficiency for the unicast plane is $1 - (12 + 12 + 12 + 4 + 4)/144.96 = 0.696$ (12 μs is the overhead for the POXN/MP-BP, and 4 μs is the switching latency). Since each server generates traffic that has an equal probability of being inter-pod or intra-pod traffic, the overall per-port maximum efficiency for the unicast plane is 0.779, according to the weighted arithmetic mean. Compared with this value, the simulation results yield a relative difference of about 7.9 %.

In the second scenario, we evaluate another traffic pattern: that of biased traffic. This means that the servers have more rack-level shuffle traffic than internal traffic. This kind of situation is commonly seen in campus and enterprise datacenters. One representative application is that of VM migration, which happens when network operators are trying to balance the load among racks. In this scenario, each server generates biased traffic following a series of proportions ($2:1$, $3:1$, $5:1$, and $10:1$) for inter-pod traffic and intra-pod traffic.

As we can see from Fig. 26, all of the traffic delays exhibit more rapid increases than the unbiased traffic in scenario 1. This is because the transmitting ports that finish sending their unicast traffic will suffer longer idle times due to the unbalanced traffic before the next cycle of data transfers, thus lowering the utilization of bandwidth and increasing the server queueing delay. In addition, ETE delay are very close to inter-pod traffic delay, since an overwhelming majority of traffic flows tend to be delivered from the source pod to other pods. Therefore, inter-pod traffic delay dominate the ETE delay for all kinds of traffic.

In accordance to our expectations, the combination of POXN/MP-BPs and POXN/MPs outperforms the architecture consisting of POXN/MPs only, since the former has advantages in terms of its protocol's algorithm efficiency (details in 2.3.1), which significantly reduce the amount of idle time required in the uncast traffic transfer phase.

Fig. 27 shows that a Sandwich Tree using POXN/MP-BPs and POXN/MPs achieves a higher normalized aggregate throughput than a Sandwich Tree using POXN/MPs only under different proportions of inter-pod and intra-pod traffic, especially for biased traffic. At a proportion of $10:1$, the mixed topology still has a 0.64 normalized aggregate throughput, in contrast with a 0.56 throughput for the POXN/MP-only architecture. It is important to note that, in the first case, the 1:1 proportion is equal to the first simulation scenario (i.e., unbiased traffic). We can conclude that the introduction of POXN/MP-BPs is better able to support a scaled-down network with multiple tiers than the POXN/MP-only solution. Furthermore, the higher traffic

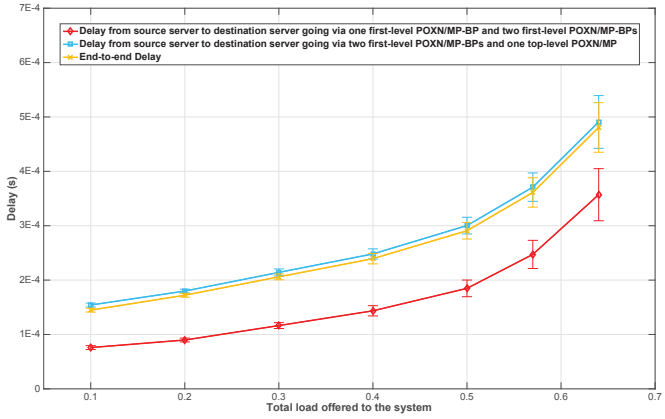

Figure 26: Delay from the source server to the destination server via one first-level POXN/MP-BP and two first-level POXN/MP-BPs; delay from the source server to the destination server via two first-level POXN/MP-BPs and one top-level POXN/MP; and ETE delay with different loads under a biased traffic scenario with a proportion of 10 : 1.

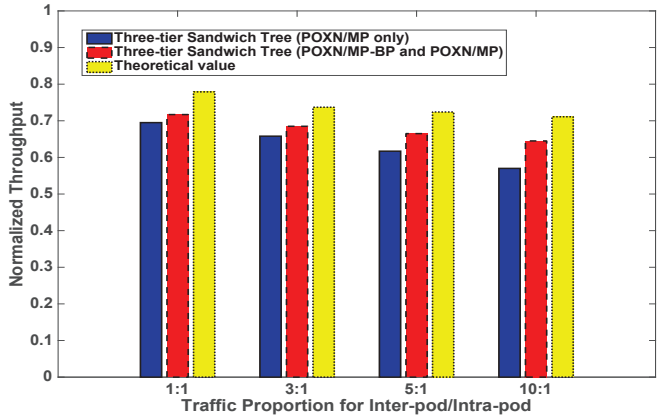

Figure 27: Comparison of the normalized aggregate throughputs for different proportions of inter-pod and intra-pod traffic for a combination of POXN/MPs and POXN/MP-BPs and POXN/MPs only in a three-tier Sandwich Tree structure.

proportion is, the better the Sandwich Tree using POXN/MP-BPs and POXN/MPs will perform.

5. Related Work

In this section, we highlight our contributions by comparing this study with the existing ones. Our main contributions can be summarized in three aspects: 1) We proposed POXN/MP-BP which enables bundled ports by changing the physical interconnection, which does not only save cost and power consumption by reducing multicast transceivers, but further lower power budget for transmitters by reducing the number of input/output ports for the coupler fabric; 2) We explored the benefits of bundled ports introduced by POXN/MP-BP which provides much more flexibility in certain scenarios, thus improving channel efficiency for both biased and unbiased traffic; and 3) We proposed a new DCN structure called Sandwich Tree, where POXN/MP and POXN/MP-BP are alternatively placed in different network tiers; therefore, the total facility cost and power consumption is drastically reduced. In what follows, we discuss works related to these three aspects.

5.1. Passive optical device in DCN

Several recent works have discussed how to utilize passive optical devices rather than electronic switch and active optical switch to serve as the core switching fabrics within a DCN. One major proposal is the use of AWG and AWGR as the key component, which enables different sizes of AWGR for intra-rack and inter-rack transmission [27]. One problem of this architecture is that large port count of AWGR may not be commercially available. Another major proposal is the use of POXN and POXN/MP based on passive optical fabrics [17, 18]. The drawbacks of these proposals are the same type of port for each host, which makes it difficult to adapt to the real scenario of DCN, such as traffic aggregation for ToR switch at the access level. In our proposal, POXN/MP-BP introduces a mechanism for bundled ports, which reduces the facility cost by reducing transceiver for multicast traffic and increases the power consumption saving by reducing the inputs/outputs of passive optical fabrics considering the same system capacity compared to POXN/MP.

5.2. Algorithm improvement

POXN/MP enables efficient transmission of unicast and multicast traffic pattern through two different planes without contention. However, POXN/MP has only been explored at device-level, which has not considered the real scenario in a DCN. In one previous study [28], 10 different categories of datacenters were investigated in terms of the ratio between the traffic spread across other racks and the traffic traversing within the rack. It shows clearly that at least 60% of traffic generated by servers in the campus and enterprise DCNs tends to leave the rack. In our proposal, POXN/MP-BP provides such flexibility that enables traffic aggregation functionality. More importantly, the mechanism for bundled ports can improve channel efficiency by reducing certain amounts of overhead during the data transfer phase for both unbiased and biased traffic. Such improvements make POXN/MP-BP competitive as the core design element in DCN architecture.

5.3. Power-efficient DCN

Non-blocking network designs based on electronic switches, such as Portland, VL2, and the fat tree, have been proposed to solve the bandwidth bottlenecks that exceed the percentage of the oversubscribed factor when a number of hosts in a pod would like to communicate with hosts in other pods [29]. However, these kinds of topologies also have disadvantages: They are expensive, have complex wiring, and consume significant power. Other proposals enabling optical technology such as c-through and Helios leverage optical circuit switch to offload traffic from traditional electrical network. The drawback is relatively longer reconfiguration time and circuit visit delay. Another relevant work is the work in [15], where all-optical solutions based on AWGR combines the distributed all-optical token (AO-TOKEN) and the all-optical NACK (AO-NACK) technologies to achieve collision-free transmission. The two main

disadvantages are the high latency caused by the retransmission of denied packets buffered in the host buffer and the high hardware cost caused by the wide usage of expensive active elements, such as FPGA and RSOA. Another all-optical solution [27] choose tunable transceivers and different sizes of AWGR to build a various sizes of DCN architectures. Due to the routing characteristics of AWGR, the receivable wavelength from the specific input port of the AWGR to the specific output port of the AWGR can only belong to a certain wavelength group, thus requiring an extra hop transmission. Compared to these approaches, Sandwich Tree applies broadcast-and-select fashion for the POXN/MP or POXN/MP-BP to well support for various traffic patterns such as unicast and multicast traffic, while leveraging electronic switches between two levels of POXN/MP and POXN/MP-BP to forward traffic to the destination rack or pod. Elimination of certain levels of electronic switches drastically reduce the power consumption and facility cost.

6. Conclusions

The work described in this paper has addressed the issue of building a cost- and power-efficient DCN capable of handling dynamic traffic patterns. Many designs are focused on building new DCNs using electronic switches and active optical switches. However, the former of these is inefficient in either cost or power consumption, while the latter suffers slow configuration times and high hardware costs. Furthermore, both electronic switches and optical switches are inefficient by nature when transmitting multicast and broadcast traffic, which prevents them from adapting to dynamic DCN traffic patterns.

In this paper, we propose a new DCN architecture, called Sandwich Tree. The introduction of POXN/MP-BPs allows the structure to handle a certain number of switches with different types of ports that facilitate traffic aggregation. Moreover, the Sandwich Tree using POXN/MP-BPs topologies significantly reduce hardware costs and power consumption compared to the Sandwich Tree using POXN/MPs topologies and traditional fat tree topologies. Furthermore, a new algorithm is proposed for the MCBDAP, which further improves the channel efficiency by reducing tuning and mismatch time. The new algorithm outperforms the previous algorithm in all scenarios, especially in cases of biased traffic.

Our simulation results demonstrate that the Sandwich Tree has better network performance than the Sandwich Tree using POXN/MPs only in multi-tier networks in terms of ETE delay and normalized throughput for unbiased and biased traffic.

References

[1] A. Singh *et al.*, Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network,SIGCOMM Comput. Commun. Rev., 45 (4), Aug. 2015, 183–197.

[2] N. Hardavellas *et al.*, Toward Dark Silicon in Servers,In: Proceedings of the IEEE Micro, 31 (4), 2011, 7–15.

[3] M. Horowitz *et al.*, Scaling, power, and the future of CMOS,In: Proceedings of the IEEE International Electron Devices Meeting(IEDM), 2005, 7–15.

[4] C. Kachris, Optical Interconnects for Future Data Center Networks 1st ed, Springer, New York, 2013 (Chapter 5).

[5] C. J. Sher Decusatis *et al.*, Communication within clouds: open standards and proprietary protocols for datacenter networking, Communications Magazine, IEEE, 50(9), 2012, 26–33.

[6] T. Benson *et al.*, Network traffic characteristics of datacenters in the wild, In: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, 2010, pp. 267–280.

[7] C. Raiciu, M. Handley *et al.*, Improving Datacenter Performance and Robustness with Multipath TCP, In: Proceedings of the ACM SIGCOMM 2011 conference, 41(4), 2011, 266–277.

[8] R. Mysore *et al.*, PortLand: a scalable fault-tolerant layer 2 data center network fabric, In: Proceedings of the ACM SIGCOMM 2009 conference on Data communication (SIGCOMM '09). ACM, 39(4), 2009, 39–50.

[9] A. Greenberg *et al.*, VL2: a scalable and flexible datacenter network, Commun. ACM, 54(3), 2011, 95–104.

[10] M. Al-Fares *et al.*, A scalable, commodity data center network architecture, In:Proceedings of the ACM SIGCOMM 2008 conference on Data communication(SIGCOMM '08). ACM, 38(4), 2008, 63–74.

[11] W. Ni *et al.*, Provisioning high-availability datacenter networks for full bandwidth communication, Elsevier Computer Networks, Special Issue on Communications and Networking in the Cloud, 68, Aug. 2014, 71–94.

[12] A. Rahbar and O. Yang, Contention avoidance and resolution schemes in bufferless all-optical packet-switched networks: a survey, IEEE Commun. Surveys Tutorials, 10(4), 2008, 94-107.

[13] N. Farrington *et al.*, Helios: a hybrid electrical/optical switch architecture for modular datacenters, In: Proceedings of the ACM SIGCOMM 2010 conference, 40(4), 2010, 339–350.

[14] G. Wang *et al.*, c-Through: Part-time optics in datacenters, In: Proceedings of the ACM SIGCOMM 2010 conference (SIGCOMM '10). ACM, 40(4), 2010, 327–338.

[15] R. Proietti *et al.*, Scalable optical interconnect architecture using AWGR-based TONAK LION switch with limited number of wavelengths, J. Lightw. Technol., 31(24), Oct. 2013, 4087-4097.

[16] A.Singla *et al.*, Proteus: a topology malleable data center network, In: Hotnets-IX Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, 2010.

[17] W. Ni *et al.*, POXN: A new passive optical cross-connection network for low-cost power-efficient datacenters, Journal of Lightwave Technology, 32(8), 2014, 1482–1500.

[18] Y. An *et al.*, A high-throughput energy-efficient passive optical datacenter network, Photonic Network Communications, 33(3), June 2016, 258–274.

[19] Y. Liu *et al.*, Quartz: a new design element for low-latency DCNs, In: Proceedings of the 2014 ACM conference on SIGCOMM, 44(4), Oct. 2014, 283–294.

[20] L. Chisvin and R. J. Duckworth, Content-Addressable and Associative Memory: Alternatives to the Ubiquitous RAM, Computer, 22(7), Jul. 1989, 51-64.

[21] Electricity price [Online]. Available: `https://www.eia.gov/electricity/monthly/epm_table_grapher.cfm?t=epmt_5_6_a`.

[22] N. Rasmussen, Calculating total cooling requirements for datacenters, American Power Conversion [Online]. Available: `http://www.apcmedia.com/salestools/NRAN-5TE6HE/NRAN-5TE6HE_R3_EN.pdf?sdirect=true`.

[23] Understanding Switch Latency Cisco Nexus 3000 series switches Data Sheet. Cisco Corp., [Online]. Available:`http://www.cisco.com/c/en/us/products/collateral/switches/nexus-3000-series-switches/white_paper_c11-661939.html`.

[24] Available:`www.fiberstore.com/c/10g-sfp-plus_63`.

[25] H. Liu, C. F. Lam, and C. Johnson, Scaling optical interconnects in datacenter networks opportunities and challenges for WDM, 2010 18th IEEE Symp. High Perf. Interconnects, 2010, 113-116.

[26] W. Ni *et al.*, Availability of survivable Valiant load balancing (VLB) networks over optical networks, Optic. Switch. Netw., 10(3), 2013, 274-289

[27] M. Xu, C. Liu and S. Subramaniam, PODCA: A passive optical data center architecture, 2016 IEEE International Conference on Communications (ICC), 2016, 1–6.

[28] T. Benson *et al.*, Understanding datacenter traffic characteristics, ACM SIGCOMM Computer Communication Review, 40(1), Jan. 2010, 92–99.

[29] G. Wang *et al.*, Your Datacenter Is a Router: The Case for Reconfigurable Optical Circuit Switched Paths, In: Proceedings of the 2009 ACM conference on SIGCOMM, Oct. 2009.