Novel Hybrid Optical-/Electrical-Switched Networks for Energy-Efficient Operation

---

A Dissertation

Presented to

the Faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the requirements for the Degree

Doctor of Philosophy (Computer Engineering)

by

Xiaoyu Wang

Decembter 2019

Approval Sheet

This dissertation is submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Computer Engineering)

Xiaoyu Wang

This dissertation has been read and approved by the Examining Committee:

Prof. Malathi Veeraraghavan, Adviser

Prof. Maïté Brandt-Pearce, Committee Chair

Prof. Zongli Lin

Prof. Haiying Shen

Dr. Abdeltawab Hendawi

Accepted for the School of Engineering and Applied Science:

Prof. Craig H. Benson, Dean, School of Engineering and Applied Science

Decembter 2019

Abstract

Ubiquitous Information and Communication Technology (ICT) devices and services are consuming a significant and growing portion of global power supplies. Increased energy usage causes economic and environmental problems. Communication networks consume around one fourth of the total ICT energy consumption; therefore, energy efficiency of network equipment requires immediate attention. Optical switching technologies have the potential to provide high network capacities in an energy-efficient manner. In this dissertation, we study novel hybrid optical-/electrical-switched network architectures for energy-efficient operation.

For large enterprise access links, we propose a two-wavelength design. In our design, one wavelength is used as part of a lower-rate static circuit for general-purpose IP traffic, while the second wavelength is dynamically configured into a high-rate access-link circuit for large dataset transfers whenever needed. A few provider-router ports are shared among a larger number of customers given that large dataset transfers are relatively infrequent. This leads to potential start-time delays, but results in significant power and cost savings. We compare two solutions for sharing high-speed provider ports, i.e., Immediate-Request (IR) mode solution and Advance-Reservation (AR) mode solution; the latter requires provider-side storage. Simulation results show that the AR-mode solution can achieve performance improvements over the IR-mode solution in terms of blocking probability and average response times. We also provide a differential cost-and-power comparison of the AR-with-storage mode and the IR mode to quantify the extra cost and power consumption introduced by the in-network storage needed with the AR mode.

For Data Center Networks (DCN), we propose an Optical Switch in the Middle (OSM)

hybrid electrical-packet/optical-circuit architecture. OSM features storage with the core Electrical Packet Switch (EPS), AR scheduling enabled by a multilayer SDN controller and Layer-2 multicast ability. Further, we identified Hadoop MapReduce applications as suitable for utilizing high-speed optical circuits despite their high reconfiguration delay, and proposed four Hadoop modifications for Hybrid Networks (HHN) to enable the effective operation of Hadoop in hybrid DCN architectures. Numerical results validate our hypothesis that it is feasible to achieve similar system-level and user-level performance with HHN, while simultaneously achieving power and cost savings with the hybrid network, when compared to original Hadoop on an EPS-only DCN.

Acknowledgments

I want to express my sincerest gratitude to my advisor, Prof. Malathi Veeraraghavan, for instilling in me the qualities of being a good researcher and engineer. Her infectious enthusiasm and unlimited zeal have been major driving forces through my graduate career. It has been an incredibly valuable experience to benefit from her knowledge, experience, and generosity, and a pleasure to work with someone who brings energy and intellectual curiosity to every discussion.

I thank Prof. Naoaki Yamanaka, Prof. Weiqiang Sun and Dr. Xiao Lin for their contributions and collaboration through multiple phases of this work. I truly appreciate their valuable feedback on my work.

I thank Prof. Maïté Brandt-Pearce, Prof. Haiying Shen, Prof. Zongli Lin, and Dr. Abdeltawab Hendawi for serving on my proposal and defense committee and providing constructive comments.

I would like to thank my fellow graduate students, Sourav, Fatma, Yizhe, Yuanlong, Xiang, Shuoshuo Reza, Elahe, and Fabrice — those who have moved on, those in the quagmire, and those just beginning — for their support, feedback, and friendship.

I would like to thank my parents, whose love and support are with me all along my Ph.D. journey. I especially want to thank my loving and caring husband, Yukang, and my adorable son, Nathan, who are the source of my courage and inspiration.

Finally, this work was carried out under the sponsorship of NSF OCI-1038058, OCI-1127340, CNS-1116081, ACI-1340910, and DOE DE-SC0002350 and DE-SC0007341 grants. I thank the National Science Foundation and Department of Energy for funding this research.

Contents

List of Tables

List of Figures

List of Abbreviations

ALR	Adaptive Link Rate
AM	Application Master
AR	Advance-Reservation
BCC	Blocked Call Clearing
BCQ	Blocked Call Queueing
CDC	Colorless Directionless Contentionless
DALC	Dynamic Access-Link Controller
DCN	Data center networks
DTN	Data Transfer Node
DWDM	Dense WDM
EPS	Electrical Packet Switch
GE	Gbps Ethernet
HDFS	Hadoop Distributed File System
HHN	Hadoop for Hybrid Networks
ICT	Information and Communication Technology
IP/MPLS	Internet Protocol/Multiprotocol Label Switching
IR	Immediate-Request

L1	Layer-1
L2	Layer-2
LPI	Low Power Idle
LR	Long-Reach
MAN	Metro-Area Network
MUX/DMX	multiplexer/demultiplexer
NM	Node Manager
OCS	Optical Circuit Switch
OSCARS	On-Demand Secure Circuits and Advance Reservation System
OSM	Optical Switch in the Middle
OTN	Optical Transport Network
REN	Research and Education Network
RJS	Regular Job Sets
RM	Resource Manager
ROADM	Reconfigurable Optical Add/Drop Multiplexers
RP	Router Port
SCU	STRONGEST Cost Unit
SDN	Software Defined Networks
SH	Shuffle-Heavy
SR	Short-Reach
ToR	Top-of-Rack
TXP	transponder

VM	Virtual Machine
WAN	Wide-Area Network
WDM	Wavelength-Division Multiplexing
WSS	Wavelength Selective Switch
YARN	Yet Another Resource Negotiator

Chapter 1

Introduction

1.1 Background

The energy consumption by Information and Communication Technology (ICT) equipment has been growing rapidly for the past decade and is expected to grow even faster in the future. In a 2018 article [1], the relative contribution of ICT to the total global footprint was projected to grow from about 1% in 2007 to 3.5% by 2020 and reach 14% by 2040. A significant part of the ICT energy consumption (28% in 2010 and an estimated 24% in 2020) is attributed to communication networks [1]. The power consumed by network infrastructure could increase from 300 TWh/year in 2015 to 430 TWh/year in 2025 [2]. As the number of customer devices, and correspondingly traffic volume, grow in the coming years, energy consumption has become a concern for the networking research community. Various solutions have been proposed to reduce network energy consumption [3], but there is still room for improvement.

Optical technologies have the potential to provide high network capacities in an energy-efficient manner. Optical fiber allows data to be sent at high rates over longer distances without regeneration when compared to copper cabling. For example, transmission rates of 10 Gbps, 100 Gbps, and even 400 Gbps are possible with optical fiber across distances of 80-100 km without regeneration, while with copper links, even at 10 Gbps, electrical signals can only be sent over distances of approximately 10 m. Using Wavelength-Division Multiplexing (WDM) technologies, the capacity of a single fiber can reach more than 10

Tbps [4]. In addition, optical switching devices consume less power and have lower prices when compared to their electrical counterparts. Consider an example comparison of power consumption of an Optical Circuit Switch (OCS) and an Electrical Packet Switch (EPS). A 64-port Glimmerglass OCS, which is data-rate agnostic, consumes 240 mW/port. In contrast, a 10-Gbps Ethernet (GE) EPS, e.g., the 48-port Arista 7148W consumes 13.5 W/port [5], and a 100GE switch port consumes 468 W [6]. For a cost comparison, consider the following: the per-port cost of a 16-port Glimmerglass OCS was only 0.2 SCU[1] when the cost of a 10GE electrical switch port was 2.1 SCUs and the cost of a 100GE electrical switch port was 30.36 SCUs [6].

In the past, optical networks were highly static in nature. Wavelengths were nailed up from point to point, leaving little room for change. Recently, increased flexibility and fast reconfiguration of optical circuits are offered by advanced optical technologies: (i) Colorless Directionless Contentionless (CDC) Reconfigurable Optical Add/Drop Multiplexers (ROADM) route wavelengths under software control, which enables centralized end-to-end provisioning of all-optical links, and dynamic service restoration at the wavelength level; (ii) tunable-optics based transceiver modules can be programmed to transmit or receive at any wavelength, without any manual intervention, vastly simplifying provisioning; and (iii) FlexGrid technologies allow for minimum spectral capacity allocations, thus maximizing fiber utilization. These technologies are increasingly used in optical Software Defined Networks (SDN). These networks have SDN controllers for provisioning and releasing Layer-1 (L1) optical circuits dynamically.

1.2 Problem Statement and Motivation

This research work explores the following two problems that arise from designing hybrid electrical/optical networks for energy-efficient operation.

[1]STRONGEST Cost Unit (SCU) is named after a project. One SCU corresponds to the 2012 cost of a 10GE 750-km optical transponder.

1.2.1 Dynamic large enterprise access links

Optical links are required for high-speed communications across long distances. Hence most Wide-Area Network (WAN) and Metro-Area Network (MAN) links are optical. Access links from large enterprises such as research universities and national laboratories are also optical because of their use of high communications rates such as 10 Gbps. As network operators upgrade their networks and enterprises purchase their network services, cost and power consumption are two major considerations. They are closely related to specific network architectures, since for example, IP-router port costs are greater for higher rates, and power consumption is more for higher-rate router interfaces and Long-Reach (LR) transponders [7]. Therefore, the *problem statement* of this work is to determine whether alternative network architectures can lower power consumption and equipment costs for high-rate enterprise access links.

Our *motivation* for addressing this problem comes from the Research and Education Network (REN) community. Core REN providers, such as Internet2 and ESnet, have upgraded their link rates to 100 Gbps, and regional RENs are making similar upgrades. Correspondingly, large universities and national laboratories are now considering upgrades of their access links to 100 Gbps. The main application driver for such an upgrade is large scientific dataset transfers. Scientists at universities and national laboratories use external supercomputing centers to run their compute-intensive big-data analytics and simulation software, and then have a need to transfer the generated (large) datasets back to their university clusters. The bottleneck link rate in such transfers is a determinant of file-transfer throughput. For example, to move a 10 TB dataset will require only a few minutes if the end-to-end bottleneck link rate is 100 Gbps instead of hours if the rate was 10 Gbps (assuming low-loss paths). Parallel file systems are used in clusters to sustain high I/O rates [8] and achieve close-to-100 Gbps transfer throughput.

Aggregate general-purpose traffic on access links from large universities and national laboratories was less than 3-4 Gbps on existing 10 Gbps Ethernet access links in 2014 [9]. Nevertheless, these enterprises were considering upgrades to 100GE in order to support large scientific dataset transfers.

1.2.2 Hybrid data center networks

Data center networks (DCN) commonly use fat-tree topologies with oversubscription at the higher layers. Inter-rack bandwidth is typically a fraction of the aggregate bandwidth within a rack, which is a design choice made to reduce capital expenditures (capex) and operating expenditures (opex).

While optical switching has the advantages of lower cost and lower power consumption when compared to electrical switching, its key disadvantage is that reconfiguration delays are on the order of μs-to-ms [10], which is significantly higher than the budget allowed for packet switching. The implication of this disadvantage is that dynamic optical circuits can only be used for large data transfers with transmission delays on the order of hundreds of ms, so that the overhead of μs-to-ms switch-reconfiguration time is small. As optical circuit setup delays would be intolerable for small and time-sensitive transfers, optical circuit switches can currently only be added as a complement to electronic packet switches, and not as a replacement. Therefore, the first *problem* to address in this work is to design a DCN architecture that incorporates OCSs.

The second part of this work considers the question of how applications can utilize these high-speed optical circuits despite their long provisioning times. While there are other suitable applications such as Virtual Machine (VM) migration and checkpointing, in this work, we focus on *Hadoop MapReduce applications*. Map tasks are typically run on hosts where input data blocks are stored. As the input data is spread over the cluster randomly in the distributed Hadoop file system, a subsequent shuffle phase is required to move map-task output to hosts on which reduce tasks are scheduled. This shuffle phase often requires data movement across oversubscribed inter-rack links.

One approach to solving this problem is to add OCSs to create hybrid EPS-OCS DCNs. Prior work on hybrid networks [5, 10–13] use techniques such as buffering packets at Top-of-Rack (ToR) switches to collect a sufficiently large amount of data before dynamically provisioning an optical circuit between two ToR switches. But without an application-level view of traffic demands and dependencies, circuit utilization and application performance could be poor [14]. Therefore, in this work, we take a *"cross-layer" approach* that modifies

the application to match its network traffic better to hybrid EPS-OCS networks, and dynamically configures optical circuits for the application when needed.

1.3 Hypothesis

The hypothesis of this research work is as follows: It is feasible to design novel hybrid optical-/electrical-switched network architectures for certain applications to achieve comparable performance as with conventional EPS-only networks but with cost and power savings.

There are two research challenges in this hypothesis formulation. The first challenge lies in the phrase "for certain applications." We need to identify applications that can benefit from high-speed optical circuits and utilize optical circuits efficiently, while tolerating the long circuit reconfiguration delays. The second challenge lies in the phrase "achieve comparable performance as with conventional EPS-only networks but with cost and power savings." Intuitively, the higher the available network bandwidth, the larger the performance improvement for network-intensive applications. On the other hand, the higher the available network bandwidth, the higher the cost and power consumption of network devices. Therefore, it is challenging to design hybrid networks to achieve cost and power savings without sacrificing application performance.

1.4 Dissertation Organization

This dissertation is organized into 6 chapters. Background, motivation, and a summary of the key contributions are provided in this chapter.

Chapter 2 presents a two-wavelength design for high-speed large-enterprise access links, where a second wavelength is used to dynamically connect to one of the shared high-speed provider IP-router ports for large dataset transfers. The goal of this design is to lower power consumption and equipment costs without having significant impact on performance. The design is described and compared with a conventional single-wavelength solution. The start-time delay performance is evaluated by modeling the access network as an $M/M/N/K/K$ queueing system. Cost-and-power analysis is conducted to quantify the savings of our dynamic design over the conventional solution.

Chapter 3 presents an alternative solution for sharing high-speed provider IP router ports in the two-wavelength design presented in Chapter 2, i.e., Advance-Reservation (AR) mode with provider-side storage. In this original design, requests to dynamically connect a second wavelength to one of the shared provider IP-router ports for large dataset transfers are handled in Immediate-Request (IR) mode. Two comparative evaluations of IR mode and AR-with-storage are presented: (i) performance comparison in terms of blocking probability and response time; (ii) analysis to quantify the extra cost and power consumption of the storage resources required in the AR-mode solution.

Chapter 4 describes a hybrid DCN architecture named Optical Switch in the Middle (OSM). OSM offers increased flexibility (when compared to prior hybrid architectures) for supporting multiple simultaneous high-speed ToR-to-ToR paths through an OCS and a core-level EPS. A multilayer SDN controller supports advanced-reservation scheduling of optical circuits, and the integration of storage in the core EPS increases the usage rate of optical circuits. This chapter also presents four modifications to Hadoop to effectively use the OSM architecture. The potential of this architecture for achieving higher compute-resource utilization while simultaneously offering users shorter job completion times is illustrated.

Chapter 5 presents a comprehensive comparative evaluation of Hadoop for Hybrid Networks (HHN) and original Hadoop on conventional EPS-only networks. The cost- and power-savings achievable in hybrid networks is quantified when compared to EPS-only networks. System-level and per-job performance is characterized to recommend parameter settings for HHN to achieve the same level of performance as with original Hadoop on EPS-only networks.

Chapter 6 summarizes our work, discusses potential future work, and concludes the dissertation.

1.5 Key Contributions

The key contributions of this work are as follows.

1. We designed a reconfigurable two-wavelength access-link architecture for large enterprises. By dynamically sharing a few high-rate provider-router ports among a larger

number of customers, the design achieves significant power and cost savings when compared with the conventional single-wavelength solution. The penalty paid by our dynamic design is the start-time delay for large dataset transfers, which is quantified and can be kept low even with a small number of shared provider IP-router ports. This work is published in the *Proceedings of the IEEE Global Communications Conference (GLOBECOM'15)* [6].

2. We designed an architecture for operating the dynamic enterprise-access design in an advance-reservation mode with storage. Comparative evaluations of the IR and AR modes characterized the benefit of the AR mode in reducing blocking probability and average response times, and the extra cost and power consumption of the AR-with-storage mode due to the required network-side storage. This work is published in the *Proceedings of the IEEE International Telecomunication Networks and Applications Conference (ITNAC'18)* [15].

3. We proposed a novel OSM hybrid packet/optical architecture for DCNs. The OSM architecture introduces integrated storage in the core EPS, and AR scheduling in a multilayer SDN controller to DCNs for the first time. These features offer applications, users and administrators higher communications speeds when needed, along with increased flexibility, when compared to previous hybrid DCN architectures. This work is published in the *Proceedings of the IEEE Conference on Communications (ICC'17)* [16].

4. We demonstrated the need for, and proposed four modifications (named as HHN) to enable the effective operation of Hadoop in hybrid DCNs. Comprehensive comparative evaluation of HHN on hybrid networks and original Hadoop on conventional EPS-only networks validates our hypothesis that it is feasible to achieve similar system-level and user-level performance with HHN, while simultaneously achieving power and cost savings with hybrid networks. This work is published in the *Proceedings of the IEEE Global Communications Conference (GLOBECOM'17)* [17] and in *IEEE/OSA Journal of Optical Communications and Networking (JOCN) 2018* [18].

In addition, contributions were made to the following: 1) A pragmatic approach of determining heavy-hitter traffic thresholds was published in a paper in the *Proceedings of 2018 IEEE European Conference on Networks and Communications (EuCNC)* [19]. 2) Hobbits: Hadoop and Hive based Internet traffic analysis was published in a paper in the *Proceddings of 2017 IEEE International Conference on Big Data* [20]. 3) Application-centric energy-efficient Ethernet with quality of service support was published in a paper in the *Electronics Letters 2015* [21]. 4) High speed 100GE adaptive link rate switching for energy consumption reduction was published in a paper in the *Proceddings of 2015 International Conference on Optical Network Design and Modeling (ONDM)* [22]. 5) Design of a time-space decoupled scheduling method for inter-DC optical networks is described in a manuscript submitted to the *IEEE Conference on Communications (ICC'19)* [23].

Chapter 2

A Dynamic Network Design for High-Speed Enterprise Access Links

2.1 Introduction

This chapter presents the design and evaluation of a dynamic high-speed access-network architecture for large enterprises. Our solution proposes the use of two wavelengths on access links: the first wavelength is used in a static circuit of lower-rate (e.g., 10GE) for general-purpose IP traffic, and the second wavelength is used in a higher-rate (e.g., 100GE) circuit that is dynamically configured for large dataset transfers whenever needed. In the provider network, each such dynamic circuit is terminated on one of N high-speed router ports that are shared between K customers, where $N < K$. A provider-network controller enables the dynamic sharing of the N provider-router ports through a reconfigurable optical platform. In addition, these shared high-speed router ports can be powered off when there are no ongoing transfers. A significant portion of this chapter is an excerpt from our published work *A dynamic network design for high-speed enterprise access links* [6]

We refer to our solution as the *dynamic solution* in contrast to the conventional *static solution* in which all customers' access links are upgraded to the higher rate, e.g., 100GE,

to carry both IP-traffic and large dataset transfers on a single wavelength. The tradeoff between power-and-cost savings of our dynamic solution relative to the static solution vs. the potential additional delay incurred for large dataset transfers (in having to wait sometimes for a free port) is evaluated in this study.

Novelty and contributions The novelty lies in our multi-wavelength design for enterprise access links, which uses a high-rate dynamic circuit for transfers of large datasets, and a lower-rate static circuit for general-purpose IP traffic. The key contributions of this work are (i) a new access-link design, (ii) a comparative evaluation of the design with a static single high-rate access link on power and cost metrics, and (iii) quantification of start-time delay penalty incurred in our design.

Section 2.2 reviews prior work. Section 2.3 describes our proposed dynamic solution, and reviews the conventional static solution. Section 2.4 presents an evaluation of the power and cost savings of the dynamic solution relative to the static solution, and also quantifies the start-time delay penalty of the dynamic solution. The chapter is concluded in Section 2.5.

2.2 Related Work

Energy-efficient techniques for networks are of increasing importance [24]. Accordingly, various technologies, ranging from hardware-level optimization to dynamic resource adaptation to novel system architectures, are being developed [25]. In Low Power Idle (LPI), a scheme adopted in the IEEE 802.3az standard, an Ethernet interface is placed in low-power mode when there are no packets to be transmitted [26]. However, LPI was reported to yield insignificant power savings on a lightly loaded 1GE university access link [27]. This is because of the overhead incurred in waking up the Ethernet interface and putting it back to sleep. Frame transmission efficiency, which is the ratio of the time spent transmitting a single frame to the sum of wake-up time, sleep time and frame transmission time, is expected to be worse on 100GE links when compared to lower-rate links due to the smaller frame transmission times. A competitor to LPI was Adaptive Link Rate (ALR) [28], in which the transmission rate of the interface is adapted to the traffic load. However, long switching times and frequent oscillations impede the ability of ALR schemes to save energy. Therefore,

our work considers a scheme that fits more into the "novel system architectures" end of the spectrum of energy-saving technologies.

The estimation of power consumption and equipment costs for new network designs requires accurate (input) values for component power and component costs. Prior work [7] [29], provided traceable and well-defined power consumption estimates for optical multi-layer network equipment. We followed the method used in this prior work to obtain updated power values from publicly available product datasheets. Cost values were presented using normalized monetary units for equipment spanning four network layers, Internet Protocol/Multiprotocol Label Switching (IP/MPLS), Ethernet, Optical Transport Network (OTN), and WDM, by Huelsermann et al. [30] and Rambach et al. [31]. As the second paper was more recent, we used component cost values from this paper in our cost evaluation.

2.3 Static and Dynamic Solutions

The *static solution* is conventionally used today. In this solution, the access-link optical circuit (carried on one wavelength) from each enterprise network terminates on a separate IP-router port within the provider's network. The port stays "always-on" allowing for file-transfer applications to be executed with no modification even for large dataset transfers. Since IP is a connectionless service, an application can simply set up a TCP connection, which does not involve any of the intermediate routers/switches, and start sending user data within IP packets.

In our proposed *dynamic solution*, two separate wavelengths are used on an access-link fiber from each enterprise for: (i) general-purpose traffic, and (ii) large dataset transfers. The wavelength for general-purpose traffic is used in a circuit that extends between a customer IP router and a provider IP router, and is static and always-on. This first link can be operated at a lower rate than the single access link in the static solution since it needs to be sized only for general-purpose traffic. The second wavelength is used in a dynamically controlled circuit for rare large dataset transfers, i.e., the circuit is setup and released dynamically only when needed. The circuit extends between a Data Transfer Node (DTN) cluster in the customer network and an IP router in the provider network. DTN clusters are part of

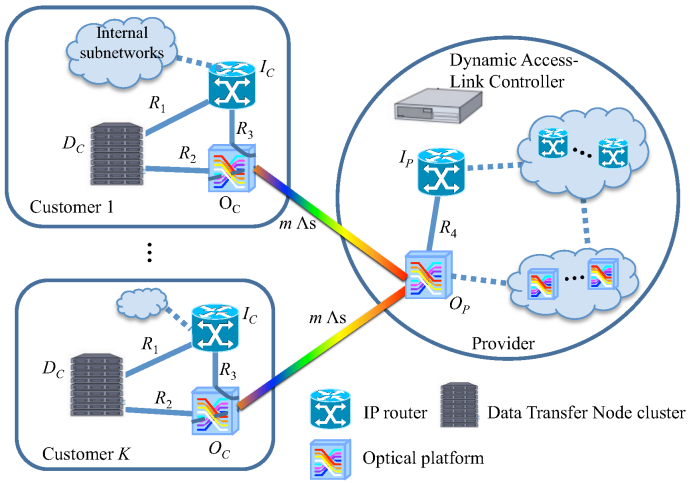


Figure 2.1: Generic system model

ScienceDMZ, which is an architecture proposed to bypass enterprise firewalls and enable high-speed network paths for large dataset transfers [32].

A *Dynamic Access-Link Controller (DALC)* (see Fig. 5.1) enables the dynamic sharing of N provider-router high-speed ports among K customers (where $N < K$), and the dynamic powering on-and-off of these ports. An application signals its need for this second access-link circuit to be established before a large dataset transfer by sending a control-plane message to the DALC. If one of the N shared provider-router ports is free, the DALC will provision the second access-link circuit by configuring the provider optical platform (see Fig. 2.1) to crossconnect the second wavelength from the corresponding customer network to the link that leads to the free provider-router port. The DALC will also add an entry in the IP rouzting table of the provider router to enable packet forwarding on to this dynamically established second access-link circuit for just the large dataset transfer flows. Similarly, a control-plane client running on the DTNs in the customer network will add an entry to the IP routing tables of the DTNs to use this dynamically established second access-link circuit for the large dataset transfer flows. If there is no available provider-router port, the controller responds to the requesting application with a delayed start time.

Fig. 2.1 illustrates a generic system model that is used to describe both the static and dynamic solutions. The model shows K customer networks, each of which consists of an IP router (I_C), an optical platform (O_C), and a DTN cluster (D_C), and a provider network

Table 2.1: Differences between static and dynamic solutions

	Feature (see Fig. 5.1)	Static Solution	Dynamic Solution
1	$O_C \leftrightarrow O_P$	$m_s = 1$	$m_d = 2$
2	DALC	No	Yes
3	$D_C \leftrightarrow I_C$	$R_{1s} = R_{2d}$	$R_{1d} = 0$
4	$D_C \leftrightarrow O_C$	$R_{2s} = 0$	R_{2d}
5	$I_C \leftrightarrow O_C$	$R_{3s} \geq R_{1s}$	R_{3d}
6	$I_P \leftrightarrow O_P$	$R_{4s} = KR_{3s}$	$R_{4d} = KR_{3d} + NR_{2d}$
7	O_C	Transponder	2 transponders and a MUX/DMX
8	O_P	K transponders	Fig. 2.2

with an IP router (I_P) and an optical platform (O_P). The DTN cluster initiates large dataset transfers while general-purpose traffic flows into the IP router I_C from other internal subnetworks. The links are marked with symbols, R_1, R_2, R_3, R_4 (R for rate) and m Λs. Using the additional s subscript for the static solution and the d subscript for the dynamic solution, Table 2.1 describes how the values are selected for these various link rates, and how the optical platforms and access links differ in the two solutions.

Row 1 of Table 2.1 shows that the static solution requires only one wavelength ($m_s = 1$) across the access link, while the dynamic solution requires two wavelengths ($m_d = 2$). To enable the dynamic sharing of the N provider-router high-speed ports, a DALC is needed only in the dynamic solution as shown in *Row 2* of Table 2.1.

The DTN cluster is connected to the customer router in the static solution, but to the optical platform in the dynamic solution, which explains why R_{1d} and R_{2s} are 0 as shown in *Rows 3 and 4* of Table 2.1. Rows 3 and 4 also show that the rate of the link from the DTN is the same in both solutions, and hence $R_{1s} = R_{2d}$. In other words, large dataset transfers to/from the DTN cluster are aggregated with general-purpose IP traffic by the IP router I_c and carried on the single wavelength circuit via the optical platform O_C on to the access link in the static solution, while in the dynamic solution, the large dataset transfers from the DTN cluster are directly fed into the second wavelength circuit at the optical platform O_C, and thus isolated from the general-purpose traffic, which uses the first wavelength circuit across the access link.

Row 5 shows that in the static solution $R_{3s} \geq R_{1s}$ because general-purpose IP traffic and large dataset transfers are merged by the IP router I_C. On the other-hand, in the dynamic

solution the link between the IP router I_C and the optical platform O_C should be sized for only general-purpose IP traffic. For example, in the dynamic solution, R_{3d} could be a 10GE link when R_{2d} is a 100GE link, while in the static solution, R_{1s} and R_{3s} could both be 100GE links given that the IP router multiplexes packets from/to the internal subnetworks with the large dataset transfer packets from/to the DTN cluster.

Row 6 shows that in the static solution, the access links from the K customers are hardwired to high-rate R_{3s} ports in the provider IP router I_P, while in the dynamic solution, K lower-rate R_{3d} ports and a smaller number N higher-rate R_{2d} ports are required in the IP router I_P.

Rows 7 and 8 show how the optical platforms required in the customer and provider networks differ in the two solutions. Since LR colored optics interfaces in the IP layer are generally more expensive than in other electrical layers [31], we assume the use of transponders that convert the gray optical signals (e.g., 1310 nm) of the IP-router interfaces to the Dense WDM (DWDM) ITU-T grid LR signals (in the C and L bands) used on the access link. Therefore, in the static solution, the optical platform required in each customer network consists of just a single transponder, and K transponders are required in the provider network.

In the dynamic solution, in each customer network, two transponders are required to convert the gray-optics signals, R_{3d} from the IP router I_C and R_{2d} from the DTN cluster switch D_C, to ITU-T grid DWDM wavelengths, and a WDM multiplexer/demultiplexer (MUX/DMX) is required to transport the two wavelengths on the same fiber. The optical line amplifiers/regenerators required on long-distance access links are common for the static and dynamic solutions, and hence are not considered in the comparison.

A design for the provider optical platform O_P required in the dynamic solution is illustrated in Fig. 2.2. Each customer's incoming composite WDM signal, after being amplified by an optical preamplifier, is demultiplexed into two signals. The R_{3d} signal carrying general-purpose IP traffic from each customer network is converted from the ITU-T grid wavelength to a gray optical signal at 1310 nm by a transponder (TXP), which is then sent to a dedicated Router Port (RP) on the provider IP router I_P. The second wavelength from each of the K customer networks is fed to a block titled *Dynamic Switch* in Fig. 2.2.

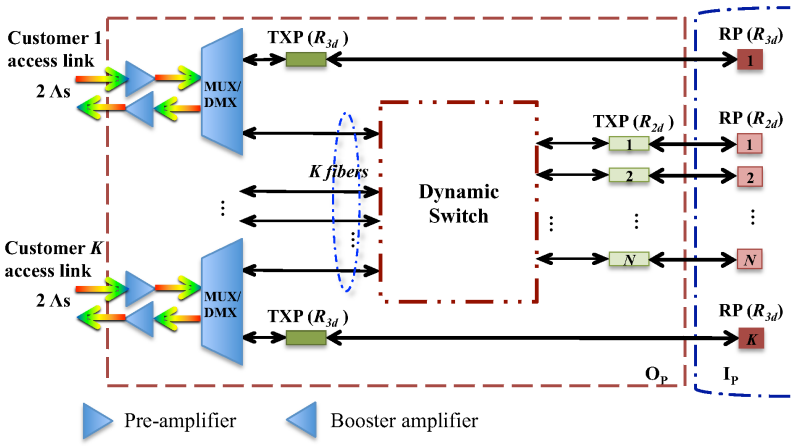


Figure 2.2: Provider optical platform O_p in the dynamic solution

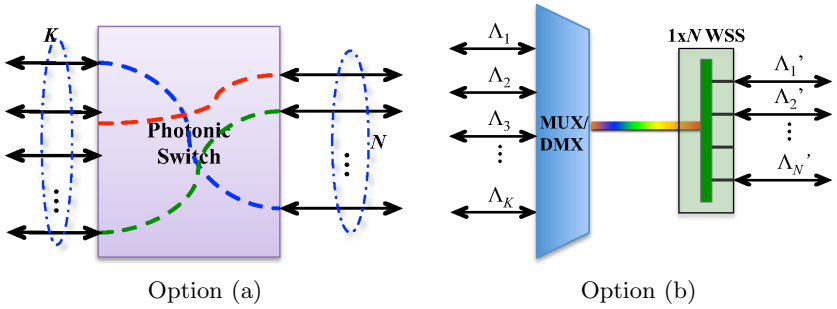


Figure 2.3: Two options for the Dynamic Switch in Fig. 2.2

This switch connects to N R_{2d} transponders that convert the ITU-T grid signals to gray optical signals, which are then conveyed to the N shared router Fports on I_P. By placing these higher-rate transponders between the optical switch and the router ports, rather than between the demultiplexers and the optical switch, we reduce the number of the higher-rate transponders from K to N ($N < K$).

Two options for the Dynamic Switch are illustrated in Fig. 2.3. In Option (a), a photonic switch is used as a reconfigurable space fabric. It can be configured dynamically to connect any incoming port to any outgoing port. In Option (b), a WDM multiplexer is used to first merge all the second wavelengths from the K customer access links onto a single fiber. Since most current transponders have tunable lasers and broadband photo-detectors [33], we assume that the DALC can dictate the particular wavelength to use for each customer R_{2d} transponder in the setup phase for the second circuit. Correspondingly, the DALC will configure the Wavelength Selective Switch (WSS) to pass through specific wavelengths (Λ_1

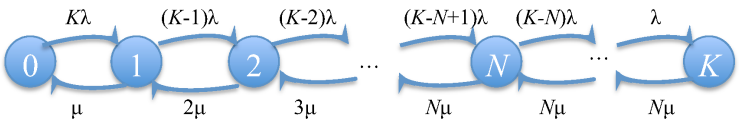


Figure 2.4: State transition diagram for the $M/M/N/K/K$ model.

to Λ_K) from customers onto selected wavelengths (Λ'_1 to Λ'_N) to the links connecting to the I_P router ports.

2.4 Evaluation

The static and dynamic solutions can be compared on power consumption and equipment costs. The dynamic solution has a disadvantage relative to the static solution in that a customer may be required to wait before starting a large dataset transfer, as described in Section 2.3. Therefore, we first characterize start-time delay in the dynamic solution, and then compare the power consumption and equipment costs of the static and dynamic solutions.

2.4.1 Start-time delay in dynamic solution

In the dynamic solution, if an application requests the setup of the high-speed access circuit for a large dataset transfer and none of the N shared ports on the provider IP router are available, then the DALC responds with a delayed start-time. The purpose of this analysis is to quantify the start-time delay under certain assumptions.

Model The system is modeled as an $M/M/N/K/K$ queueing system (a.k.a., a finite-population *Blocked Call Queueing (BCQ)* system [34]), in which calls are queued when resources are unavailable rather than rejected as in a *Blocked Call Clearing (BCC)* system. This model assumes the following: a Poisson arrival process, exponential service-time distribution, N servers, K buffers, and a finite population of size K. In effect, this model assumes that each customer issues only one request at a time for the dynamic access-link setup.

The state transition diagram for this BCQ system is shown in Fig. 2.4. The probability p_n of being in state n is

$$p_n = \begin{cases} \rho^n \binom{K}{n} p_0, & 0 \le n < N \\ \rho^n \binom{K}{n} p_0 \frac{n!}{N^{n-N} N!}, & N \le n \le K \end{cases} \tag{2.1}$$

where

$$p_0 = \left[\sum_{n=0}^{N-1} \rho^n \binom{K}{n} + \sum_{n=N}^{K} \rho^n \binom{K}{n} \frac{n!}{N^{n-N} N!} \right]^{-1} \tag{2.2}$$

and ρ, traffic load, is the ratio of per-customer call arrival rate λ to service rate μ, i.e., $\rho \triangleq \frac{\lambda}{\mu}$.

To quantitatively characterize the start-time delay D in the dynamic solution, the metric considered here is the probability $P(D > \tau)$ that a call is delayed longer than a threshold τ [35]

$$P(D > \tau) = \frac{\rho}{KU} e^{-N\tau\mu} \sum_{n=N}^{K} \left[p_n (K - n) \sum_{i=0}^{n-N} (N\tau\mu)^i / i! \right] \tag{2.3}$$

where U is system utilization, and is given by:

$$U = \frac{1}{K} \left(\sum_{n=1}^{N-1} n p_n + N \sum_{n=N}^{K} p_n \right) \tag{2.4}$$

The average number of customers being served is KU.

Numerical results Fig. 2.5 shows the probability of calls being delayed by a value greater than the threshold τ, which is set to 20 minutes, as a function of K, the number of customers. The plots correspond to different values of the number of shared provider-router ports (N), and traffic load (ρ). Together N, K, and ρ determine the system utilization U shown in (2.4), and the probability of calls receiving a start-time delay D greater than the threshold τ, given by (2.3), is determined by these parameters and mean service time $1/\mu$, which is assumed to be 800 seconds. This 800-second value was chosen assuming a mean large dataset size of 10 TB and an end-to-end bottleneck link rate of 100 Gbps. While 20 minutes is larger than the mean service time, the total delay will still be acceptable given the much longer total scientific-simulation workflow times.

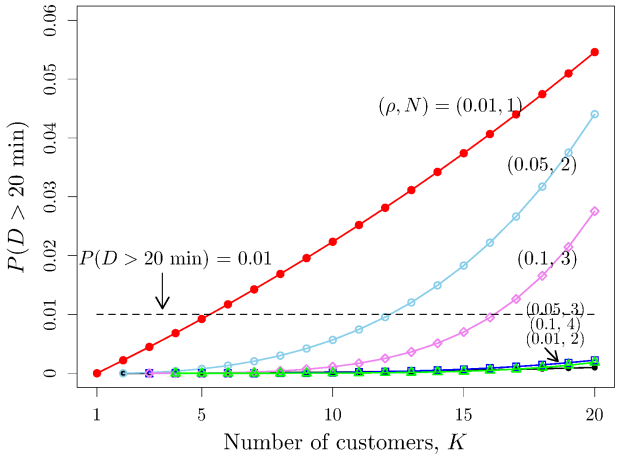


Figure 2.5: Probability of calls receiving a start-time delay greater than 20 minutes, as a function of the number of customers (K), for different values of the number of shared provider-router ports (N), and traffic load (ρ)

Fig. 2.5 shows that under light loads $\rho = 0.01$, increasing N from 1 to 2 drops the probability of calls receiving a delayed start-time greater than 20 minutes to be under 0.01 (or 1%) even when the number of customers K is as high as 20. In other words, two shared provider-router ports are sufficient to keep the start-time delay penalty small under low loads (at $\rho = 0.01$, with $\mu = 1/800$ second, the call arrival rate is roughly 1 call per day per customer). We chose the above values based on the fact that large dataset transfers are rare events that are initiated by a limited number of users. Similarly, we see that the minimum N values needed to keep the probability of start-time delay metric below 1% are 3 and 4 for traffic load values of 0.05 and 0.1, respectively. In other words, if traffic load increases by a factor of 10, e.g., by increasing the arrival rate of large dataset transfer requests, then 20 customer networks can share just 4 provider-router ports using our dynamic solution while incurring a small delay penalty.

Table 2.2 presents a sensitivity analysis for different values of the threshold τ, traffic load ρ, and service rate μ. It shows that for a relatively small service rate, i.e., μ_2, the minimum value of N needed is not sensitive to the delay threshold τ.

Table 2.2: Minimum N values needed for $P(D > \tau) < 0.01$

τ (min)	$K = 10$				$K = 20$			
	ρ_1, μ_1 †	ρ_2, μ_1	ρ_1, μ_2	ρ_2, μ_2	ρ_1, μ_1	ρ_2, μ_1	ρ_1, μ_2	ρ_2, μ_2
1	2	4	2	4	3	6	3	6
5	2	4	2	4	2	5	3	6
10	2	3	2	4	2	5	3	6
20	2	3	2	4	2	4	3	6
30	2	2	2	4	2	3	3	6

† $\rho_1 = 0.01,\ \rho_2 = 0.1$ and $\mu_1 = 1/800,\ \mu_2 = 1/8000$

Table 2.3: Notation

Symbol	Meaning
P, C (subscript)	Provider, Customer
I, O	IP router, Optical platform
s, d (subscript)	Static solution, dynamic solution
$\mathbb{P}, \mathbb{C}$ (superscript)	Power consumption, Cost
$\Phi^{\mathbb{P}}_{sC}$, $\Phi^{\mathbb{C}}_{sC}$	Power consumption and cost of the customer-network components in the static solution

2.4.2 Power and cost comparisons

This subsection presents a comparison of the power consumption and equipment costs of the static and dynamic solutions making certain assumptions.

Model We use a component-based model [7] to characterize the differential power consumption of the static and dynamic solutions. The power-consuming components in the network systems (IP router, optical platform and DTN cluster switch in customer and provider networks) can be divided into two categories, chassis and line cards. Since the power consumption of a chassis is the same in both solutions, it is left out of the comparison.

Table 2.4 lists the power consumption and costs of components, and Table 2.3 explains the notation. In the first column of Table 2.4, the symbols Φ_{sC}, Φ_{sP}, Φ_{dC} and Φ_{dP} are used to represent the components in customer (C) and provider (P) networks in the static (s) and dynamic (d) solutions. The component column in Table 2.4 lists the IP router (I), optical platform (O) and DTN cluster switch (D), at ends of links identified by their rates. For example, $I_C(R_{1s})$ denotes a line card with rate R_{1s} in the IP router of a customer network, and $O_P(R_{4s})$ denotes a transponder card with rate R_{4s} of the optical platform in the provider network, in the static solution (see links marked rates R_1 and R_4 in Fig. 2.1).

Using the symbol $\mathbb{P}$ to denote power, $\Phi^{\mathbb{P}}_{sC}$, $\Phi^{\mathbb{P}}_{sP}$, $\Phi^{\mathbb{P}}_{dC}$ and $\Phi^{\mathbb{P}}_{dP}$ represent the power

consumption of the customer-network and the provider-network components in the static and dynamic solutions, respectively. Each of these power values is determined by summing the power of individual components multiplied by the corresponding multiplicative factors shown in the fourth column of Table 2.4. For example, the power consumption across the K customer networks in the dynamic solution, $\Phi^{\mathbb{P}}_{dC}$, is given as follows,

$$\Phi^{\mathbb{P}}_{dC} = K\left(O^{\mathbb{P}}_{C}\left(R_{3d}\right) + I^{\mathbb{P}}_{C}\left(R_{3d}\right) + O^{\mathbb{P}}_{C}\left(\mathrm{MD}_{4\mathrm{ch}}\right)\right) + KU\left(O^{\mathbb{P}}_{C}\left(R_{2d}\right) + D^{\mathbb{P}}_{C}\left(R_{2d}\right)\right) \tag{2.5}$$

The first term in (2.5) has a K factor, which corresponds to the always-on access links for general-purpose IP traffic from each of the customer networks. The factor KU in the second term of (2.5), used for the dynamic access-link circuits, describes the average number of customers under service, as described in Section 2.4.1. In the dynamic solution, the DTN switch port and the optical-platform transponder within the customer network, and the optical-platform transponders and shared router ports in the provider network, can be powered-off when they are not in use. The presence of the application-to-DALC signaling phase allows these ports to be powered on as part of the dynamic access-link configuration phase. Hence instead of K, we use the factor KU. The difference in power consumption between the static and dynamic solutions is

$$\Delta^{\mathbb{P}} = \Delta^{\mathbb{P}}_{C} + \Delta^{\mathbb{P}}_{P} \tag{2.6}$$

where $\Delta^{\mathbb{P}}_{C}$ and $\Delta^{\mathbb{P}}_{P}$ separate out the power savings in the customer networks and provider network, respectively, and are defined as follows,

$$\Delta^{\mathbb{P}}_{C} = \Phi^{\mathbb{P}}_{sC} - \Phi^{\mathbb{P}}_{dC} \tag{2.7}$$

$$\Delta^{\mathbb{P}}_{P} = \Phi^{\mathbb{P}}_{sP} - \Phi^{\mathbb{P}}_{dP} \tag{2.8}$$

The cost model is similar to the power model and the only difference comes from the multiplicative factors. For instance, the provider-network cost in the dynamic solution, $\Phi^{\mathbb{C}}_{dP}$,

Table 2.4: Component power consumption and costs

	Component	Power (W) $\mathbb{P}$	MF†	Cost (SCU) $\mathbb{C}$	MF†
Static solution					
Φ_{sC}	$I_C(R_{1s})$	468	K	30.36	K
	$I_C(R_{3s})$	468	K	30.36	K
	$O_C(R_{3s})$	204	K	15	K
	$D_C(R_{1s})$	530	K	NA	-
Φ_{sP}	$I_P(R_{4s})$	536	K	35.28	K
	$O_P(R_{4s})$	204	K	15	K
Dynamic solution					
Φ_{dC}	$O_C(R_{3d})$	16.2	K	1	K
	$I_C(R_{3d})$	168	K	2.1	K
	$O_C(\mathrm{MD}_{4\mathrm{ch}})$	0 (passive)	K	0.1	K
	$O_C(R_{2d})$	204	KU	15	K
	$D_C(R_{2d})$	530	KU	NA	-
Φ_{dP}	$O_P(R_{3d})$	45(4 ports)	K	1	K
	$O_P(\mathrm{MD}_{4\mathrm{ch}})$	0 (passive)	K	0.1	K
	$I_P(R_{3d})$	298.8(4 ports)	K	33.38(14 ports)	K
	$O_P(R_{2d})$	204 (2 ports)	KU	15	N
	$I_P(R_{2d})$	936.9(4 ports)	KU	35.28	N
	$O_P(\mathrm{PS})_a$	50	1	3.25	1
	$O_P(\mathrm{MD}_{40\mathrm{ch}})_b$	0 (passive)	1	0.9	1
	$O_P(\mathrm{WSS})_b$	63	1	4	1

† Multiplicative factors

is

$$\Phi_{dP}^{\mathbb{C}} = K\left(O_P^{\mathbb{C}}\left(R_{3d}\right) + I_P^{\mathbb{C}}\left(R_{3d}\right) + O_P^{\mathbb{C}}\left(\mathrm{MD}_{4\mathrm{ch}}\right)\right) + N\left(O_P^{\mathbb{C}}\left(R_{2d}\right) + I_P^{\mathbb{C}}\left(R_{2d}\right)\right) + O_P^{\mathbb{C}}\left(\mathrm{DS}_{a/b}\right) \quad (2.9)$$

in which the multiplicative factor is N for the components $O_P(R_{2d})$ and $I_P(R_{2d})$, instead of KU for the power consumption $\Phi_{dP}^{\mathbb{P}}$. The final term in (2.9) is $O_P^{\mathbb{C}}(\mathrm{DS}_{a/b})$, which denotes the power consumption of the Digital Switch shown in Fig. 2.2. As Fig. 2.3 illustrates, there are two options for the Digital Switch, which is the reconfigurable unit. The power consumption in these two options are given by $O_P^{\mathbb{P}}(\mathrm{PS})_a$ and the sum of $O_P^{\mathbb{P}}(\mathrm{MD}_{40\mathrm{ch}})_b$ and $O_P^{\mathbb{P}}(\mathrm{WSS})_b$ (see Table 2.4), where PS stands for Photonic Switch (Option (a)), and WSS stands for Wavelength Selective Switch (Option (b)).

Input assumptions To compute numerical values for power and cost savings, we make the following assumptions. First, we choose the following link rates: $R_{1s} = R_{2d} =$ 100GE, $R_{3s} =$ 100GE, and $R_{3d} =$ 10GE. As Table 2.1 shows, all link rates for the static and dynamic solutions can be determined from these four values.

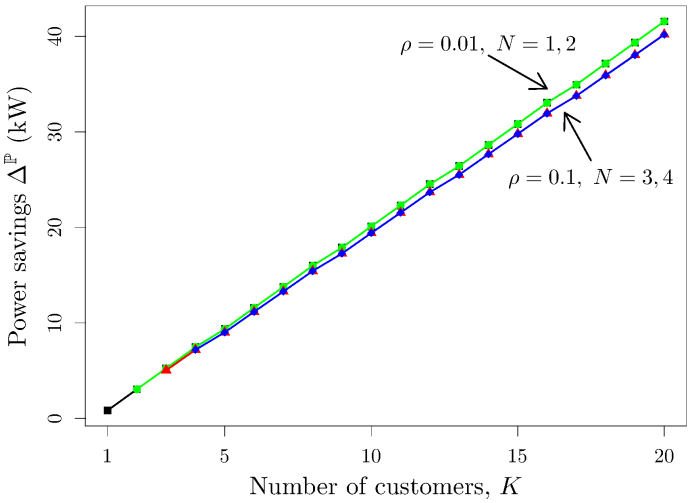


Figure 2.6: Power savings of the dynamic solution relative to the static solution for different values of the number of customers (K), shared provider IP router ports (N), and traffic load (ρ)

Table 2.4 lists our input assumptions for the power and costs of the components. The power numbers were obtained from various vendor datasheets, and compiled into a technical report, which is posted on a public Web site [36]. The cost values are obtained from a 2013 paper [31], which defines the unit SCU, as STRONGEST Cost Unit, named after the project. One SCU corresponds to the 2012 cost of a 10GE optical transponder with a reach of 750 km. All cost values are normalized to this cost.

Combining the component power and cost numbers of Table 2.4, and the U values from (2.4) for different values of K, N, and ρ, we computed numerical values for power savings and cost savings using the equations described above.

Numerical results Our delay analysis showed that to meet the requirement that $P\ (D > \tau) < 1\%$ for $\tau = 20$ min, the minimum number of shared provider-router ports, N, needed was 2, 3 and 4, for traffic load ρ values of 0.01, 0.05 and 0.1, respectively. In other words, these are the maximum N values needed to meet our delay-performance requirement.

Here, we consider the question of whether smaller values of N can lead to improved power savings. Therefore, under the low-load ρ value of 0.01, we considered the power savings, $\Delta^{\mathbb{P}}$, when N was lowered to 1 from the 2 value needed for delay performance, and for $\rho = 0.1$, we lowered N to 3 from the 4 value needed for delay-performance. Fig. 2.6 shows that the power savings are trivial (the plots for $\rho = 0.01$, $N = 1$ and 2 overlap, so do

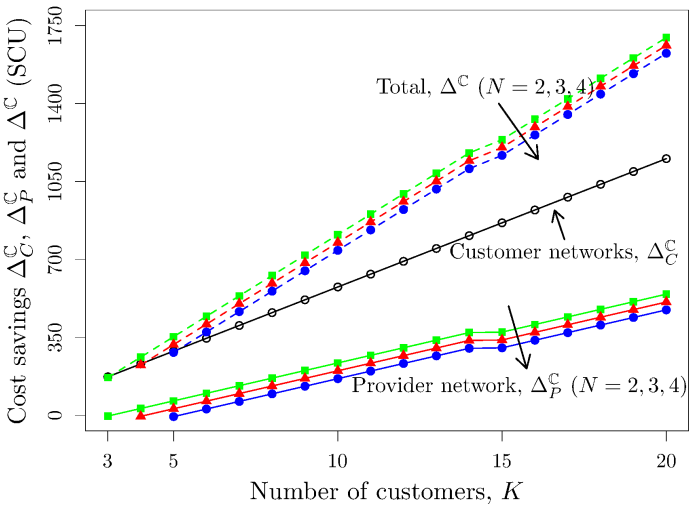


Figure 2.7: Equipment cost savings of the dynamic solution relative to the static solution for different values of the number of customers (K) and shared provider IP router ports (N)

the plots for $\rho = 0.1$, $N = 3$ and 4). Therefore, N should not be reduced from the 2 and 4 numbers required for delay performance under traffic loads of 0.01 and 0.1, respectively. In other words, given the negligible power savings, it is not worth sacrificing delay performance.

A key point illustrated in Fig. 2.6 is that significant power savings (measured in kW) is possible with our dynamic solution relative to the static solution. In addition, the average power savings per customer network, $\Delta_C^{\mathbb{P}}/K$, is more than 1.4 kW for both the $\rho = 0.01, N = 2$ and $\rho = 0.1, N = 4$ cases[1], which is twice the power consumed by the chassis of a typical edge router that supports 100GE ports [36].

Fig. 2.7 shows the consumer-network and provider-network cost savings, $\Delta_C^{\mathbb{C}}$ and $\Delta_P^{\mathbb{C}}$, respectively, and the total cost savings $\Delta^{\mathbb{C}}$. These plots show that significant equipment cost savings are feasible with the dynamic solution relative to the static solution. Current-day costs of a 10GE LR transponder is approximately \$40K, which means that with 1000 SCU, the savings are in the millions of dollars. Fig. 2.7 also illustrates that lowering N from 4 to 2 does not yield significant cost savings. The slight dip in the plots when K is 14 occurs because the selected provider router slot capacity was 140 Gbps, which means that for K larger than 14, a second card is required to accommodate the dedicated router ports for the general-purpose IP traffic.

[1]The per-customer network power savings does not vary a lot for different values of K, and the standard deviations are less than 0.1 in both cases.

The dynamic solution requires two wavelengths across the access link while the static solution requires only one wavelength. However, this requirement will not necessarily increase the cost to customers. This is because in the static solution, an enterprise needs to lease a high-speed (e.g., 100GE) static circuit, and the provider correspondingly needs to charge a high price for this circuit because it requires a dedicated transponder and router port. In contrast, in the dynamic solution, the provider could lower the cost for the high-speed access-link wavelength because this wavelength is not connected, i.e., it is left "hanging" until the customer network sends a signaling message to connect it to one of N shared transponders and router ports as shown in Fig. 5.1. The savings in cost to the provider through the use of shared transponders and router ports can be passed along to the customer, which could then be used by the customer to cover the price of the lower-speed wavelength for general-purpose IP traffic. Furthermore, some large enterprises already have dark-fiber leases, which allows the enterprises to light-up wavelengths on their own as needed. In this case, the second wavelength required in the dynamic solution will not result in additional expenses for the customer beyond those characterized in the cost analysis above.

In summary, for our assumed values, we can state that having 2 shared provider-router ports when traffic load is 0.01, and 4 when traffic load is 0.1, yield considerable cost and power savings while keeping the probability of start-time delay exceeding 20 minutes below 1%.

2.5 Conclusions

Significant power and cost savings are possible with a reconfigurable two-wavelength large-enterprise access network design if high rates are required only infrequently, e.g., for large dataset transfers. A lower-rate static optical circuit can be used on one wavelength for general-purpose IP traffic to/from the enterprise, and a higher-rate circuit can be established and released dynamically whenever needed by the enterprise for a large dataset transfer. This design allows for the sharing of a few high-rate provider-router ports that are powered on-and-off dynamically, and lower-rate IP-router ports for general-purpose traffic. Our evaluation shows that just two shared high-rate provider-router ports are sufficient at low

loads to support 20 customers, and at higher loads, 4 shared high-rate provider-router ports are sufficient. At these values, power savings are more than 40 kW and equipment cost savings are in millions of dollars. The penalty paid by our dynamic design is the start-time delay incurred by a customer having to wait if no provider-router port is available. The probability of this delay exceeding 20 minutes is kept below 1%.

Chapter 3

Comparison of Two Sharing Modes for Dynamic Enterprise Access Links

3.1 Introduction

In Chapter 2, we proposed a solution in which dynamic requests to connect a second access link to one the shared higher-speed provider IP-router ports were handled in IR mode. If the number of shared ports is small, blocking probability can be high if the number of enterprises sharing these ports is engineered to keep costs low. If the request for a high-speed circuit is blocked, the enterprise data transfer application falls back to using the lower-speed always-on access link, which results in increased transfer time.

To address this problem, in this chapter, we consider an alternative solution for sharing high-speed provider ports, i.e., AR mode. This AR-mode of operation allows for lowering blocking probability by holding reservations in a time window, thus allowing more large dataset transfers to enjoy the higher-speed access link, without decreasing the number of enterprises sharing the router ports. However, there is a cost to this AR mode. It requires the provider to deploy in-network storage. Several portions of this chapter is selected from our published work *Comparison of two sharing modes for a proposed optical enterprise-access*

SDN architecture [15]

This chapter presents two comparative evaluations of (i) the previous solution using strictly IR mode, and (ii) the new solution using AR mode and provider storage servers. First, we carried out a simulation study to compare *performance metrics*: blocking probability and response time. Next, the same simulation setup was used to determine the *amount of storage space* required to support the AR-mode solution. This latter study was required for a second comparative evaluation: *cost and power consumption analysis*.

Our key findings are as follows: (i) Both performance metrics, blocking probability and average response times, are lower with AR mode when compared with the IR mode. (ii) Storage is required in the provider network to support AR mode. (iii) To achieve a given blocking probability, we can either use more storage or more 100GE ports connecting the storage servers to the IP-routed network in the AR mode. (iv) The AR-with-storage mode solution costs more and consumes more power than the IR-mode solution. Further, SSD-based storage servers are more cost- and power-efficient than HDD-based storage servers.

Section 3.2 provides further details on the alternative AR-mode with network storage solution. Section 3.3 presents a simulation study to compare the performance of the IR-mode and AR-mode solutions. Since the AR-mode requires storage, Section 3.4 quantifies the amount of storage required. Section 3.5 presents a differential cost and power comparison of the IR-mode and AR-mode solutions. Section 3.6 reviews related work, and the conclusions drawn from this work are presented in Section 3.7.

3.2 Alternative AR-Mode with Storage Solution

First, we provide a brief review of the previous IR-mode solution presented in Chapter 2. This solution is illustrated in Fig. 3.1. Each of the K customers (large enterprises) have: (i) a static lower-speed dedicated link (e.g., 10GE) to a provider IP-router port for general-purpose IP traffic (black solid line), and (ii) a second higher-speed lightpath (e.g., 100GE) that terminates at the provider optical platform (red solid line) and remains unconnected ("hanging") until instructed by the optical SDN controller. To support the K

static 10GE links from the customer networks, the provider router, shown in Fig. 3.1 has K dedicated 10GE ports (shown in black). This router also has N_{IR} shared 100GE ports (shown with solid red lines).

When a customer requires a 100GE optical circuit for a high-speed large dataset transfer, its second 100GE lightpath is connected, dynamically by an SDN controller, through the provider optical platform (see Fig. 3.1) to one of the N_{IR} shared 100GE ports on the provider IP router. The SDN controller only supports the IR mode of operation, which means circuit requests are granted if one of shared provider IP-router ports is free, and blocked otherwise. When a high-speed circuit request is blocked, the application running on the enterprise's datacenter will switch to using the default IP path through the internal-subnetworks, customer border IP router and the static 10GE access link. As noted in Section 3.1, if K and N_{IR} are chosen to keep costs low, blocking could be high, which means a significant fraction of large dataset transfers could be routed to the lower-speed access links and thus experience increased transfer times.

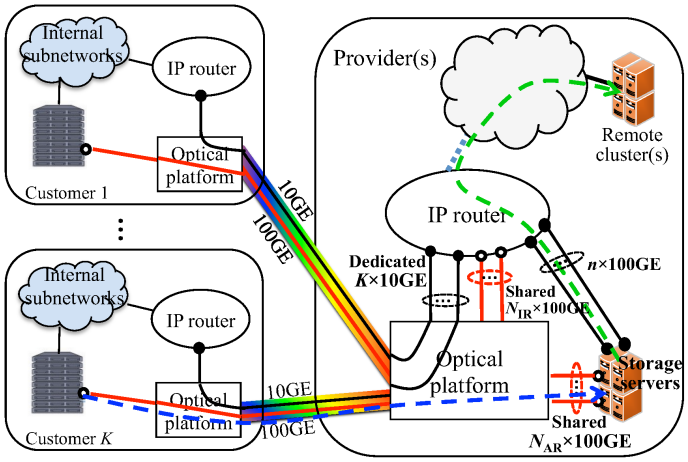


Figure 3.1: Dynamic high-speed port sharing solution

To address this problem, we developed a new solution, which has two features: (i) AR mode for sharing high-speed provider equipment ports, and (ii) network storage. The storage server cluster, as shown in Fig. 3.1, is required if the sharing mode is AR. This is because the AR mode requires a priori knowledge of circuit duration. But if the dynamic optical circuit is created to connect a customer's second lightpath to one of the shared 100GE ports

on the provider IP router (see Fig. 3.1) instead of connecting it to a shared port on a storage server, then the end-to-end path has an IP-segment (from the provider IP router to the remote cluster) on which available capacity is unpredictable, and therefore circuit duration is unknown. The storage server cluster offers a means to disassociate the customer's dynamic optical circuit from the IP-segment to the remote server.

Fig. 3.1 shows that the provider storage servers have shared N_{AR} 100GE ports. These ports are connected dynamically to customer datacenters via the hanging 100GE lightpaths from the enterprise networks. The storage servers are also connected to the provider IP router via n 100GE ports. An example optical circuit is shown with a *dashed blue line* indicating that the second lightpath from customer K has been connected dynamically by the SDN controller to one of the shared 100GE ports on the provider storage servers.

The end-to-end path consists of two segments as shown in Fig. 3.1: (i) a *circuit segment*, which is the dynamically established optical circuit from the customer's data-transfer cluster switch to a storage server at the provider PoP (blue line), and (ii) an *IP segment*, which is the IP-routed path from the provider storage server to the remote end of the data transfer (green line). Since a two-phase transfer of a large dataset will double the delay, we propose to use "blocks" for pipelining. For example, a 1TB file would be divided into 1000 1-GB blocks, and as soon as the provider storage server receives the first 1-GB block from the customer, the provider storage server will initiate the transfer of the block to the remote end over the IP-routed path. Since the data rate available to the flow on the IP-routed path to the remote end will vary, the storage server will need temporary storage to buffer data for the flow.

There is another design difference between the IR-mode and AR-mode solutions. In the IR-mode solution, if a customer datacenter is connected via an optical circuit to a high-speed provider-router port to serve a data transfer, and a second data transfer request arrives from the same customer network, then the second transfer is allowed to use the existing optical circuit. Thus, the duration of an optical circuit is unknown when the circuit is first requested. An optical circuit is released only when all transfers using the circuit complete.

On the other hand, in the AR-mode solution, since circuit duration is required in the setup phase, a second data transfer request from a customer network that is already connected to

a storage server will not be allowed to use the existing circuit. Instead, the request will be scheduled for a future start time if resources are available within the AR window. With this design choice, no packet drops will occur at the top-of-rack switch in the customer's datacenter cluster. Retransmissions will only be required for handling bit errors, if any. Therefore, this design choice allows the requesting application to determine duration based on file size and circuit rate a priori, i.e., when submitting its request to the SDN controller.

Finally, the use of a finite AR window implies that even AR calls could be blocked, and such blocked calls will fall back to the default lower-speed access link.

3.3 Performance Comparison of the IR- and AR-Mode Solutions

We use a simulation study to compare our prior-work IR mode solution with an alternative AR-mode-plus-storage solution on two performance metrics: blocking probability and response time. Section 3.3.1 describes our simulation setup. Section 3.3.2 presents the blocking probability results, and Section 3.3.3 presents response time performance.

3.3.1 Simulation setup

We implemented an event-driven simulator using Python to generate high-speed circuit requests of varying dataset sizes in IR- and AR-modes from K customer networks. Simulation parameters are summarized in Table 3.1. The arrival process for circuit requests is assumed to be Poisson, and the dataset size distribution is assumed to follow a truncated Pareto distribution with a mean of 3 TB. The dynamically shared high-speed ports are 100GE, while the static lower-speed access links are 10 GE. The IP-segment capacity is assumed to be $n\times$100GE.

The main computation was that of file transfer time. Details of the methods used for computing transfer time in the IR- and AR-modes are presented below.

Each simulation run was executed until 1 million circuit requests were generated. Call blocking probability and response times were determined from these runs. These runs were long enough to make the standard deviations a small fraction (a maximum of 0.36%) of the

average response times. All simulation runs were executed on a University of Wisconsin-Madison cluster called CHTC.

Table 3.1: Simulation Parameters

Parameters	**Symbol**	**Values**
Number of shared high-speed ports N	N_{IR}, N_{AR}	2, 4
Number of customers	K	10, 15
Number of 100GE links between provider storage servers and IP router, and number of 100 Gbps capacity between IP router and remote cluster	n	2, 4
Per-customer load (No. of reqs. per day)	λ	$1, \cdots, 30$
File size	F	Truncated Pareto
Shape (Truncated Pareto)	β	0.3
Min file size (Truncated Pareto, GB)	L	10
Max file size (Truncated Pareto, TB)	H	100
Circuit reconfiguration delay (ms)	d	10
Circuit capacity (Gbps)	R_c	100
Static lower-speed access-link rate (Gbps)	R_l	10
Background traffic percentage on IP paths	-	40%
Packet-loss factor	α	$0.75, 1$
AR reservation window size (s)	W	$1, \cdots$,6K
File open-and-close overhead (ms)	τ	1
Size of each file block	b	$\sqrt{R_c F \tau}$

IR-mode transfer time computation Since an end-to-end TCP connection is used in IR mode, the rate of a flow carried on a dynamic circuit is computed as $\min\{r_c(t), r_{IP}(t)\}$, where $r_c(t)$ and $r_{IP}(t)$ are the achievable rates on the circuit and the IP segments, respectively. The value of $r_c(t)$ is equal to 100 Gbps divided by the the number of flows on the circuit (varies with time). We assume that 40% of the $n \times$ 100GE IP-path capacity is occupied by background traffic[1]. Therefore the total capacity available for large data transfers is $60n\alpha$ Gbps, where α ($0<\alpha\leq 1$) is the packet-delivery factor (1−packet loss rate). The flow rate on the IP segment, $r_{IP}(t)$ is computed as $60n\alpha$ Gbps divided by the number of concurrent data transfers. Every time when a new transfer is added, or an old transfer is completed, all the flow rates are adjusted using the methods discussed above. The response time of a transfer is computed as $t_c - t_a$, where t_c is the transfer completion time, and t_a is the request arrival time.

Blocked requests are sent to the default 10GE access links. These links are also assumed to have a 40% background traffic load, and hence only 6 Gbps is available for the large data

[1]A Google 2013 paper [37] stated that " WAN links are typically provisioned to 30-40% average utilization".

transfers that were denied an optical circuit to a high-speed provider port. Per-flow rate and response times are computed using the same method as described above for the large transfers sharing the $n \times 100$GE IP segment.

AR-mode transfer time computation We assume an advance reservation window of length W seconds, and that the size of storage is unlimited. The amount of storage required is estimated in a simulation study, which is presented in Section 3.4. An SDN controller checks for high-speed port availability within the reservation window, and if the circuit request can be accommodated within that window, the SDN controller replies to the circuit requester with a future start time. Requests that cannot be met are blocked, and the corresponding large data transfers will occur over the default 10GE access link.

For a data transfer of size F, with blocks of size b for pipelining, the total transfer time in AR mode is $\frac{b}{R_c} + \lceil \frac{F}{b} \rceil \cdot \tau + t_{e2e,TCP}$, where R_c is the circuit rate (100 Gbps). The first term corresponds to the store-and-forward delay of one file block. The second term is the file open-and-close overhead, which is the product of the number of file blocks and τ, the time required for one set of file open-and-close operations (assumed to be 1 ms). The simulation used the optimal block size that minimized the delay from the first two terms; this optimal block size is $\sqrt{R_c \cdot F \cdot \tau}$. The third delay term is computed using the method described for the IR mode.

3.3.2 Performance metric: Blocking probability

Fig. 3.2 shows how much AR could improve the blocking-probability performance over IR. We define the improvement as $(P_b^{IR} - P_b^{AR})/P_b^{IR}$, where P_b^{IR} and P_b^{AR} are the blocking probabilities in AR and IR modes, respectively. The blocking probability is always smaller with AR than with IR, and hence the improvement values shown in Fig. 3.2 are always positive.

Consider a *baseline case (black line)* in which the number of sources K is 10, the number of shared high-speed ports N is 2, the reservation window size W is 2000 s, and the IP-path packet-delivery factor α is 1, which means that there is no packet loss, and the entire $60n$ Gbps is available for equal sharing among all concurrent flows. At a per-customer load of

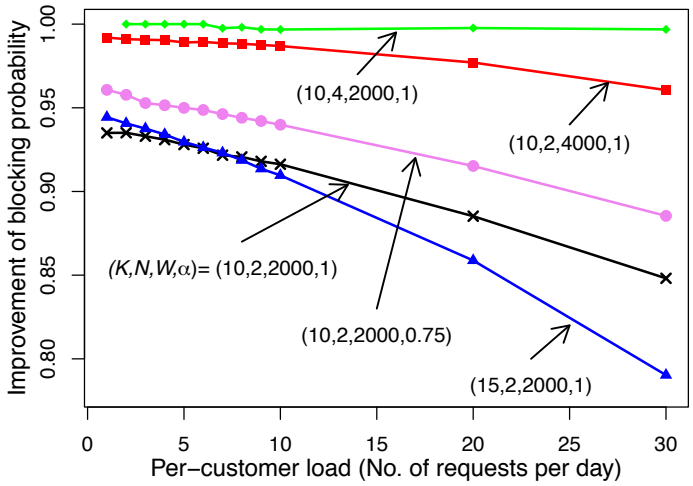


Figure 3.2: Improvement in blocking probability with AR over IR; $n = 2$

20 high-speed circuit requests per day, the AR-mode CBP is 0.015 (1.5%), while it is 0.13 (13%) in the IR mode.

To study the effects of different parameters on blocking probability, we varied the value of only one parameter at a time, and generated separate plots. With a larger reservation window W of 4000 s *(red line)*, the improvement is higher than in the baseline case since more circuit requests can be admitted. In both cases, the improvement decreases with increased traffic load, but the red line decreases at a slower rate. This implies that a larger reservation window is needed for a higher load.

The *blue line* corresponds to a larger number of sources, i.e., $K = 15$, where the improvement drops quickly when the per-customer load increases. However, adjusting N correspondingly would allow AR to maintain its advantage over IR. The *green line* illustrates this point. When we increased the number of shared high-speed ports N to 4, the AR blocking probability was near-zero, which corresponds to an improvement value of almost 1.

Finally, with a higher IP-path loss rate α value of 0.75 *(magenta line)*, the improvement is higher when compared to the baseline. This is because a higher loss rate implies lower throughput on the IP path, which then requires that the circuit be held longer in IR mode. The longer the occupancy of a shared high-speed router port by one customer, the higher the probability that other requests will be blocked. In contrast, in AR mode, because the circuit is terminated at the provider's storage servers, the circuit duration of a transfer is unaffected

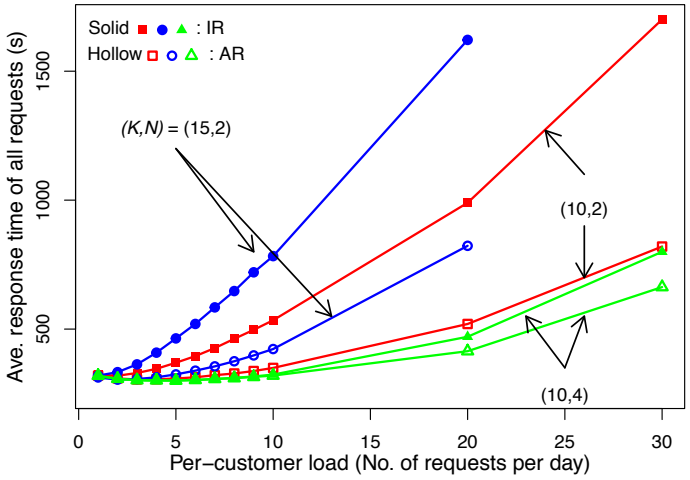


Figure 3.3: Response time comparison; $W = 2000$ s, $n = 2$, $\alpha = 1$

by the packet loss rate on the IP segment as long as storage space is not a bottleneck.

3.3.3 Performance metric: Response time

Fig. 3.3 shows that for all three sets of the (K, N) parameter values (where N is used to indicate both N_{IR} and N_{AR}), the average response time across all requests is smaller with AR mode than with IR mode. Besides, AR mode outperforms IR mode more when the per-customer load increases.

Table 3.2 compares the response time in IR mode with the response time in AR mode for different reservation window sizes. Even with a very small reservation window of 1 s, the percentage of requests handled over circuits in AR mode is 91.6% vs. 87.3% in IR mode (first column). This improvement occurs because the minimum file size is 10GB, and with truncated Pareto distribution, a large proportion of the files will be small. The transmission time of a file of size 10GB is 0.8 s on a 100GE circuit, which is less than 1 s. Therefore, AR mode drops the blocking probability by 4.3%.

The second column of Table 3.2 shows the average response time of requests that were handled over circuits in the IR mode and the average response time of AR-mode requests that did not have to wait for a circuit. We observe that the average response time in AR mode is always higher than in IR mode. This is because in AR mode previous requests could have been admitted to time slots in which no-wait-period AR requests were simultaneously

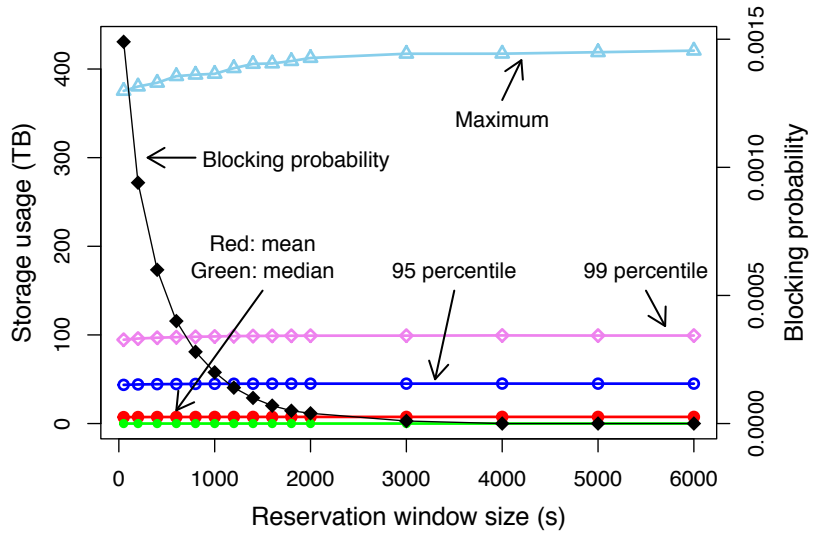
Table 3.2: Average response time for different reservation window sizes when $K = 10$, $N = 2$, $n = 2$, $\lambda = 20$, $\alpha = 1$

		Percentage of requests over circuits	Average response time of requests over circuits without a wait period (s)	Average response time of requests over circuits waiting in AR but blocked in IR (s)	Average response time of all requests (s)
IR		87.3%	332.9	5511.5	989.9
AR reservation window size W (s)	1	91.6%	333.1	418.3	673.6
	1000	96.9%	349.3	875.7	511.2
	1500	97.9%	353.8	1039.1	496.4
	2000	98.6%	357.5	1179.2	492.0
	2500	99.0%	359.7	1300.2	493.9
	3000	99.33%	360.8	1396.8	495.7
	4000	99.71%	363.8	1553.5	506.8
	5000	99.89%	365.4	1645.4	515.3
	6000	99.96%	365.5	1687.7	516.8

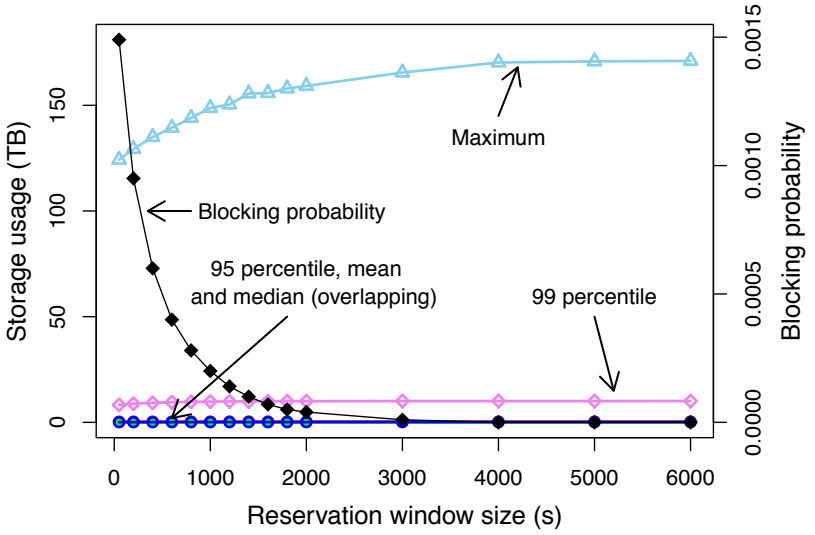
granted. The probability of there being only 1 large data transfer request at a time is higher in IR mode than in AR mode because the latter essentially has a time buffer to hold previous requests. In AR mode, the probability of 2 calls sharing the IP segment (of $n \times 100$ Gbps, which is 200 Gbps since $n = 2$ in the results of Table 3.2) is higher. Therefore, the transmission times for AR-mode requests without a wait period are higher than the transmission times of IR-mode calls, which likely enjoy 100 Gbps rates. Furthermore, the probability of a new AR-mode request being granted service in a set of time slots when there are other simultaneous transfers is higher the longer the reservation window size. Therefore, the average response time in AR mode increases with W.

The third column of Table 3.2 shows the average response time of IR-mode blocked requests vs. AR-mode queued requests. In IR mode, the blocked requests were sent to the 10-GE limited default IP-routed path, while in AR mode, these requests found a waiting spot in the reservation window, were essentially queued and served later. The average response time for such calls in AR mode is significantly smaller than in IR mode. Although these requests experienced non-zero start-time delay in AR mode, they enjoyed much higher circuit capacities when compared to the lower 6-Gbps paths available in IR mode.

Since blocked transfers are executed on lower-rate (10 Gbps) paths, their overall transfer times are longer, and hence when the averaging of response time is done across all calls, the AR mode outperforms the IR mode as seen in Fig. 3.3 and the last column of Table 3.2.



(a) Storage usage when $n = 2$



(b) Storage usage when $n = 4$; green and red lines overlap with the blue line

Figure 3.4: Storage usage and blocking probability in AR mode when $K = 10$, $N_{AR} = 4$, $\lambda = 20$, $\alpha = 1$

3.4 Storage Use Estimation

As seen in the previous section, the AR mode achieves better performance in terms of both blocking probability and response time when compared to the IR mode; however, it requires the use of network storage. In this section, we present simulation results that quantify the amount of storage required for AR-mode operation. There is an interesting tradeoff between

the amount of storage required with the number of 100GE ports used to connect the provider storage servers to the IP-routed network (see Fig. 3.1).

Fig. 3.4 shows how the amount of storage used and blocking probability change with the size of the reservation window. Since the amount of storage used varies as requests arrive and depart, we executed 100 simulation runs. Fig. 3.4 shows the mean, median, 95 and 99 percentiles, and maximum values of used storage across the 100 simulation runs.

Our key observations from Fig. 3.4 are as follows: (i) The maximum storage used increases with the reservation window size when W is small, but saturates. This is because at a given load level (e.g., $\lambda = 20$ per-customer requests/day), there is a reservation window size at which blocking probability drops close to 0. Increasing the reservation window size beyond that level, without changing load, does not increase the number of circuit requests admitted, and hence the storage used does not increase. (ii) The blocking-probability values in Fig. 3.4a when $n = 2$ are the same as those in Fig. 3.4b when $n = 4$. This is because blocking is incurred on the optical circuit side (under the assumption of infinite storage capacity), not on the link between the storage servers and the IP router, which is determined by n. The maximum storage usage when n is 2 is larger than when n is 4, because it takes longer to move stored datasets over the lower-capacity links to the IP router. An interesting observation here is that we can either increase storage or increase the number of 100GE ports used to connect the storage servers to the IP-routed network to achieve the same blocking probability (since with finite storage, availability of space in the storage servers will need to be included when deciding whether to admit or block a circuit request). (iii) The maximum storage usage observed across the 100 runs (at $W = 2000$) was larger when $n = 2$ as expected (420 TB) vs. 170 TB when $n = 4$. Furthermore, the IQR of storage space used was greater, 3805 GB when $n = 2$ vs. 1.31 GB when $n = 4$.

3.5 Cost and Power Consumption Analysis

This subsection presents a differential cost and power comparison of the IR and AR-with-storage modes. The network devices in the customer networks, the optical platform in the

Table 3.3: Components with differing quantities in IR and AR modes

Components	Quantity		Unit price (USD)
	IR	AR	
100G provider IP-router port	N_{IR}	n	2K [38]
100G SR transceiver	N_{IR}	n	0.2K [39]
100G storage-server NIC	0	$N_{AR} + n$	see Table 3.4
Storage server with drives	0	see Table 3.4	see Table 3.4

provider network, and the static lower-speed dedicated ports in the provider IP router are used in both solutions, and are therefore omitted from the comparison.

The components that are different in the AR and IR modes are listed in Table 3.3. The number of 100GE ports and transceivers in the provider IP router is N_{IR} in the IR mode, and n in the AR mode. We assumed that all transceivers are of Short-Reach (SR) type, since the provider IP router and optical platform in both modes, and storage servers in AR mode, are located at the same PoP (data center). The configuration of storage servers needed in the AR mode depends upon the specific type of storage drives. Two options: HDD and SSD, are considered in our study (see Table 3.4).

To achieve 100 Gbps disk I/O rates, multiple storage drives need to work in parallel because the read/write rates of each drive are on the order of a few hundred Mbps to a few Gbps depending on the technology e.g., HDD or SSD. Given the rate difference in HDD and SSD drives, we can either use a larger number of slower but less expensive HDD drives, or a smaller number of faster but more expensive SSD drives to achieve 100 Gbps disk I/O. We label these two design choices as HDD- and SSD-based architectures, respectively.

For this cost-power comparison, we choose the following configuration: $N_{IR} = N_{AR} = 4$, and $W = 2000$. We chose two values for n: 2 and 4, in keeping with our storage usage analysis in Section 3.4. As noted at the end of Section 3.4, the maximum storage usage observed across the 100 simulation runs was 420 TB when $n = 2$ and 170 TB when $n = 4$. We use these storage sizes in our cost and power analysis below.

HDD-based architecture Our HDD storage-server architecture is based on the 100G storage system proposed by Suerink [51]. In Suerink's design, 112 8TB NL-SAS HDD drives are hosted in two AssuredSAN 4004 drive platforms, which are in turn connected to an IBM Power system that supports 100GE NICs. This configuration can only fill up (fully utilize)

Table 3.4: Prices, power consumption and quantities of storage-server components in the example HDD- and SSD-based architectures

Architectures	Components	Quantity		Unit price (USD)	Unit power(W)
		$n = 2$	$n = 4$		
HDD-based	IBM Power system S822L	2	2	15K [40]	1810 [41]
	Mellanox dual-port QSFP28 NICs	3	4	1.2K [42]	-†
	Broadcom 9405W storage adapters	12	16	0.5K [43]	-†
	PAC Storage GS 3000 60 Bays	12	16	20K [44]	1200 [44]
	PAC Storage host board	24	32	0.5K [44]	0 [45]
	Dell NL-SAS 12 Gbps HDD, 1TB	672	896	0.19K [46]	-‡
SSD-based	Supermicro 2028R-NR48N	4	2	6.1K [47]	1600 [47]
	Supermicro 2028U-TN24R4T+	1	0	3.6K [48]	1600 [48]
	Supermicro dual-port QSFP28 NICs	3	4	1.2K [49]	-§
	Samsung 960, 2TB NVMe SSD	210	85	1K [50]	-§

† The power consumption of the NICs and storage adapters is non zero, but their power supply is provided by the S822L. Thus only the power consumption of the S822L is included when calculating the total power consumption.
‡ The power supply of the HDD drives is provided by the PAC storage systems. Thus only the power consumption of the PAC storage systems is included when calculating the total power consumption.
§ The power supply of the NICs and SSD drives is provided by the Supermicro servers. Thus only the power consumption of the servers is included when calculating the total power consumption.

a single 100 Gbps link.

To support $(N_{AR} + n)$ 100 Gbps links on the storage servers, we need $(N_{AR} + n) \times 2$ drive platforms and $(N_{AR} + n) \times 112$ HDD drives. We found that a PAC storage GS 3000 system is similar to the AssuredSAN 4004 platform but cheaper, and hence selected the PAC storage system. We replaced the 8TB drives in Suerink's design with 1TB drives because even when $n = 2$, the maximum storage required is 420 TB as noted above, but the system require 112 drives to achieve the 100 Gbps disk read/write throughput, and 1TB drives are cheaper than 8TB drives. Therefore, for the $n = 2$ configuration, with $N_{AR} = 4$, the system requires 12 PAC storage GS 3000 systems and 672 Dell 1TB HDD drives in the HDD-based architecture (see Table 3.4). Similary, when $n = 4$, we require $(N_{AR} + n) * 2$ drive platforms, i.e., 16, and $(N_{AR} + n) * 112$ drives, i.e. 896 TB. Unfortunately, the smallest sized drive supporting 12 Gbps read/write rates is 1 TB. Therefore, even though the solution requires only 420 TB (when $n = 2$) and 170 TB (when $n = 4$), our designed system will have 672 and 896 TB, respectively.

Since each Power system supports 8 PCIe storage adapters [41], we need 2 Power systems to connect 12 or 16 PAC storage systems. In the $n = 4$ configuration, each Power system

is fully loaded with 8 Broadcom storage adapters (see Table 3.4), while in the $n = 2$ configuration, each Power system is loaded with only 6 storage adapters. Each PAC storage system is connected through two storage host boards to one of the PCIe storage adapters on the IBM Power system. Since we require 12 and 16 storage adapters on the Power systems for $n = 2$ and $n = 4$ configurations, respectively, the system requires 24 and 32 PAC storage host boards as listed in Table 3.4.

Finally, when the number of 100GE NICs on the storage servers, $N_{AR}+n$ (see Table 3.3), is 6 and 8, for the $n = 2$ and $n = 4$ configurations, respectively, the number of Mellanox dual-port QSFP28 NICs required is 3 and 4, respectively (see Table 3.4).

SSD-based architecture We follow the guidelines from ESnet [52] to build 100 Gbps storage servers based on SSD. To achieve 100 Gbps disk I/O, 10 NVMe SSD drives are recommended. On the other hand, given that the maximum size per NVMe SSD drive available on the market is 2-TB, we need 210 and 85 2-TB drives to build storage sized at 420 TB ($n = 2$) and 170 TB ($n = 4$), respectively. To hold these drives, we need multiple Supermicro servers, each of which has 48 (2028R-NR48N) or 24 (2028U-TN24R4T+) NVMe drive bays (see Table 3.4).

All 85 2-TB drives can be accommodated in the two 48-bay 2028R-NR48N servers for the $n = 4$ configuration. However, for the $n = 2$ configuration, we propose using four 48-bay 2028R-NR48N servers to support 192 drives, and one 24-bay 2028U-TN24R4T+ server to accommodate the remaining (210-192=18) drives.

Finally, 3 and 4 dual-port QSFP28 NICs are included to support the six and eight 100GE NIC requirement for the $n = 2$ and $n = 4$ configurations, respectively. This requirement is the same in both HDD- and SSD-based architectures.

Total cost and power comparison for HDD- and SSD- based architectures in AR mode Table 3.5 compares the actual storage size, total cost and power consumption of storage servers under the two target configurations in the HDD- and SSD-based architectures. The results are computed directly from the component quantities, per-unit prices, and per-unit power consumption levels listed in Table 3.4.

Table 3.5: Total cost and power consumption of storage servers required in the example HDD- and SDD-based architectures

Storage-server configurations		**Solution architectures**	**Actual storage sizes**	**Total cost (USD)**	**Total power consumption (kW)**
Target storage sizes	**No. of 100GE links between provider storage servers and IP router**				
420 TB	$n = 2$	HDD	672 TB	419.3K	18.0
		SSD	420 TB	241.6	8.0
170 TB	$n = 4$	HDD	896 TB	549.0K	22.8
		SSD	170 TB	102K	3.2

Our findings are as follows: (i) In both target configurations, the actual storage sizes provided by the HDD-based architecture are larger than than the target storage sizes, i.e., 672 TB vs. 420 TB, and 896 TB vs. 170 TB. This is because the HDD-based architecture requires a large number of HDD drives (i.e., 112) to support 100 Gbps disk I/O rates, and the minimum HDD drive size supporting 12 Gbps read/write rates is 1TB. In contrast, the actual storage sizes are the same as the target storage sizes in the SSD-based architecture. (ii) For both target configurations, the SSD-based architecture has lower cost and power consumption than the HDD-based architecture. (iii) The SSD-based architecture with $n = 4$ has the least total cost, i.e., \$102K, and the least total power consumption, i.e., 3.2 kW, among the four choices.

Cost and Power Comparison of IR and AR modes We use the AR-mode configuration with the most cost- and power-efficient SSD-based design for comparison with IR mode. This design uses four 100GE links between the provider storage servers and IP router, i.e., $n = 4$. When the number of dynamically shared 100GE provider router ports, N_{IR}, is 4 in the IR mode, and n is also 4 in the AR mode, the cost and power consumption of the 100GE ports on the provider IP router are the same (see Table 3.3). Differences in cost and power consumption under the two modes are only due to the cost and power consumption of storage servers in the AR mode (see Tables 3.3 and 3.5). Therefore, the AR-with-storage solution costs \$102K more and consumes 3.2 kW more power than the IR solution. Since the total cost and power savings achieved by the dynamic access link solution in IR mode are in millions of dollars and more than 20 kW, respectively (when $K = 10$, $N_{IR} = 4$), when

compared to the conventional static solution using K 100GE provider router ports [6], the dynamic access link solution in the AR-with-storage mode can still achieve significant cost and power savings.

In return for this extra cost and higher power-consumption, the AR-mode brings performance improvement over the IR mode in terms of blocking probability and response time (see Section 3.3.2 and Section 3.3.3).

3.6 Related Work

Previous papers on leveraging optical SDN for bulk data transfers offer the following contributions. Lu *et al.* proposed to facilitate efficient bulk-data transfers in EONs with malleable reservations, which performs adjustable routing and spectrum assignment [53]. Jin et al. presented Owan, a traffic management system that optimizes bulk transfers over wide area networks by dynamically reconfiguring optical devices to change the network-layer topology [54]. Samadi et al. proposed a software-defined inter-datacenter network architecture to enable on-demand scale out of data centers on a metro-scale optical network [55]. The architecture consists of a combined space/wavelength switching platform and an SDN control plane, which enables end-to-end bulk data transfer and VM migrations across data centers with less than 100 ms connection setup time and close to full link-capacity utilization.

Storage has been introduced to facilitate bulk data transfers. Patel et al. proposed time-shift circuit switching to shift the data transfer on a link to times when the bandwidth is available by utilizing storage [56]. Laoutaris et al. developed analytical models for transferring bulk data through single-hop and single-path transfers [57]. and showed the huge potential of storage for transferring multi-terabyte data on a daily basis at no additional cost They further proposed NetStitcher [58], a system that employs storage to stitch together unutilized bandwidth for bulk data transfer. Wu et al. built a bulk data transfer system employing SnF based on the Beacon platform and OpenFlow APIs with practical online algorithms to optimize routing [59]. Prior work proposed deploying storage in OCS networks to facilitate delay-tolerant bulk data transfers [60]. To tackle the routing and scheduling

issues, a routing framework, named time-shift multilayer graph, was proposed to perform spatial routing and temporal scheduling within bulk data transfers. The authors further applied slotted network operations to OCS networks with storage [61].

Compared to all this prior work, our contribution is a comparative analysis of AR mode with storage, and IR mode. In addition, the previous work focussed on deploying storage in core networks, while we introduced storage into enterprise access networks.

3.7 Conclusions

This paper compared two options for the design of a proposed optical enterprise-access SDN architecture. In both options, enterprises use a second wavelength that is dynamically connected to a shared high-speed port in the provider network just for demanding applications such as large data transfers. The two options were: (i) AR mode with provider storage, and (ii) IR mode with end-to-end TCP connections. Simulation results show that the AR-mode-plus-storage solution was able to improve data-transfer throughput by decreasing call block- ing probability, thus ensuring that a greater fraction of large transfers use the higher-speed paths. The AR-mode solution requires the use of network storage, which costs $102K more and consumes 3.2 kW more power than the IR-mode solution in an example configuration when $N_{IR} = N_{AR} = n = 4$, and $W = 2000$.

Chapter 4

Optical Switch in the Middle (OSM) Architecture for DCNs with Hadoop Adaptations

4.1 Introduction

In this chapter, we present a novel *Optical Switch in the Middle (OSM)* hybrid architecture for DCNs, and propose four adaptations for Hadoop to operate effectively in the OSM architecture. A significant portion of this chapter is an excerpt from our published work *Optical switch in the middle (OSM) architecture for DCNs with Hadoop adaptations* [16]

Besides enabling the dynamic creation of high-speed ToR-to-ToR circuits, as done previously, OSM offers the ability to connect ToR switches to the core EPS via high-speed optical circuits through the OCS. This feature enables a ToR switch to engage in high-speed communications with multiple ToR switches simultaneously. In addition to this basic feature, the OSM architecture includes: (i) integrated *storage* in the core EPS, and (ii) *AR* scheduling in a *multilayer SDN controller*. Both features were originally designed for wide-area networks. To the best of the authors' knowledge, this is the first time these concepts have been proposed for use in DCNs. These additional features offer applications, users and administrators

higher communications speeds when needed, along with increased flexibility, when compared to previous hybrid DCN architectures.

The second part of this chapter considers the question of how applications can utilize these high-speed optical circuits despite their long provisioning times. As stated in Section 1.2, we focus on *Hadoop MapReduce applications* in this work. We demonstrate the need for, and then propose four modifications to enable the effective operation of Hadoop in the OSM architecture. Our modifications will allow for higher CPU utilization of the compute nodes while simultaneously offering users shorter job completion times.

Our two main *contributions* are: (i) a novel OSM hybrid packet/optical architecture for DCNs, and (ii) Hadoop adapations to enable users and DCN owners to enjoy the benefits of the OSM architecture.

The remainder of this chapter is organized as follows. Section 4.2 briefly surveys related work and provides background on Hadoop. Section 4.3 presents our novel OSM architecture. Section 4.4 describes an experimental study to characterize the communication patterns and computational needs of a Shuffle-Heavy (SH) MapReduce application. Section 4.5 presents our four modifications to enable efficient use of OSM by Hadoop. The chapter is summarized in Section 4.6.

4.2 Related Work and Background

Section 4.2.1 reviews related work on hybrid packet/optical datacenter network architectures, while Section 4.2.2 offers a brief tutorial on Hadoop.

4.2.1 Related work

Hybrid electrical/optical datacenter network architectures using OCS include Helios [5], c-Through [11], OSA [12], Mordia [10], and REACToR [13], among others. Free-space optics-based reconfigurable interconnects, FireFly [62], Diamond [63], Graphite [64] and ProjecToR [65], have also been proposed for DCN. In most of these approaches, traffic is aggregated and/or monitored to determine the pairs of ToR switches that should be interconnected. For example, ProjecToR proposes to derive probabilities that two ToR

switches will communicate from historical traffic matrices. While the advantage of these approaches is that applications do not need to be modified unlike our approach, the disadvantage is that these solutions cannot fully leverage optical circuits.

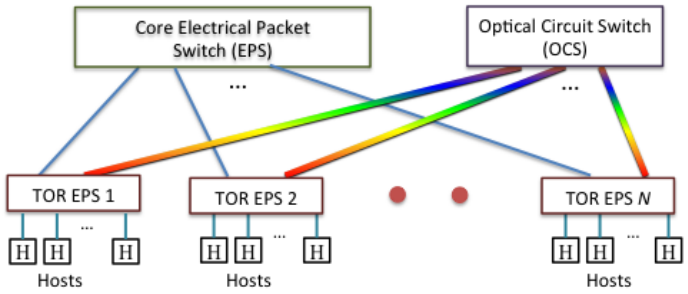
Yamashita et al. [66] proposed a Hadoop triggered hybrid data-center orchestration architecture for reducing power consumption. The architecture identifies shuffle-heavy jobs by estimating shuffle data sizes, and redirects the shuffle traffic on to optical circuits. While this approach uses application-level information to trigger circuit setup, reconfiguring circuits for small shuffle flows may introduce high reconfiguration overhead.

A third track of related work comes from the cloud computing research community. Besides delay scheduling [67], there are other solutions that try to avoid inter-rack transfers. The Shufflewatcher solution [68] proposes to monitor network traffic and to use shuffle-aware map and reduce task placement algorithms in a manner that reduces shuffle traffic. Another network-aware scheduling approach [69] proposes handling large jobs with predictable job characteristics with an offline planned scheduling solution to reduce shuffle traffic. Wang et al. [70] propose scheduling reduce tasks near the nodes where map output is generated so that inter-rack shuffle traffic can be reduced. Hybrid EPS-OCS networks, with dynamic circuit management, offer an alternative solution to this inter-rack shuffling problem.

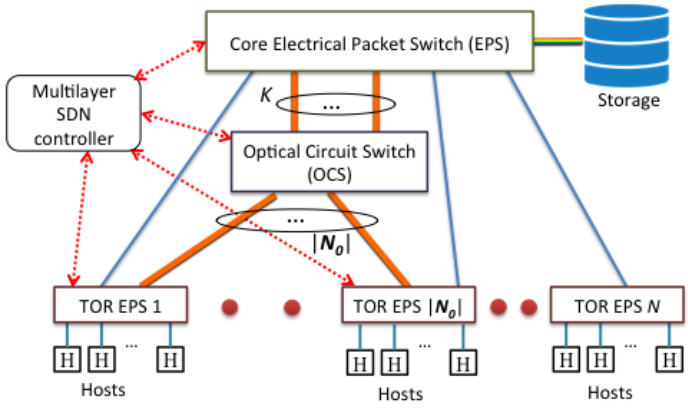
4.2.2 Background

Hadoop is commonly used for storing and analyzing large datasets [71]. Hadoop has three main components: (i) Hadoop Distributed File System (HDFS) for storing and accessing datasets, (ii) Yet Another Resource Negotiator (YARN) for resource scheduling, and (iii) MapReduce for submitting data analysis jobs that consist of a set of parallel map tasks followed by a set of parallel reduce tasks.

When a MapReduce job is submitted to analyze dataset d, its `Application Master (AM)` determines the nodes on which the blocks of dataset d are located, and requests containers for map tasks on these nodes. The Hadoop strategy is to "bring-code-to-data" rather than "data-to-code." Movement of an input dataset is thus avoided. However, there is a phase in MapReduce in which data movement is unavoidable. This phase, called *shuffle*,



(a) Helios-type hybrid network



(b) Optical switch in the middle hybrid network

Figure 4.1: Hybrid electrical/optical DCN architectures

is used to move map output to the nodes on which reduce tasks are executed. A MapReduce job with a large map output is referred as a *shuffle-heavy* job.

In Hadoop, a feature called *reduce slow start* is used to reduce the impact of communication delay during shuffle. Reduce tasks are assigned to nodes and initiated after just a small fraction of map tasks have completed (default: 5%) in order to conduct the shuffle data movement in parallel with the execution of the remaining map tasks. However, the disadvantage is that CPU resources could be wasted by reduce tasks consuming containers while waiting for map tasks to complete.

4.3 Optical Switch in the Middle (OSM) Architecture

Fig. 4.1b shows our proposed Optical Switch in the Middle hybrid network architecture. The optical circuit switch in the middle interconnects the ToR EPSs and the core EPS. Besides the OCS, the OSM architecture requires high-speed transceivers in the ToR switches of at

least some subset $\mathbf{N}_o$ of the racks in the cluster, and a number K of high-speed transceivers in the core EPS, where $K < |\mathbf{N}_o|$. These K ports are shared dynamically by signaling the SDN controller, shown in Fig. 4.1b, to connect a ToR switch to the core EPS when needed. The SDN controller can also handle requests for direct ToR-to-ToR optical circuits as in other hybrid network architectures. ToR-to-core circuits are particularly useful when a ToR switch needs to engage in high-speed communications simultaneously with multiple ToR switches.

In contrast, in the Helios/c-Through hybrid architectures (shown in Fig. 4.1a), to support such communications, either multiple optical circuit setup and release cycles are required (which adds delay), or WDM support is needed. The disadvantage of WDM is the cost incurred for the WDM multiplexers/demultiplexers or wavelength selective switches and the additional high-speed transceivers needed in the TOR EPSs. Other solutions based on tunable lasers require modifications to the ToR EPSs. Given the port counts of today's optical circuit switches (e.g., 1000 ports [5]), the scalability of OSM is better than that of Helios-like architectures with WDM. The OSM architecture does pay the penalty of needing K high-speed transceivers in the core EPS, which is not required in the Helios/c-Through architecture.

Fig. 4.1b shows a storage unit connected to the core EPS. An explanation for this storage unit is provided after we describe the use of advanced-reservation scheduling in the multilayer SDN controller.

Advanced-Reservation (AR) scheduling In wide-area optical and path-based Layer-2 (L2) networking, AR schedulers such as On-Demand Secure Circuits and Advance Reservation System (OSCARS) [72] have been developed and deployed. However, the use of AR scheduling has not been proposed (to our knowledge) for use in hybrid datacenter networks. In the OSM architecture, even though the links to the core EPS from the OCS makes it easier to have multiple simultaneous ToR-to-ToR high-speed communications, support for AR in the multilayer SDN controller would help limit the costs of deploying the additional equipment. This is because AR mode of channel sharing allows for high channel utilization to co-exist with low call blocking probability even when the number of channels shared is small. For

example, in the Immediate-Request (IR) mode of channel sharing, if the number of shared channels is 10, call blocking probability will be as high as 23% even when utilization is only 80%. But with AR-mode channel sharing, 95% link utilization can be achieved with a call-blocking probability of only 1% [73].

The catch however is that to support channel sharing in the AR mode, all requests for circuits should specify durations. Without knowledge of when ongoing calls will depart, the SDN controller cannot assign future start times to new incoming calls. Some applications can meet this requirement, as we will later demonstrate with Hadoop MapReduce jobs. Other applications include VM migrations, checkpointing, and file replication in distributed file systems such as HDFS. In all these applications, as the size of the data to be transferred is known a priori, the duration for which a circuit is required can be estimated accurately from the rate and other parameters.

In addition to handling reservation requests, the SDN controller is also responsible for provisioning the OCS at the start time of reservations, and for configuring flow tables in the ToR EPSs, and possibly the core EPS. Thus, the OSM architecture uses a multilayer SDN controller.

Storage The use of disk storage at IP routers has been popularized recently by Named Data Networking, Information or Content Centric Networking [74]. In recent work [60], we proposed the use of assistive storage in wide-area optical circuit networks, and here, we propose to leverage this concept in the OSM architecture. Storage at network switches is especially useful in datacenter networks since scheduling of circuit resources will often depend on the availability of compute resources. As will be demonstrated later in the Hadoop application, inter-rack shuffling over an optical circuit requires two links, i.e., from the map-rack TOR switch to the OCS and from the OCS to the reduce-rack TOR switch. The presence of storage at the core EPS decouples the map-rack-to-storage data movement from the data movement from storage to the reduce rack. These two actions can be performed at different instants in time based on optical link availability. Hence, the addition of storage at the core EPS can improves channel utilization and reduce communication delays if the network is highly loaded since the scheduler will not need to wait for both optical links to

become available simultaneously.

4.4 Hadoop Application Characterization

A set of experiments were executed to gain insights into whether-or-not Hadoop can be adapted to work effectively in hybrid networks. The specific objectives of our experiments were (i) to quantify the network traffic generated by a shuffle-heavy Hadoop application, (ii) to determine how CPU, memory and network resources affect job completion time, and (iii) to study the impact of *reduce slow-start*, as defined in Section 4.2.2.

Our *findings* are that (i) the network can become the bottleneck when compute resources are not scarce, (ii) as YARN scheduling does not take into account network bandwidth, reduce tasks can sometimes be concentrated on one node leading to increased shuffle communication delay, and (iii) reduce slow start feature can cause both CPU under-utilization and increased job completion time.

4.4.1 Experimental setup: Hardware and software

Two NSF-funded testbeds were used: InstaGENI [75] and Chameleon. In both systems, Hadoop-2.7.1 was installed on a 3-node cluster for these experiments. The first two experiments were run on the InstaGENI testbed, while the third experiment was run on the Chameleon testbed.

All three components of Hadoop described in Section 4.2.2 were installed and executed. The HDFS `namenode`, which manages the filesystem namespace, was run on a master node, and an HDFS `datanode` was run on each of the two worker nodes on which an input dataset of size 10 GB was distributed into eighty 128-MB blocks. This dataset was generated by running a Hadoop benchmark, `TeraGen`. The YARN `Resource Manager (RM)` was run on the master node, and one YARN `Node Manager (NM)` was run on each of the two worker nodes. Finally, we chose a shuffle-heavy Hadoop MapReduce benchmark, `TeraSort` [76], to run on this 3-node cluster.

Each worker node on InstaGENI has 32 physical cores, but using the YARN NM parameters, each worker node was restricted to run with either 4 containers or 16 containers,

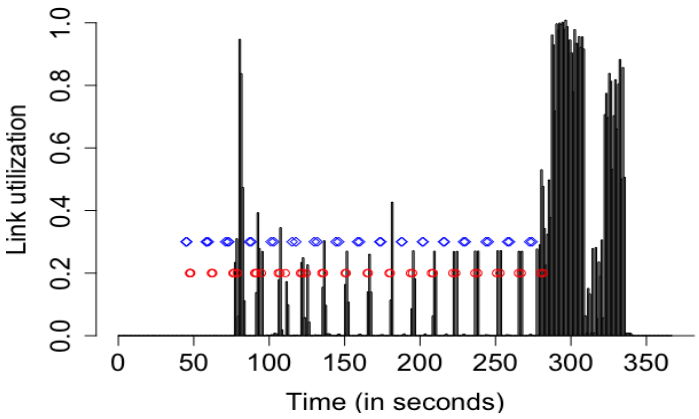


Figure 4.2: Per-sec network traffic sent by worker node 0 normalized to 400 Mbits, $N_c = 8$, $R = 400$ Mbps

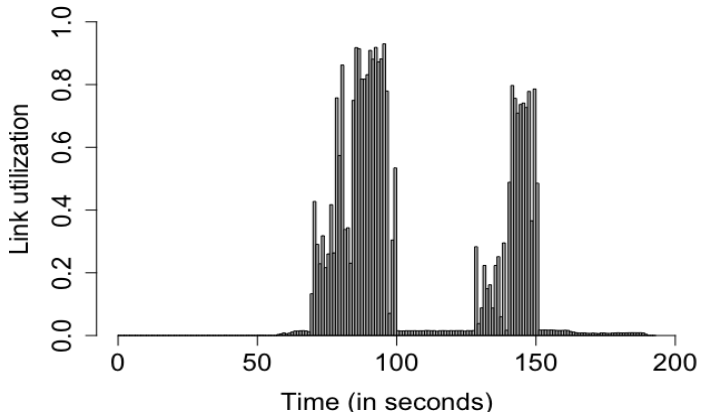
with one container consuming one physical core and 2-GB memory. This YARN feature enabled a study of the impact of compute resources on job completion time. MapReduce parameters allow a user to specify the number of input blocks processed by each map task, which was chosen to be 1. Since there were 80 HDFS blocks, `TeraSort` ran with 80 map tasks. The number of reduce tasks, which is a run-time argument, was set to 10. All other Hadoop parameters were left unchanged at their default values.

Network bandwidth was controlled in our experiments to study its impact on job completion time. The Linux traffic control utility `tc` was used to rate-limit outgoing traffic from each worker node to a set value R.

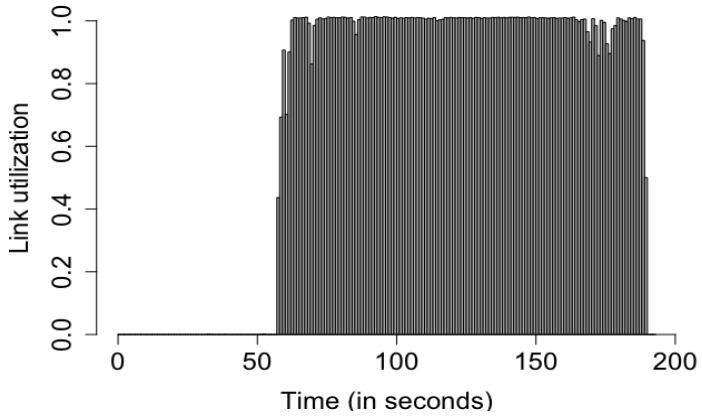
Job completion time was computed by subtracting the job `submitTime` from the job `finishTime` as reported in the Hadoop job history logs, and the Linux `ifconfig` command was invoked to obtain per-sec bytes sent/received on the data-plane NIC of both worker nodes. The `ifconfig` measurements reported were due primarily to shuffle traffic since map input data was not moved (map tasks were assigned to the nodes on which blocks were located), and reduce output was saved locally.

4.4.2 Experimental results

Shuffle traffic patterns Fig. 4.2 shows link utilization of the worker-node 0 NIC as a function of time, where link utilization is defined as the ratio of the per-second network



(a) Per-sec network traffic sent by worker node 0



(b) Per-sec network traffic received by worker node 0

Figure 4.3: Per-sec network traffic sent and received by worker node 0, $N_c = 32$, $R = 400$ Mbps

traffic sent or received by a worker node on its NIC to the `tc` rate R. The blue and red dots in Fig. 4.2 indicate the completion times of map tasks on worker node 0 and worker node 1, respectively. Network traffic bursts occur when map tasks complete. This is explained by the shuffle process in which intermediate map output files are moved to nodes running the reduce tasks. The high link-utilization phase towards the end is due to the fact that 7 out of 10 reduce tasks had not been scheduled until all map tasks completed, and all map output was available. This run was compute-limited as the total number of containers across both worker nodes was only 8, and hence the NIC was largely idle.

In contrast, Fig. 4.3 shows that network bandwidth becomes the limiting factor when the number of containers was increased to 32 across both nodes. The network traffic received

Table 4.1: TeraSort job completion time; R: link rate; N_c: total number of containers

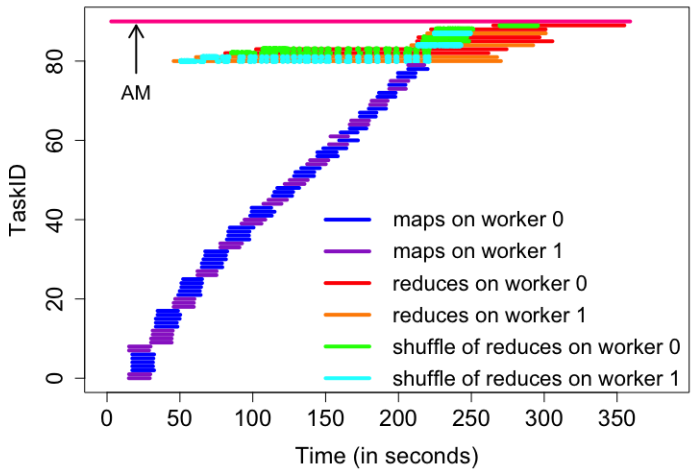
(R, N_c)	Job completion time
(100 Mbps, 8)	470s
(400 Mbps, 8)	336s
(100 Mbps, 32)	346s
(400 Mbps, 32)	160s

by worker-node 0 was sent by worker-node 1 as there were only two worker nodes. A second observation from Fig. 4.3 is the imbalance in link utilization. Just by chance, in this run, 8 of the 10 reduce tasks were assigned to containers on worker-node 0, and hence the map output traffic was higher from worker-node 1 to worker-node 0 than in the opposite direction. Finally, the long period from 58 sec to 190 sec during which the worker-node 0 NIC was fully utilized in the receiving direction suggests that the shuffle traffic for this application can saturate (fully utilize) a 400-Mbps link.

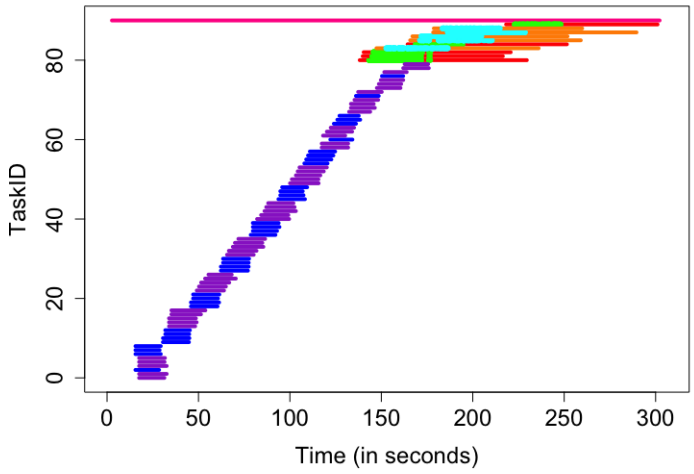
Impact of compute and network resources Table 4.1 compares the completion time under four settings of compute and network resources. Increasing the number of containers (available concurrently to a job), N_c, from 8 to 32, results in a significant reduction of job completion time. Further, the reduction in job completion time achieved by increasing link rate R is more significant in the setting with 32 containers than in the setting with just 8 containers. In other words, when CPU resources are no longer a constraint, network resources become more important.

Impact of reduce slow-start `TeraSort` was run on Chameleon with link rate R set to 1 Gbps, and the number of concurrently allowed containers for the job was limited to 10 (across two worker nodes). The reduce slow-start factor was set to 5% (default) in the first run of the application, and to 80% in the second run.

Fig. 4.4 illustrates the duration of each task in the two runs. `TaskIDs` 0-79 correspond to map tasks, `TaskIDs` 80-89 correspond to reduce tasks, and `TaskID` 90 corresponds to `AM`. As the total number of containers was 10, there were initially 5 map tasks on worker-node 0 (blue lines) and 4 map tasks on worker-node 1 (purple lines), because the fifth container on worker-node 1 was used for the AM, which ran throughout the duration of the job (pink line). The `AM` coordinates all the map and reduce tasks. Recall that there were 80 map tasks,



(a) Reduce slow-start factor is 5%



(b) Reduce slow-start factor is 80%

Figure 4.4: Start and finish time of map tasks, reduce tasks and shuffling, $N_c = 10$, $R = 1$ Gbps

and hence there are many staggered "waves" of map tasks in Fig. 4.4. The red and orange lines represent the reduce tasks, and the green and cyan lines show the shuffle phase (shuffle is executed by the reduce tasks, and hence these lines overlap with the reduce-task red and orange lines).

A comparison of Figs. 4.4a and 4.4b shows that containers were allocated to reduce tasks as early as 46 sec when reduce slow start was 5% (i.e., after 4 map tasks complete), but only after 143 sec, when reduce slow start was 80% (i.e., after 64 map tasks complete). With the 5% reduce slow start, the job completion time was longer (357s vs. 305s) and the completion of all map tasks was longer (230s vs. 180s). The extra time incurred for shuffling data by

starting later was not significant. The computation portion of the reduce tasks was the same in both runs.

The conclusion from this example is that if a significantly higher-rate connection, e.g., an optical circuit can be setup just for the shuffle phase for such jobs then high values of reduce slow start, e.g., 80% can be used to improve computational efficiency. In the 5% reduce slow start run, the containers assigned to reduce tasks were operating at a low CPU usage level while awaiting the completion of the remaining map tasks. Besides causing lower CPU utilization, the completion time was higher in the 5% run. This experiment illustrates that if the shuffle phase can be sped up, then both higher CPU utilization and lower job completion time can be achieved. However, since the costs of increasing the rates of all links in a datacenter network are high, our proposed solution calls for adding only a limited number of high-speed links in conjunction with an optical circuit switch, and using this additional network infrastructure just for shuffle-heavy jobs to realize gains in CPU utilization and job completion times.

4.5 Modified Hadoop for OSM

Certain aspects of Hadoop need modifications in order to leverage the benefits of the OSM architecture and avoid its pitfalls. These modifications include: (i) a new HDFS data-block placement policy for *shuffle-heavy datasets*, which are large datasets to which shuffle-heavy MapReduce jobs either are known to, or could potentially, be submitted, (ii) a new rack-queue based task scheduling algorithm for YARN/job AMs, (iii) shuffle decoupling and reduce task scheduling, and (iv) shuffling over optical circuits.

Data-block placement policy The default data-block placement policy used in HDFS randomly stores input data blocks across the cluster, resulting in a scattered distribution of map output across many racks (recall that map tasks are run on the nodes on which blocks are stored). Such a scattering of map output runs counter to the state desired in the OSM architecture, which is a concentration of map output to a few racks. Such a concentration is required so that the aggregate map output size on a rack is large enough to justify the overhead of circuit setup delay for shuffle.

Even in the OSM architecture, it seems reasonable to retain the assumption that map input data should not be moved because such an initial data transfer will add an unnecessary additional delay. But if map tasks are run on the nodes on which the input data blocks are stored, the only way to ensure a concentration of map output to a few racks is to execute a preemptory move by changing the HDFS data-placement policy to concentrate map-input data blocks to a few racks instead of scattering these blocks to many racks.

HDFS recommends the use of multiple (default: 3) replicas for reliability reasons. In our modified HDFS, when a large dataset is submitted for storage, the data placement algorithm looks for a rack with the most amount of available disk space in $\mathbf{N}_o$, the set of racks with links to the OCS. No two replicas of any block of a dataset are stored on the same rack.

Rack-queue based task scheduling In current Hadoop, a cluster-wide queue is used to hold container requests from all jobs, and node-local and rack-local assignments of map tasks to nodes on which their corresponding dataset blocks reside are achieved through mechanisms such as delay scheduling [67]. However, as analysis shows dataset blocks should be spread among many nodes to increase the probability of node- or rack-local container assignments. Such spreading is counter to the OSM architecture need for concentrating dataset blocks. Therefore, map task scheduling cannot rely on delay scheduling to achieve rack locality.

In response, we propose a modification to YARN to support *per-rack queues* to which i) task requests from only shuffle-heavy jobs are allowed, and ii) tasks in the rack queues get priority over tasks in the cluster queue. Without these privileges, it will be challenging for the YARN scheduler to concentrate map tasks to a small set of racks. If the set $\mathbf{N}_o$ is the whole cluster, and the load of shuffle-heavy jobs is a considerable fraction of the total work load, then thresholds should be instituted to prevent starvation of non-shuffle-heavy jobs.

Fig. 4.5 shows that *per-rack queues* are maintained for a rack set $\mathbf{R}_{SH}(t)$, which consists of racks currently serving shuffle-heavy jobs. The set $\mathbf{R}_{SH}(t)$ changes dynamically as shuffle-heavy jobs enter and leave the system. As the modified HDFS uses racks in the set $\mathbf{N}_o$ with the most amount of available space when storing a new large dataset, at any instant in time, any of the racks in set $\mathbf{N}_o$ could be in set $\mathbf{R}_{SH}(t)$.

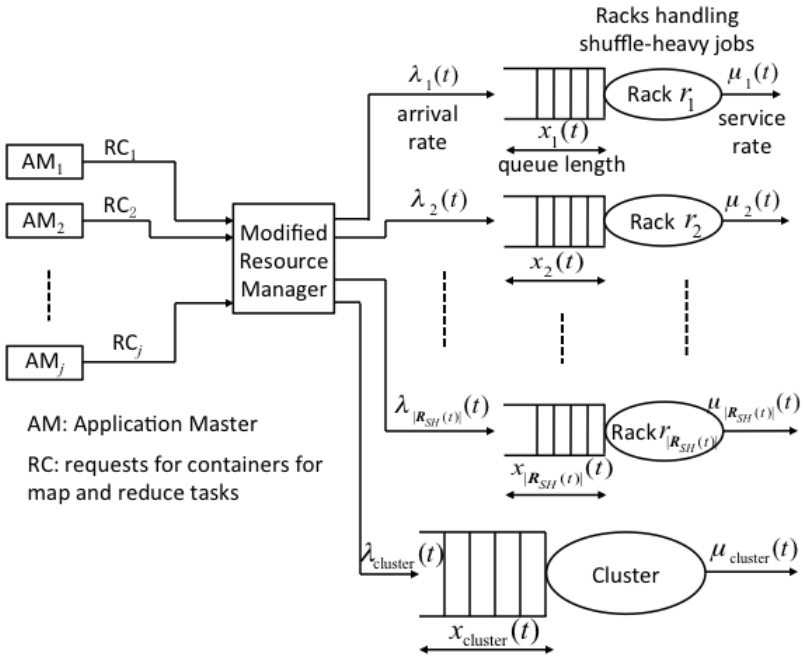


Figure 4.5: Per-rack queueing for shuffle-heavy jobs in modified Hadoop

The AM of a shuffle-heavy job submits a request for the total number of containers required for all its map tasks to the modified RM as shown in Fig. 4.5. The request also specifies the set of racks on which these containers should ideally be assigned (determined by the AM querying the HDFS for its input-block locations). The RM will take into account user/job fairness, and the rates at which tasks depart from the different per-rack queues in deciding how to divide an AM's request for containers between its specified racks. For example, the AM of the TeraSort job described in Section 4.4 cannot decide how best to divide its request for 80 containers to multiple (at least 3 racks if there were 3 replicas of the dataset) racks. The RM can leverage its knowledge to divide this request for 80 containers in a manner that reduces job completion time without adversely affecting other jobs.

The proposed modifications to support only rack-local queues and not node-local queues is because waiting times will be longer in node-local queues, and these longer waiting times are unjustifiable if intra-rack disk-I/O access rates are close to local disk-I/O access rates [77].

Shuffle decoupling and reduce task scheduling In current Hadoop, shuffling is done by the reduce tasks. Each reduce task only pulls the subset of the map output that it requires. If reduce tasks are assigned containers on arbitrary racks, then the size of map output that

needs to be shuffled to any single rack may not justify the overhead of optical circuit setup. Therefore, a modification is needed both to the rack selection process for reduce tasks, and to the shuffling process.

Just as blocks of a shuffle-heavy dataset, and corresponding map tasks, are clustered into a small set of racks, reduce tasks should also be concentrated to a few racks so that map output can be shuffled via optical circuits. We propose limiting the number of reduce racks to a small value, e.g., 1 to 3, because of the constraint on K, the number of ports on the core EPS connected to the OCS in the middle.

An interesting co-scheduling problem arises between map task completions, container assignments for reduce tasks, and scheduling optical circuits for the shuffle phase. We propose a solution to this problem by decoupling shuffle from reduce tasks, and integrating the shuffle function into the YARN NM. But before we describe how the AM of a job uses one or more NMs to accomplish shuffling, first, we need to address how racks are selected for the reduce tasks.

The job AM knows the number of reduce containers it needs, but it has no way of selecting specific racks on which to request these containers. But the AM needs to know the reduce racks in order to coordinate the steps required to shuffle the map output. Therefore, in our modified Hadoop, an AM first submits a request for a specified number of reduce containers, and then waits for the RM to decide which racks to use. If there are racks in set $\mathbf{N}_o \setminus \mathbf{R}_{SH}(t)$ that can be spared from the cluster queue, the RM can choose to use a subset of these racks, pending fairness considerations, since these racks will have shorter wait times. If no such racks are available, the RM will have to select a small subset of racks from its current $\mathbf{R}_{SH}(t)$ set. The RM will inform the AM through the modified AM-RM protocol of the selected set of racks so that the AM can complete the shuffling task.

Meanwhile, to avoid containers from idling while waiting for shuffle to complete, the RM marks reduce-task requests as being in a `waiting` state, which is then modified to a `ready` state upon receiving notification from the AM that the shuffle phase is complete. In the interim, the RM can skip container requests in the `waiting` state when assigning freed containers.

Shuffling over optical circuits After the AM receives the list of racks on which its reduce tasks will be assigned containers, the AM can start shuffling map output from the map racks to the selected reduce racks. Our proposed solution for shuffling over optical circuits uses three ingredients: (i) advance reservation of circuit resources, (ii) storage at the core EPS layer, and (iii) reliable multicast through the core EPS.

Since the number of containers to be allocated on each reduce task is as yet-unknown, there is no easy way to determine a method for dividing map output and moving different portions to different reduce racks. Instead, given the availability of the higher-rate transceivers on the optical circuits, we propose to move the whole map output from a map rack to each of the identified reduce racks. This solution offers the RM flexibility to assign freed containers on a reduce rack to any reduce task. Furthermore, the OSM architecture allows for a L2 multipoint VLAN to be created in the core EPS (interconnecting point-to-point optical circuits from each TOR switch to the core EPS), and then sending the map output from each map rack via a *reliable multicast* transport protocol simultaneously to all reduce racks.

The multilayer SDN controller maintains an *advance-reservation* window for: (i) the link between each TOR switch in set $\mathbf{N}_o$ and the OCS, (ii) each of the K links between the OCS and core EPS, and (iii) the link from core-EPS to *storage* (see Fig. 4.1b). Since NMs on the map and reduce racks are always available, depending on circuit resource availability, the multilayer SDN controller will choose the shorter of two options: (i) wait for all the ToR-to-core optical circuits to become available to then create an L2 multipoint VLAN for a reliable multicast of the map output from each map rack simultaneously to all the reduce racks, or (ii) use a circuit from the map-rack TOR switch to the core EPS and move the map output to the storage depot, and then use subsequent circuits to move the map output either individually, or in a reliable multicast over an L2 multipoint VLAN, from the storage depot to the reduce racks.

When the map output files from all map racks have been moved to a reduce rack, the AM can notify the RM to change the state of the request for containers on that rack to the `ready` state for immediate allocation to reduce tasks.

4.6 Summary

This chapter presented a novel Optical Switch in the Middle (OSM) hybrid electrical-packet/optical-circuit architecture for datacenter networks. Features such as integrated storage in the core EPS, advanced-reservation scheduling in a multilayer SDN controller, and L2 multicast are part of the OSM architecture. Experiments with a shuffle-heavy Hadoop MapReduce application showed that features such as reduce slow start are used to counter long communication delays incurred in the shuffle phase, but that this feature sacrifices CPU utilization. Four modifications were proposed to adapt Hadoop to the OSM architecture. Our next step is to undertake a simulation study to evaluate Hadoop-in-OSM.

Chapter 5

Evaluation Study of Hadoop for Hybrid Networks (HHN)

5.1 Introduction

In this chapter, we present a simulation-based evaluation of the modified Hadoop (which was presented in Chapter 4 and is referred to as HHN) on hybrid networks, and compare its performance with that of original Hadoop on EPS-only networks. Bulk of this material is presented from our published work *An evaluation study of a proposed Hadoop for hybrid networks (HHN)* [17] © 2017 IEEE, and *Evaluation study of a proposed Hadoop for data center networks incorporating optical circuit switches* [18] © 2018 IEEE.

Our simulation approach for performance comparison of HHN and original Hadoop on EPS-only networks consisted of: (i) creating a variety of workloads by mixing synthetic regular jobs (those with a shuffle size less than 2 GB) with SH jobs drawn from the real-world Facebook-2010 traces [78], with different ratios of regular jobs to SH jobs; (ii) using different system (network) configurations, e.g., 4-rack and 12-rack systems, and 75 and 100 for the percentage of ToR switches that are connected to the OCS in the hybrid network; and (iii) changing the job arrival rate to study the system under high levels of CPU utilization. The performance metrics used for the comparative evaluation included: (i) system metrics, such as makespan and CPU utilization, and (ii) per-job metrics, such as response times and

unfairness.

Given the cost and power advantages of hybrid networks over EPS-only networks, we identified workloads in which HHN performance was worse than the original Hadoop performance on EPS-only networks, and characterized the performance degradation under different system configurations and different CPU loads. We then evaluated the benefit of changing a critical Hadoop parameter, the number of replicas used for storing datasets. A small increase from 2 to 3 yielded significant performance improvements for HHN.

Our *key findings* are as follows: (i) hybrid networks can achieve significant savings in cost and power consumption when compared to EPS-only networks; (ii) restricting SH datasets to a few racks in order to concentrate map output so that optical circuits can be used in the shuffle phase of MapReduce jobs can cause increased waiting delays for containers, and consequently increase SH job response times and job unfairness; (iii) as a consequence, if the percentage of SH jobs in a workload is high, e.g., 20%, or there are very large SH jobs (i.e., jobs that require processing of very large datasets), the limitation on the number of racks from which containers can be assigned to SH-job map tasks could result in longer makespans and lower CPU utilization because hosts in racks that do not have SH datasets could be idle; (iv) the relative degradation of per-job response times in HHN is smaller in larger systems, i.e., networks with more racks, and lowering the percentage of ToR switches connected to the core OCS favors regular jobs over SH jobs and vice versa; (v) HHN performance can be improved significantly by increasing the number of input-block replicas even by just 1, e.g., from 2 to 3; and (vi) In small systems at high loads, without container preemption, original Hadoop could enter a deadlock leading to significantly longer response times and makespan, while in contrast, HHN handles the problem elegantly by decoupling the shuffle phase from reduce tasks.

The rest of the chapter is organized as follows: Section 5.2 reviews our proposed modifications to Hadoop for matching traffic to the characteristics of hybrid networks. Our comparative evaluation of this modified Hadoop with original Hadoop is presented in Section 5.3. Section 5.4 presents our conclusions.

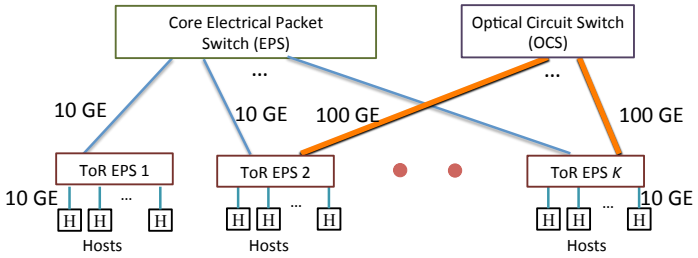


Figure 5.1: An example hybrid EPS-OCS DataCenter Network (DCN) architecture

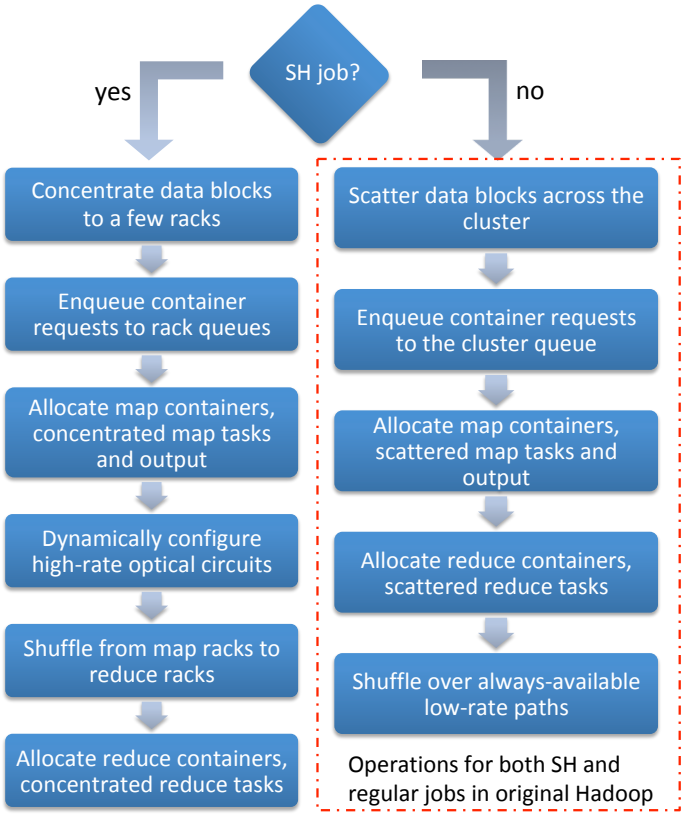


Figure 5.2: Flowchart of HHN

5.2 Our Proposed Hadoop for Hybrid Networks

Fig. 5.1 illustrates an example hybrid network architecture for which HHN is designed. Lower-rate (10 GE) links that connect ToR EPSs and the core EPS are used for general-purpose traffic, while higher-rate (100 GE) links between ToR EPSs and the OCS are used in ToR-to-ToR dynamic optical circuits setup/released through the OCS by a controller (not shown in Fig. 5.1). Not all ToR switches need to be connected to the OCS, e.g., ToR1 is not connected to the OCS. Our model assumes that ToR EPSs in K_o racks in a system of K

racks, where $K_0 \leq K$, are connected to the OCS.

To make Hadoop work effectively in such a hybrid network, we proposed modifications to the following [16]: (i) how a dataset is stored by HDFS, (ii) how the scheduler assigns containers to map tasks of SH jobs, (iii) how the scheduler assigns containers to reduce tasks of SH jobs, and (iv) how map output is shuffled over optical circuits. The different operations for SH and regular jobs in HHN are summarized in Fig. 5.2. See Section 4.5 for detailed description of HHN.

5.3 Evaluation

We evaluate the performance of the HHN solution by comparing it against the original Hadoop in an EPS-only network. To achieve a fair comparison, we assume that the ToR-to-core links in the EPS-only network have the same capacity as the transceiver rates in the OCS segment of the hybrid network. Our *hypothesis* is that, compared to the EPS-only network, the HHN solution can offer almost equivalent job performance but with power and cost savings.

To test the hypothesis, we first analyze the price and power consumption of the two types of DCN architectures (see Section 5.3.1), and then conduct a detailed simulation study to compare job performance, which could potentially be worse in the HHN solution. In the EPS-only network, all links are of high-rate and are always available; in contrast, in the hybrid network solution, the high-rate circuits have to be setup dynamically across the OCS when needed, and therefore, we expect job performance to be worse in HHN.

The purpose of our simulation is to quantify job performance, and recommend parameter settings to achieve the same level of performance as with original Hadoop on EPS-only networks. Results showed the validity of our hypothesis.

Section 5.3.1 compares the price and power consumption of hybrid and EPS-only DCN architectures. Section 5.3.2 describes our simulator, input parameters, workloads and evaluation metrics. Section 5.3.3 provides an in-depth analysis of a single SH job execution, while the remaining subsections present simulation studies with multiple SH and regular jobs. Section 5.3.4 presents the results of our comparison of HHN and original Hadoop

Table 5.1: Input parameters for a comparison of three DCNs

Components	Number of ports needed in different architectures			Price (USD)	Power(W)
	hybrid-100%	hybrid-75%	EPS-only	per port	per port
OCS port [1]	K	$0.75K$	-	400	0.15 (lower), 0.6 (higher)
10G EPS port	$2K$	$2K$	$2K$	-	-
10G SR transceiver	$2K$	$2K$	$2K$	-	-
100G EPS port [2]	K	$0.75K$	$2K$	900 (lower), 2000 (higher)	12.5 (lower), 42 (higher)
100G SR transceiver [3]	K	$0.75K$	$2K$	2500	1.5
fiber [4]	$2K$	$1.75K$	$2K$	13	0

[1] The price and power consumption values for an OCS port were obtained from Calient for the S320 OCS [79], and for Glimmerglass Intelligent Optical System 600 [80], respectively.
[2] The lower and higher prices for a 100G EPS port were obtained for Arista 7160 and 7280SRAM-48C6 [81], respectively. The power consumption values were obtained from datasheets for Cisco Nexus 7700 [82], Juniper QDX10002 [83], Arista 7280SRAM-48C6 [81], and Huawei CloudEngine 12800 [84].
[3] The price and power consumption values for a 100G transceiver were obtained for Arista QSFP-100GBASE-SR4 [85], and Cisco QSFP-100G-SR4 [86], respectively.
[4] The price of fiber was obtained from fs.com [87].

for a baseline setting of system parameters. Section 5.3.5 presents the effects of changing two key system parameters. Section 5.3.6 presents the effect of changing one key Hadoop parameter. Section 5.3.7 presents generalized results for multiple traces with the same parameter settings.

5.3.1 Power and cost evaluation

This subsection presents a differential power and cost comparison of example hybrid and EPS-only DCNs. Since the down-link ToR switch ports (ports connected to the servers) are the same in all DCNs, these ports are omitted from the comparison.

Two configurations of the hybrid architecture with K ToR switches, as illustrated in Fig. 5.1, are modeled here: hybrid-100% and hybrid-75%, where the 100% and 75% values denote the percentage of ToR EPS connected to OCS in the two configurations, respectively.

Table 5.1 lists the *number of different types of ports* in the hybrid-100%, hybrid-75%, and EPS-only DCNs. The OCS is present only in the hybrid DCNs, and the number of OCS ports in these hybrid DCNs depends on the percentage of ToR EPS connected to the OCS. The total number of 10G EPS ports (including all the ToR switch and core EPS ports) is $2K$ in all three DCNs. For a fair comparison, we assumed that the ToR-to-core capacity in the EPS-only DCN is 110 G (100G + 10G) per ToR switch to match the total ToR-to-core

Table 5.2: Cost and power consumption comparison of three 100-rack DCNs

Architectures	**Cost (USD)**		**Power (kW)**	
	lower	higher	lower	higher
EPS-only (baseline)	\$682.6K	\$902.6K	2.8	8.7
hybrid-100% (savings)	\$300.0K	\$410.0K	1.4	4.3
hybrid-75% (savings)	\$470.3K	\$532.8K	1.7	5.4

capacity in the hybrid-100% architecture. The number of 10G and 100G transceivers are the same as the number of 10G and 100G ports, respectively. The number of 100G EPS ports is $2K$ in the EPS-only DCN since 100G ports are required in the ToR switches (for the uplinks) and the core EPS. But, in the hybrid DCNs, the OCS ports terminate the 100G uplinks from the ToR switches, and hence only K and $0.75K$ 100G EPS ports are required at the ToR switches, in the hybrid-100% and hybrid-75% DCNs, respectively. The number of fiber links required in the EPS-only DCN is $2K$ because as stated above, we assumed that each ToR switch has two uplinks: 10G and 100G. In the hybrid DCNs, one fiber is required from each ToR switch to the core EPS, and a second fiber is required from each of the ToR switches that is connected to the OCS.

Next, we explain how we obtained the *price and power-consumption values* listed in Table 5.1. The OCS-port price was obtained in June 2018 for a 320×320 switch. Since the total number of 10G EPS ports is $2K$ in all three DCNs, the price and power-consumption values are not required for our differential comparison, and thus not listed in Table 5.1. Since price and power consumption values are variable, we offer two values and mark them as "lower" and "higher." These values are not necessarily the minimum and maximum values, since some vendors offer discounts, and other vendors offer products without warranties or maintenance contracts. Table 5.1 shows two June-2018 prices for a 100G EPS port: \$900 and \$2000, which correspond to per-port prices of a standard switch vs. a deep-buffer switch. The amount of buffer space and the switch sizes account for the difference in per-port prices. Transceiver prices vary significantly. Third-party vendors offer lower-priced transceivers, but without warranties. The transceiver price listed in Table 5.1 was obtained directly from a switch vendor in June 2018. All prices are retail values, including warranties, and are without discounts.

Table 5.2 compares the cost and power consumption of the hybrid and EPS-only DCNs,

assuming a system size of 100 racks. We present a baseline cost and power-consumption value for the EPS-only DCN (these values do not represent total price or total power consumption since 10G ports were omitted), and the savings achieved in the hybrid DCNs when compared to the EPS-only DCN. The cost savings of the hybrid-100% DCN over EPS-only network were $300,000 and $410,000, when using the lower and higher values for component prices, respectively. For the hybrid-75% DCN, the cost savings are even higher. Similarly, the power savings of hybrid-100% and hybrid-75% DCN over EPS-only DCN are 4.3 kW and 5.4 kW, respectively, when using the higher numbers for component power-consumption values. The additional cost and power savings of hybrid-75% DCN over the hybrid-100% DCN come from the smaller number of OCS ports needed in the hybrid-75% DCN. Finally, since optical switches generate less heat than electrical switches, hybrid DCN architectures can achieve additional cost savings in cooling systems.

5.3.2 Simulation methodology

Simulator We implemented an event-based simulator model of HHN. The HDFS-simulation module allows all blocks of a SH dataset to be stored in a specified number of racks. The YARN-simulation module supports per-rack queues for SH jobs to request containers for map and reduce tasks, while regular jobs enqueue their container requests in a cluster queue. Hadoop fair scheduler with delay scheduling is used to allocate containers for tasks in the cluster queue.

The network model is fairly coarse. The required optical links are assumed to be available whenever needed. To estimate the time to move map output for SH jobs, the size of map output is divided by the optical network transceiver rates, and a switch reconfiguration delay of 10 ms is added. For map output of regular jobs, which is transferred on paths traversing only the ToR and core EPS, flow instantaneous rates are computed by dividing the rate of the link carrying the maximum number of concurrent flows by the number of flows. Flow rates are updated every 1 ms. Map output transfer time for regular jobs is computed from the total map output size and the per-ms transfer sizes.

For comparison, we also simulated original Hadoop on an EPS-only DCN. In the original Hadoop system, resource requests of both shuffle-heavy jobs and regular jobs are enqueued in a single cluster queue, which is served by the Hadoop fair scheduler with delay scheduling.

The simulator, written in Python, has 1000+ lines of code. All simulation runs were executed on a University of Virginia HPC cluster called Rivanna [88].

Table 5.3: Simulation parameters

System parameters	**Value**
Number of racks	2[†], 4, 12
Number of hosts per rack	20
Number of containers per host	16
K_o/K (percentage of ToR EPS connected to OCS)	75%, 100%
Intra-rack link rate	8 Gbps
Inter-rack EPS link rate in hybrid network	10 Gbps
Optical link rate in hybrid network	100 Gbps
Inter-rack link rate in EPS-only network	110 Gbps
Hadoop parameters	**Value**
Number of replicas of each input block	2, 3[‡]
Reduce slow start	90%

[†] Only in Section 5.3.3
[‡] Number of replicas set to 3 only in Section 5.3.6

Parameters The default values of simulation parameters are shown in Table 5.3. Unless otherwise specified, these default values are used in all the runs. Specifically, we chose the EPS-only link rates (110 Gbps) to be the sum of the EPS-link (10 Gbps) and OCS-link rates (100 Gbps) of the hybrid network. The intra-rack link rate used in computing transfer times is 8 Gbps becaused we assumed the background traffic rate to be 2 Gbps on the 10 Gbps intra-rack links.

Workloads We started with the Facebook 2010 (FB-2010) workload, which provides the following information for each job: (i) arrival time instant, (ii) input dataset size, (iii) shuffle data size, and (iv) reduce output size. Assuming that each map task processes one input block of size 128 MB, and that the number of reduce tasks is equal to the number of map tasks divided by 8, we derived the number of map tasks and number of reduce tasks for each job from the size of its input dataset.

In the FB-2010 workload, more than 50% of the jobs are very small jobs, with only one or two map tasks. With the given job arrival times, the original workload results in low CPU utilization, i.e., around 10%, in our simulation. To achieve higher CPU utilization levels, we

Table 5.4: Trace composition; RJS: Regular Job Sets

No. of maps	**Percentage of job types**	
	RJS1	RJS2
1-9	40%	20%
10-99	40%	50%
100-499	18%	28%
500-10000	2%	2%
Shuffle size		
0	10%	
0-0.8 GB	70%	
0-2 GB	20%	

	TS1	TS2
Regular jobs	RJS1	RJS2
SH jobs in Sections 5.3.4 to 5.3.6	first 40 SH jobs in FB-2010 workload	first 60 SH jobs in FB-2010 workload with input size $<$ 800 GB
SH jobs in Section 5.3.7	-	randomly picked SH jobs in FB-2010 workload with input size $<$ 800 GB

generated two *Trace Sets*, *TS1* and *TS2*, which consist of larger (artificially created) regular jobs, and SH jobs that were directly drawn from the FB-2010 workload. The composition of two Regular Job Sets (RJS) are shown in Table 5.4. Uniform distributions are used to select the group (based on number of map tasks), and the specific number of map tasks within the selected group. Uniform distribution is used similarly to select the shuffle size of a regular job. We defined jobs with a shuffle-data size larger than 2 GB as shuffle-heavy jobs because the duration to transfer this data on 100 Gbps links is sufficiently longer than the 10-ms optical circuit setup delay overhead. The compositions of trace sets TS1 and TS2 are shown in Table 5.4. We used the first 40 SH jobs from the FB-2010 workload in TS1. For TS2, we included the first 60 SH jobs whose input-data sizes were smaller than 800 GB because we found that one very large SH job can skew the results, as described in Section 5.3.4.

Each trace in the trace sets was generated by varying two parameters, i.e., job arrival rate λ and SH-job percentage p_s. The inter-arrival times of jobs were decided by drawing samples of an exponentially distributed random variable with parameter λ. The SH-job percentage p_s was used to set the percentage of SH jobs in a trace. For each job, a Bernoulli distributed sample with parameter p_s was drawn to decide whether the job should be a SH job or a regular job. If it was a SH job, then its parameters were taken from one SH job in the FB-2010 workload following certain rules shown in Table 5.4. For example, all traces in TS1 have the same 40 SHJs, in the same *relative order*. When generating traces for TS2,

the SH jobs in each trace are *randomly* selected from the SH jobs in the FB-2010 workload that have input-data sizes smaller that 800 GB.

Evaluation metrics We used two types of metrics, per-job metrics and system metrics. Per-job metrics include *job response time* and per-job *unfairness*. The system metrics used to characterize the overall performance of the system are *makespan* and *CPU utilization*.

Job response time is defined as $t_j^c - t_j^a$, where t_j^a is the arrival instant of job j and t_j^c is the job completion time. Per-job *unfairness* is defined as:

$$f_j = \int_{t_j^a}^{t_j^c} \max\left\{ d_j(t) - \frac{a_j(t)}{R}, 0 \right\} dt \tag{5.1}$$

where

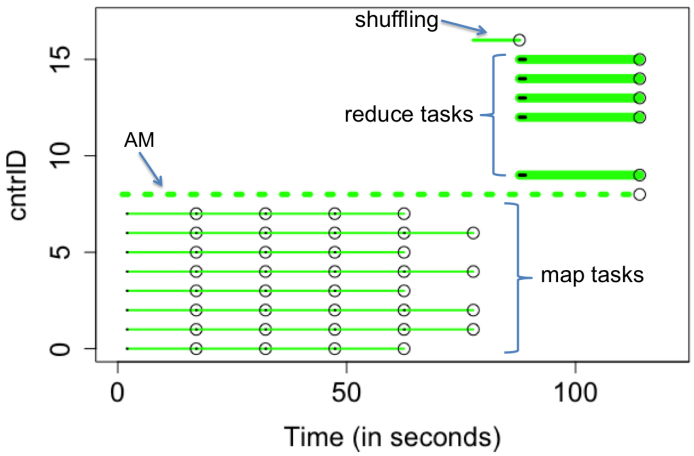
$$d_j(t) = \min\left\{ \frac{1}{N(t)}, \frac{r_j(t)}{R} \right\} \tag{5.2}$$

At time instant t, the percentage of resources deserved, $d_j(t)$, by job j (from a fairness point of view), depends upon the number of jobs $N(t)$ in the system, and the ratio of the resources requested $r_j(t)$ to the number of system resources R. For example, if a system has only 2 jobs, and 1 job requires 5 containers and the second job requests and receives the remaining 10 containers in the system, the percentage of resources deserved by the first job is 1/3, not 1/2. The instantaneous unfairness of a job is the difference between the amount of its deserved resources and the amount of its allocated resources, $a_j(t)$. Per-job unfairness f_j is computed by integrating its instantaneous unfairness over the job's lifetime.

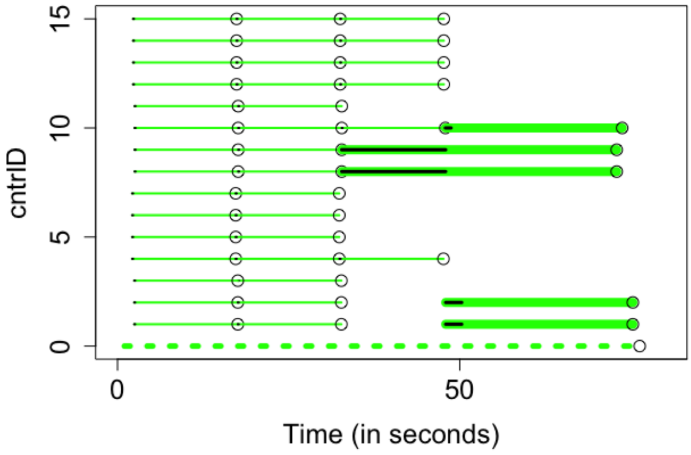
Makespan is defined for a trace consisting of J jobs as $t_J^c - t_1^a$. *CPU utilization* for a trace of J jobs is the average utilization of all containers in the system over the time period of $[t_1^a, t_J^c]$.

5.3.3 Effect of clumping on a single shuffle-heavy job

Using the modified data-block placement policy, the input dataset of a shuffle-heavy job is concentrated to a few racks, which limits the amount of computing resources accessible for the job. To study the effect of input-data clumping, we start with a single shuffle-heavy job in a small system. We simulated a cluster of two racks, in which there are a total of 16



(a) HHN in the hybrid network



(b) Original Hadoop in the EPS-only network

Figure 5.3: Start and finish times of map tasks, reduce tasks and shuffling of a single shuffle-heavy job

containers indexed from 0 to 15. The SH job consists of 36 map tasks and 5 reduce tasks. In HHN, the input dataset is stored only in the first rack, while the dataset is stored in both racks for original Hadoop. The optical link rate is 5 Gbps in HHN. The inter-rack electrical link rate is 500 Mbps in the hybrid network and 5.5 Gbps in the EPS-only network. Here we use lower link rates when compared to the values listed in Table 5.3 because we simulate only one job, and use a smaller system (with only 16 containers).

Fig. 5.3 illustrates how containers are allocated to the SH job when it runs on the two networks. The dashed line represents the AM container. The thin and thick lines in green correspond to map-task containers and reduce-task containers, respectively. The black

Table 5.5: Comparison of HHN with original Hadoop in the EPS-only network for TS1 traces on a 12-rack system

SH-job percentage p_s	Job arrival rate λ (/sec)	Trace properties		Makespan (s)		CPU utilization (%)	
		Last-job arrival time (s)	Number of jobs in trace	HHN-75%	Original Hadoop	HHN-75%	Original Hadoop
5%	0.3	3097	800	3140.4	3139.7	28.4	28.4
	0.6	1767	804	1810.4	1809.8	50.0	49.9
	0.9	1354	801	1403.5	1397.0	63.3	64.1
10%	0.3	1579	397	1644.1	1622.2	29.2	29.7
	0.6	890	399	1151.0	954.8	48.0	50.7
	0.9	688	402	1092.0	1005.2	57.3	59.0
20%	0.3	754	197	1132.3	797.1	29.9	34.9
	0.6	438	201	922.4	598.1	46.3	57.3
	0.9	332	202	830.0	560.1	60.7	72.8

segments shows the time period when map output is being shuffled. The job is completed faster in original Hadoop than in HHN (75s vs. 113s). This is because the job can only use the 8 containers in the first rack to execute map tasks due to its concentrated input dataset in HHN, while it can use all the 15 containers (except for container 0 used by the AM) to execute map tasks. On the other hand, thanks to the decoupled shuffle phase from reduce tasks, reduce containers do not need to sit idle when waiting for the shuffle phase to finish with the modified Hadoop (see shorter black segments in Fig. 5.3a than in Fig. 5.3b). Next, we study how multiple SH jobs and regular jobs interact.

5.3.4 Comparison in a baseline setting

The system parameters and Hadoop parameters in this baseline setting are as specified in Table 5.3, with the number of racks set to 12, and the percentage of ToR EPS connected to OCS set to 75%. The notation *HHN-75%* is used to represent this configuration.

The job traces used for these runs were generated with TS1 settings. A total of 9 different traces were generated by combining three values of λ and three values of p_s. Generation of jobs for a trace was terminated when the 40 SH jobs from the FB-2010 workload were included as per our TS1-workload specification (see Workloads paragraph in Section 5.3.2).

Table 5.5 first shows two trace properties and then compares makespan and CPU utilization for the two Hadoop versions. The last-job arrival time is useful to interpret the makespan. The reason for the difference in the number of jobs in traces is as follows. When

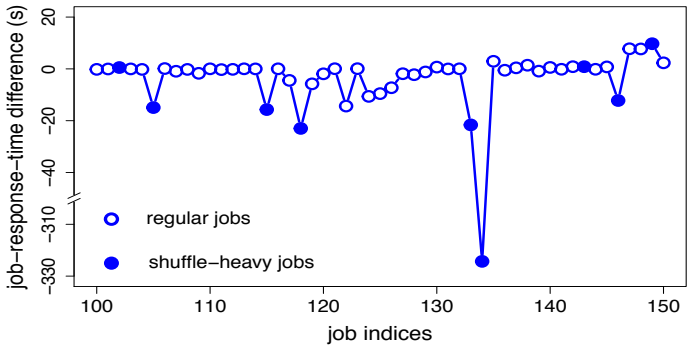
SH jobs constitute only 5% of the trace, approximately 800 jobs were required before the 40 SH jobs from FB-2010 could be included, while with 10% and 20% of the trace being SH jobs, approximately 400 and 200 jobs, respectively, were required to include the 40 SH jobs.

System metrics The makespan in the two solutions, HHN-75% and original Hadoop (in the EPS-only network), are almost the same when the trace has a large number of jobs, e.g., 800. This is because in all 9 traces (recall that the same 40 SH jobs were included in all TS1 traces), there was one large SH job with 19000 map tasks, which arrives in the second half of the traces. In HHN, each SH job can use containers in only two racks (since each dataset has only 2 replicas; see Table 5.3). This results in longer job response times for SH jobs than in the original Hadoop solution where SH jobs can use containers in any rack.

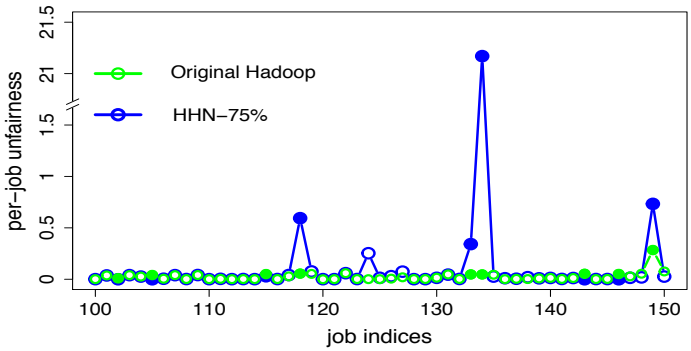
With the longer traces, this large SH-job response time was hidden by the large number of jobs that came after it, resulting in the same makespan. However, with shorter traces (i.e., the 200-job traces), the large SH job was still running when all the other jobs had completed. The makespan difference between the two networks is 324 s under the setting p_s = 20% and $\lambda = 0.6$ /sec. This difference is almost equal to the job response time difference for the large SH job.

Table 5.5 also shows CPU utilization of the two types of systems for each trace. When the percentage of SH jobs in the trace is small, e.g., 5%, CPU utilization is the same in the HHN-75% and original Hadoop solutions, since regular jobs dominate and these jobs can be assigned containers in any rack. But when that percentage increases to 20%, with the constraint of assigning containers in only 2 racks for each SH job, CPU utilization is lower in the HHN-75% solution.

Per-job metrics Fig. 5.4a shows the difference in job response time between original Hadoop and HHN-75% for a range of 51 jobs (job 100 to job 150) in the trace. Most of the jobs with longer response times in HHN-75% than in original Hadoop were SH jobs. This is because containers in a maximum of 2 racks can be assigned to SH-job map tasks. On the other hand, this constraint sometimes helps the regular jobs that arrive near SH jobs to finish faster in HHN-75%, e.g., job 135 finishes 2.9 sec earlier in HHN-75%. Job 134, which



(a) Job response time in original Hadoop minus that in HHN-75%



(b) Per-job unfairness

Figure 5.4: Performance comparison of HHN-75% and original Hadoop in a 12-rack system, 50 jobs in a TS1 trace with $p_s = 20\%$, $\lambda = 0.6$ (view in color mode)

has the largest response time difference (i.e., 324 s), is the large SH job that caused the makespan difference discussed earlier.

Fig. 5.4b shows per-job unfairness for the two solutions. Overall, the Hadoop fair scheduler with delay scheduling used in the EPS-only original Hadoop solution achieves better fairness. In the HHN-75% solution, the unfairness of large SH jobs is higher because these jobs are constrained to use containers in only two racks. Even though the modified YARN in the HHN solution offers SH jobs preferential treatment by allowing only SH jobs to place container requests in per-rack queues, SH jobs experience higher unfairness. Comparing Figs. 5.4a and 5.4b, we observe a mirror-like pattern in the two metrics, i.e., unfairly treated SH jobs usually have longer completion times.

Table 5.6: Comparison of system metrics in two HHN configurations and original Hadoop on EPS-only network; $p_s = 20\%$

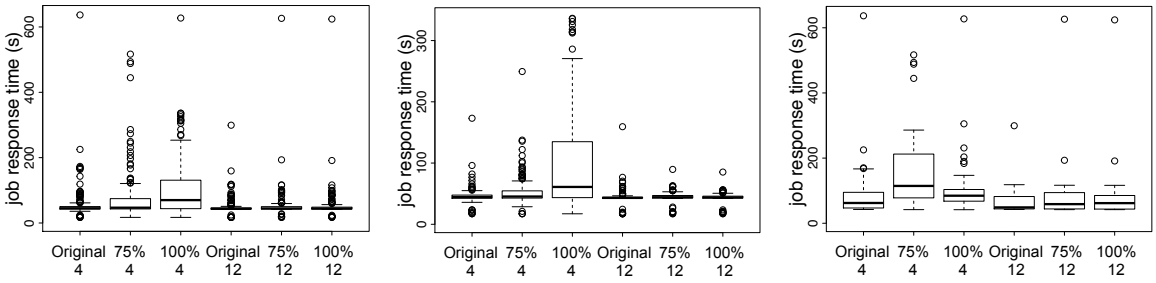
Trace		TS1		TS2	
Metrics		Makespan (s)	CPU utilization (%)	Makespan (s)	CPU utilization (%)
4 racks; $\lambda = 0.3$ ($TS1$), 0.25 ($TS2$)	HHN-75%	1351.3	80.7	1409.8	83.0
	HHN-100%	1346.6	80.1	1403.3	82.7
	Original Hadoop	1141.7	82.7	1409.0	82.4
12 racks; $\lambda = 1.6$ ($TS1$), 1.2 ($TS2$)	HHN-75%	773.4	72.8	487.8	81.2
	HHN-100%	771.2	72.4	489.6	80.8
	Original Hadoop	446.3	83.9	486.8	80.4

5.3.5 Sensitivity to system parameters

We examined the impact of two system parameters on system and per-job metrics: (i) system size, and (ii) the percentage of ToR EPSs connected to OCS in the hybrid network. Two system sizes were used: 4 racks and 12 racks. Two values of the percentage of ToR EPSs connected to OCS were assumed: 75% and 100% (see Table 5.3). These two cases are denoted by HHN-75% and HHN-100%. In HHN-100%, SH datasets are allowed to be stored on all racks, but the number of replicas per dataset is still only 2.

Two types of job traces were used: TS1 and TS2 (see Workloads paragraph of Section 5.3.2). Job arrival rate λ was increased in these runs relative to the values used in the runs described in Section 5.3.4. In selecting λ, we tried to make the CPU utilization in the EPS-only (original Hadoop) solution the same for the 4-rack and 12-rack cases. For the TS1 trace, approximately the same CPU utilization was achieved with λ values of 0.3 /sec and 1.6 /sec for the 4-rack and 12-rack cases, respectively. For TS2, these numbers were 0.25 /sec and 1.2 /sec for the 4-rack and 12-rack cases, respectively.

System metrics Table 5.6 compares the system metrics, makespan and CPU utilization, under different settings. First consider the values obtained for traces generated with the TS1 input. The effect of system size on makespan is as follows. The percentage difference in makespan between the original Hadoop and HHN solutions in a 4-rack system was smaller than in the 12-rack system. This is because the time taken for all jobs to complete in the 4-rack system provided more overlap of the large SH-job execution time with the execution times of other jobs. The effect of the percentage of ToR EPS connected to OCS in the hybrid



(a) All jobs (b) Regular jobs (c) Shuffle-heavy jobs

Figure 5.5: Job response time comparison; 4 and 12: number of racks; Original: original Hadoop on EPS-only network; 75% and 100%: HHN-75% and HHN-100%; TS1 input; p_s=20%; λ=0.3 (4 racks) and 1.6 (12 racks)

network, 75% vs. 100%, on makespan was not significant since the number of overlapping SH jobs was not high.

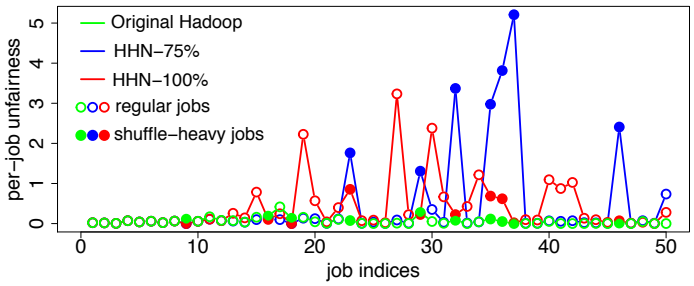
The CPU utilization in the 12-rack case is lower in the HHN solutions. This is because of the 2-rack constraint on SH jobs.

Next, consider the results obtained with TS2 traces. Recall that SH jobs with more than 800-GB input datasets were excluded in TS2. The effects of the one large SH job (the input dataset size for this job was 2.375 TB), which were described in Section 5.3.4, are not seen in the results for TS2. The makespan is almost the same in the original Hadoop, and HHN solutions in both 4-rack and 12-rack cases. Also, the CPU utilization is slightly better in the HHN solution.

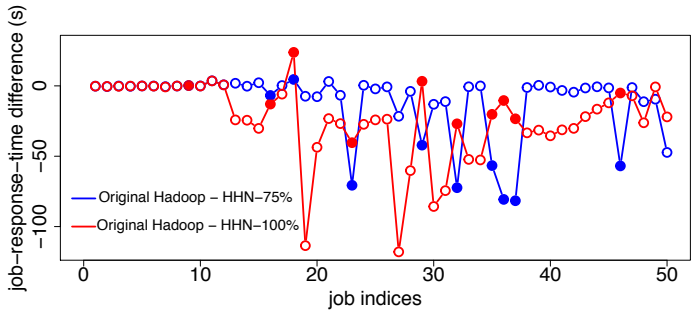
We conclude that for large input datasets, either more than 2 replicas should be created to spread out blocks on more racks (replicas of a block should necessarily be stored on different racks for reliability reasons), or even with just 2 replicas, the datasets should be spread out to more than 2 racks. However, the input datasets should not be so splintered between racks that the per-rack map output becomes too small to justify the use of optical circuits for shuffling.

Job response time Fig. 5.5 shows boxplots to compare job response times for various configurations[1]. The TS1 input was chosen as it had worse results than TS2 as seen in Table 5.6.

[1]One SH job in the 4-rack instance of the HHN-75% configuration took 846.3s. This point was dropped from the graph for better visualization of the differences.



(a) Per-job unfairness



(b) Job response time diff. between original Hadoop and HHN-75%

Figure 5.6: Per-job metrics for the first 50 jobs of a TS2 trace; 4-rack system; $p_s = 20\%$; $\lambda = 0.3$ (view in color mode)

We make the following observations: (i) Larger systems, i.e., systems with more racks outperform smaller systems (the job arrival rate λ values were chosen to make the CPU utilization the same in the 4-rack and 12-racks cases for the original Hadoop configuration). (ii) In smaller systems, increasing the percentage of ToR EPS connected to OCS in the hybrid network affects regular jobs adversely, e.g., the job response time is longer for regular jobs in HHN-100% configuration than in the HHN-75% configuration, while the opposite is true for SH jobs.

Per-job unfairness To gain a better insight into per-job unfairness, we present this metric along with job response time for a particular setting: 4-rack system, TS2 trace, $p_s = 20\%$, and $\lambda = 0.3$. Intuitively, if SH datasets are stored in all the racks of a hybrid network, regular jobs are likely to be treated unfairly, since SH jobs receive preferential treatment with their use of per-rack queues. This effect should be more obvious in a system with a smaller number of racks, because with a larger number of racks, it is less likely for multiple

shuffle-heavy jobs to be scheduled on all the racks at the same time. Thus, we choose the 4-rack configuration to present the results.

Fig. 5.6 presents the results. The original Hadoop on an EPS-only network offers the best fairness. Regular jobs suffer higher unfairness in HHN-100% when compared to HHN-75%. This is reversed for SH jobs.

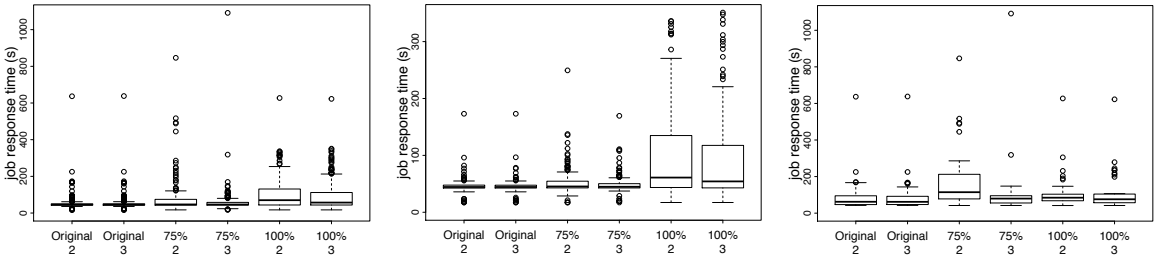
This simulation run illustrates well the effects on job response time of the system parameter, percentage of ToR EPS connected to OCS in the hybrid network. Therefore, Fig. 5.6b has been added to the job unfairness figure. It is apparent that SH jobs enjoy shorter response times in HHN-100% than in HHN-75%, since all 4 racks in HHN-100% can be used to store SH datasets, and hence there is a smaller likelihood of multiple concurrent SH jobs competing for containers in the same 2 racks. With this trace, this exact scenario occurs when three consecutive SH jobs (job 35-37) have to share the same 2 racks based on the location of their dataset replicas. Fig 5.6 shows that unfairness level shoots up in the HHN-75% (blue) configuration for these three jobs, and simultaneously, job response time increases.

Regular jobs enjoy the same short completion times in HHN-75% as they do in the original Hadoop on EPS network, because there is always one rack out of the four racks that does not run SH jobs.

5.3.6 Sensitivity to a Hadoop parameter

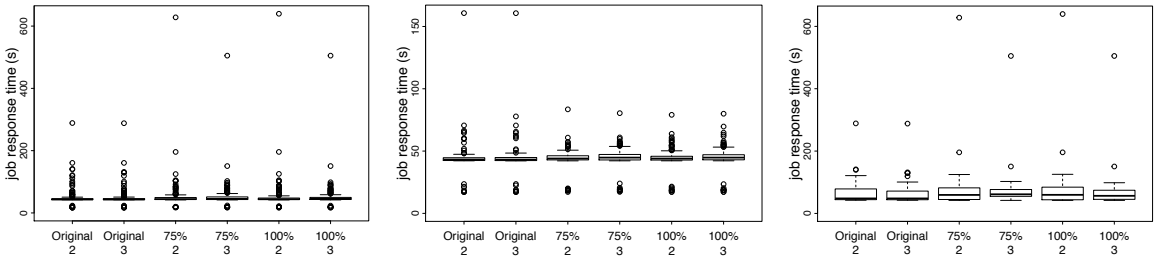
Here we study the impact of one Hadoop parameter, i.e., number of replicas of each input block, on the system and per-job metrics. Two values of the number of replicas were used: 2 and 3 (see Table 5.3). The traces used in this subsection are from TS1.

Job response time Fig. 5.7 and Fig. 5.8 illustrate the effect of the number of dataset replicas on job response time in a 4-rack system and a 12-rack system, respectively. We make the following observations: (i) in the smaller system with 4 racks, if only 75% of the ToR switches are connected to the OCS, and 3 replicas are used, response times for both regular jobs and SH jobs are statistically similar to the response times with original Hadoop on an EPS-only DCN; and (ii) in the larger system with 12 racks, SH-job response



(a) All jobs (b) Regular jobs (c) Shuffle-heavy jobs

Figure 5.7: Job response time comparison in a 4-rack system; 2 and 3: number of replicas; Original: original Hadoop on EPS-only network; 75% and 100%: HHN-75% and HHN-100%; TS1 input; p_s=20%; λ=0.3



(a) All jobs (b) Regular jobs (c) Shuffle-heavy jobs

Figure 5.8: Job response time comparison in a 12-rack system; 2 and 3: number of replicas; Original: original Hadoop on EPS-only network; 75% and 100%: HHN-75% and HHN-100%; TS1 input; p_s=20%; λ=1.5

Table 5.7: Makespan comparison of different number of replicas for TS1 traces in a 12-rack system

SH-job percentage p_s	Job arrival rate λ (/sec)	Number of replicas	HHN-75%	HHN-100%	Original Hadoop
5%	0.3	2	3140.4	3140.4	3139.7
		3	3140.2	3140.2	3140.1
	0.6	2	1810.4	1810.4	1809.8
		3	1810.2	1810.2	1810.0
	0.9	2	1403.5	1403.5	1397.0
		3	1399.3	1402.4	1397.1
20%	0.3	2	1132.3	1132.3	797.1
		3	1010.1	1010.1	796.8
	0.6	2	922.4	922.4	598.1
		3	803.1	803.1	599.2
	0.9	2	830.0	830.0	560.1
		3	701.2	699.5	560.0

times are reduced when using 3 replicas when compared to the 2-replica configuration, while regular-job response times are almost the same in all configurations.

Makespan In Section 5.3.4 we observed that the makespan in HHN-75% is higher than that in original Hadoop when the SH-job percentage is high (i.e., 20%). This is mainly due to the longer response time of a very large SH job in HHN than in original Hadoop. This occurs because in HHN, each SH job can use containers in only two racks when the number of replicas is 2. Since having 3 replicas helps reduce response times of SH jobs in a 12-rack system (see Fig. 5.8), we expect it to also reduce makespan in HHN.

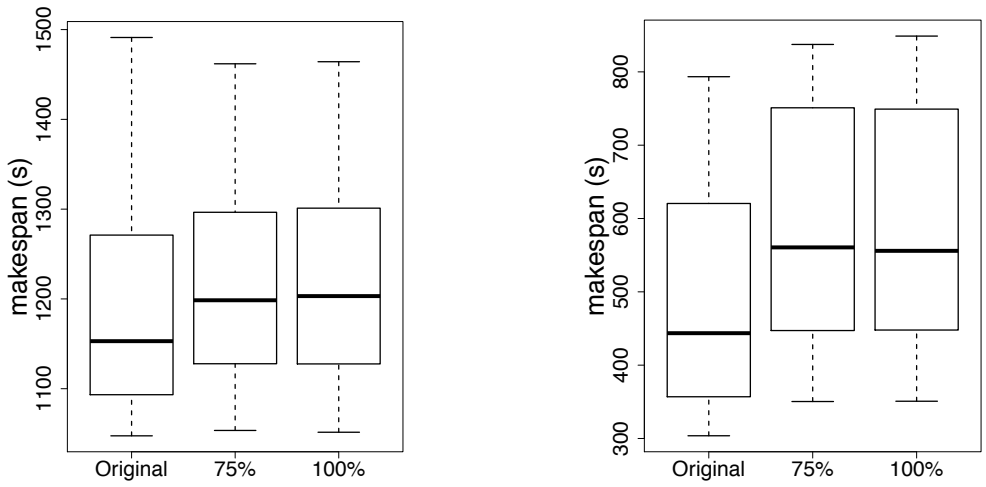
Table 5.7 compares the makespan of TS1 traces in HHN and original Hadoop under two settings for the number of replicas. When the SH-job percentage is small, i.e., 5%, the makespan in both HHN and original Hadoop is not affected by the number of replicas. In contrast, the makespan in both HHN-75% and HHN-100% is reduced when having more replicas when 20% of the jobs are SH, while the makespan in original Hadoop remains roughly the same for 2 and 3 replicas.

In summary, if the percentage of SH jobs is high, using a higher number of input-block replicas (e.g., 3) improves both job response time and makespan performance in HHN.

5.3.7 Multiple taces with the same trace parameters

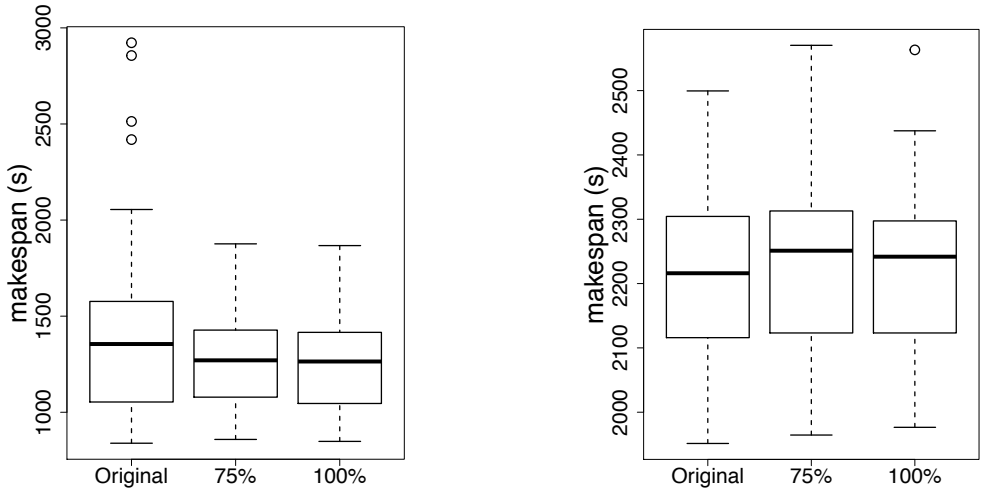
The simulation results presented previously were all obtained using a single trace for each trace-parameter combination (SH-job percentage and job arrival rate). To test whether our conclusions were independent of the specific traces used, we generated multiple traces, i.e., 30 traces, for each trace-parameter pair (see Section 5.3.2). All the traces used in this subsection are from TS2.

We first compare the makespan performance for various configurations. Fig. 5.9 shows boxplots of makespan for a 12-rack system at high load ($\lambda = 1.5$). Under both 5% and 20% SH-job scenarios (Fig. 5.9a and Fig. 5.9b, respectively), original Hadoop achieves shorter makespan than the two HHN configurations, but the difference is less significant when SH-job percentage is 5%. These observations are consistent with the ones made in Section 5.3.4. The longer makespan in HHN occurs because HHN limits SH jobs to run tasks on only a few racks even when there are idle containers in other racks. In addition, HHN-75% and HHN-100% have virtually the same performance.



(a) $p_s = 5\%$ (b) $p_s = 20\%$

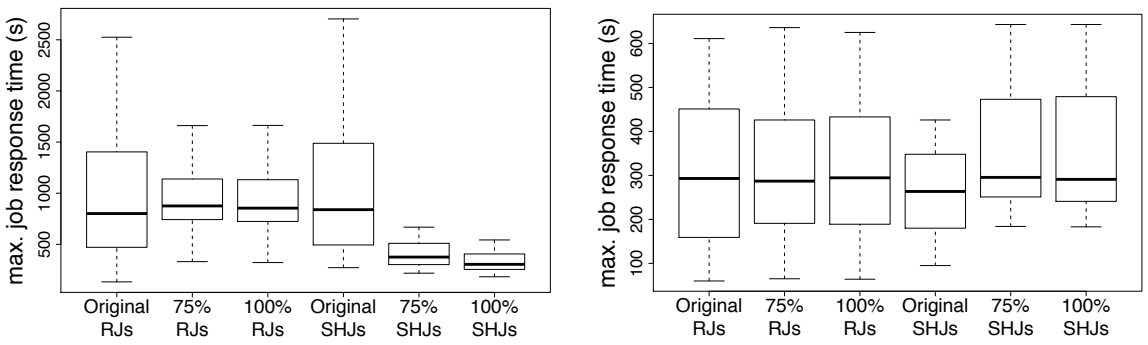
Figure 5.9: Makespan comparison in a 12-rack system with different SH-job percentages; Original: original Hadoop on EPS-only network; 75% and 100%: HHN-75% and HHN-100%; TS2 input; $\lambda = 1.5$



(a) $\lambda = 0.3$ (b) $\lambda = 0.1$

Figure 5.10: Makespan comparison in a 4-rack system at different loads; Original: original Hadoop on EPS-only network; 75% and 100%: HHN-75% and HHN-100%; TS2 input; $p_s = 20\%$

Fig. 5.10 shows the makespan performance in a 4-rack system. When the load is high ($\lambda = 0.3$), original Hadoop ends up with much longer makespans than HHN for some traces. This is because when the system is small, if there are several jobs with a large number of reduce tasks, most or even all of the containers could be allocated to those reduce tasks,

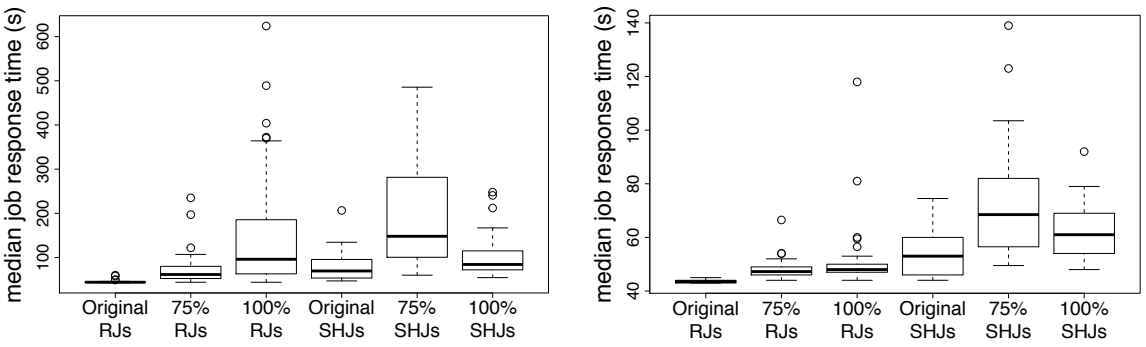


(a) 4-rack system, $\lambda = 0.3$ (b) 12-rack system, $\lambda = 1.5$

Figure 5.11: Maximum job response time comparison; Original: original Hadoop on EPS-only network; 75% and 100%: HHN-75% and HHN-100%; RJs: regular jobs, SHJs: shuffle-heavy jobs; TS2 input; $p_s = 20\%$

leaving insufficient containers for map tasks. The system could enter a deadlock, i.e., reduce tasks wait for all map tasks to complete before starting execution, while map tasks wait for reduce tasks to complete in order to obtain containers to run. This problem is handled in original Hadoop by preempting reduce containers, i.e., killing reduce tasks and allocating the freed containers to map tasks, which, however, leads to wasted CPU resources. The possible deadlock results in worse makespan performance for original Hadoop than HHN. When the job arrival rate is lower ($\lambda = 0.1$), original Hadoop on EPS-only network works slightly better than HHN.

Next, we consider job response times. Fig. 5.11 shows boxplots of the maximum response time of regular jobs and SH jobs in each of the 30 traces. For regular jobs in the 4-rack system, the distribution of maximum job response time is more spread for original Hadoop than HHN, which could be explained by the deadlock situation described above. Original Hadoop and HHN have similar distributions for maximum regular-job response time in the 12-rack system. For SH-jobs in the 4-rack system, the maximum job response time is smaller for HHN than for original Hadoop, since the priority given to SH-jobs allows these jobs to obtain containers faster than they deserve in a fair scheduler. In contrast, large SH-jobs finish slower in HHN than in original Hadoop in the 12-rack system. Although SH-job container requests are placed in per-rack queues, the map tasks of each SH-job are limited to use containers in only 2 racks. However, the worse performance of large SH jobs in HHN



(a) 4-rack system, $\lambda = 0.3$ (b) 12-rack system, $\lambda = 1.5$

Figure 5.12: Median job response time comparison; Original: original Hadoop on EPS-only network; 75% and 100%: HHN-75% and HHN-100%; RJs: regular jobs, SHJs: shuffle-heavy jobs; TS2 input; $p_s = 20\%$

would be largely improved by using more input-block replicas (see Section 5.3.6).

As maximum job response time indicates the performance of very large jobs, we use median job response time to capture the performance of medium-sized jobs. There are two interesting findings in the median job response times shown in Fig. 5.12. The first is that in both the 4-rack and 12-rack systems, regular jobs perform worse in HHN-100% than in HHN-75%. This occurs because when SH jobs are allowed to use all the racks in a system, regular jobs are likely to wait longer to obtain containers. The second finding is that with both system sizes, medium-sized SH jobs finish slower in HHN-75% than in original Hadoop and HHN-100%. This is because SH jobs need to wait in per-rack queues for service. The extra waiting time is longer in HHN-75% than in HHN-100%, and this waiting time is a larger portion of the response time for median-sized jobs than for very large jobs.

5.4 Conclusions

The work showed that it is feasible to modify certain data center applications so that the network traffic generated by these modified applications are better able to handle the high reconfiguration delays of Optical Circuit Switches (OCS) in hybrid electrical packet switch (EPS)/OCS networks. Specifically, this work proposed and evaluated a modified Hadoop designed for Hybrid Networks (HHN). Our evaluation results show that the HHN solution can achieve almost the same system-level performance metrics, makespan and CPU utilization,

and per-job performance metrics such as response time and fairness, as the original Hadoop running on an EPS-only network with the same high-rate links as in the optical subsystem of the hybrid network. Since these high-rate links are always-on in the EPS-only network, and require more expensive high-speed transceivers in the EPS-only network, power consumption and costs are higher for the EPS-only network when compared to the hybrid network.

Chapter 6

Conclusions and Future Work

We first summarize the work presented in this dissertation and draw three key conclusions, and then discuss potential future work to advance our current research.

6.1 Summary and Conclusions

In this dissertation, we presented our work on novel hybrid optical-/electrical-switched networks for energy-efficient operation. Our main contributions are as follows: (i) We proposed a new two-wavelength design for large-enterprise access links, which lowers power consumption and equipment costs without having a significant impact on performance; (ii) we proposed an alternative solution for sharing high-speed provider ports in the access-link design, i.e., AR mode with storage, and conducted comparative evaluations of AR mode and IR mode; (iii) we designed a hybrid DCN architecture named Optical Switch in the Middle (OSM), which offers increased flexibility for supporting multiple simultaneous high-speed ToR-to-ToR paths; and (iv) we proposed and evaluated Hadoop for Hybrid Networks (HHN), which achieves performance comparable to that of original Hadoop in EPS-only networks, while simultaneously achieving cost and power savings with hybrid networks.

Chapter 2 presented a reconfigurable two-wavelength large-enterprise access network design. In our design, the first wavelength is used as a lower-rate static optical circuit for general-purpose IP traffic, and the second wavelength is dynamically configured into a higher-rate circuit whenever needed for large dataset transfers. This design allows for the

sharing of a few high-rate provider IP-router ports among a larger number of enterprises. Our evaluation shows that only two and four shared high-rate provider-router ports are sufficient to support 20 customers at low loads and higher loads, respectively. At these values, power savings are more than 40 kW and equipment cost savings are in millions of dollars. The probability of start-time delay, which is the penalty paid by our dynamic design because of sharing, exceeding 20 minutes is kept below 1%. Therefore, our *conclusion* is that significant power and cost savings are possible with our proposed dynamic large-enterprise access-link design, while the start-time delay probability is kept low.

Chapter 3 proposed an alternative solution for sharing high-speed provider IP-router ports in our large-enterprise access-link design, i.e., AR mode. A storage server cluster is required if the sharing mode is AR. We conducted comparative evaluations of the IR-mode solution in the original design and the newly proposed AR-mode solution. Simulation results show that the AR-mode-plus-storage solution was able to improve data-transfer throughput by decreasing call blocking probability. The use of network storage of the AR-mode solution costs $102K more and consumes 3.2 kW more power than the IR-mode solution in an example configuration when the number of shared high-speed provider ports is four. Since the total cost and power savings achieved by the dynamic access link solution in IR mode are in millions of dollars and more than 20 kW, respectively, when compared to the conventional static solution, the dynamic access link solution in the AR-with-storage mode can still achieve significant cost and power savings. The *conclusion* is that the AR-with-storage solution can achieve better performance than the IR-mode solution in terms of blocking probability and average response time, while the AR solution costs more and consumes more power than the IR solution.

Chapter 4 presented a novel hybrid DCN architecture named Optical Switch in the Middle (OSM). By adding multiple simultaneous high-speed ToR-to-ToR paths through an OCS and an EPS at the core level, OSM offers increased flexibility when compared to prior hybrid DCN architectures. To effectively use the OSM architecture, we demonstrated the need for application modifications, and then proposed four modifications to Hadoop, and illustrated the potential of this architecture to achieve higher compute-resource utilization while simultaneously offering shorter job completion times.

Chapter 5 presented a comprehensive comparative evaluation of Hadoop for Hybrid Networks (HHN) on hybrid EPS/OCS networks and original Hadoop on conventional EPS-only networks. Cost and power analysis showed that hybrid architectures can save more than $500 K and consume 5 kW less power than the EPS-only architecture when considering example 100-rack DCNs. When the percentage of shuffle-heavy (SH) jobs is small, e.g., 5%, the HHN performance is the same as that of original Hadoop on an EPS-only network. When the percentage of SH jobs is large, e.g., 20%, the HHN performance is almost the same even at high loads, and even with a smaller number of input-block replicas, when we placed an upper bound on the per-job input-data size. Therefore, our *conclusion* is that it is feasible to achieve similar system-level and user-level performance with HHN, while simultaneously achieving power and cost savings with the hybrid network when compared to EPS-only networks.

6.2 Future Work

This work can be extended in the following directions:

1. The end-to-end path in the large-enterprise access-link design consists of both a circuit segment and an IP segment. This design can be extended to support end-to-end L1 circuits, which have zero packet retransmissions due to congestion losses, and thus are rate guaranteed and suitable for large dataset transfers.

2. For the DCN work, the proposed OSM architecture can be prototyped, and the HHN modifications can be implemented and evaluated on a real Hadoop cluster. Further theoretical models can be created and analyzed to generalize the conclusions made based on simulations, and to recommend Hadoop parameter settings for optimized performance.

Bibliography

[1] Lotfi Belkhir and Ahmed Elmeligi. Assessing ICT global emissions footprint: Trends to 2040 & recommendations. *Journal of Cleaner Production*, 177:448–463, 2018.

[2] How the global tech industry can shrink its electricity use. Telecom from the Economic Times. July, 2018. [Online]. Available: `https://telecom.economictimes.indiatimes.com/tele-talk/how-the-global-tech-industry-can-shrink-its-electricity-use/3158`.

[3] M Nishan Dharmaweera, Rajendran Parthiban, and Y Ahmet Şekercioğlu. Toward a power-efficient backbone network: The state of research. *IEEE Communications Surveys & Tutorials*, 17(1):198–227, 2015.

[4] BT labs delivers ultra-efficient terabit superchannel. June, 2018. [Online]. Available: `http://home.bt.com/tech-gadgets/future-tech/bt-labs-delivers-ultra-efficient-terabit-superchannel-11364187351803`.

[5] Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. Helios: A hybrid electrical/optical switch architecture for modular data centers. *ACM SIGCOMM Computer Comm. Review*, 41(4):339–350, 2011.

[6] X. Wang, M. Veeraraghavan, M. Brandt-Pearce, T. Miyazaki, N. Yamanaka, S. Okamoto, and I. Popescu. A dynamic network design for high-speed enterprise access links. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'15)*, pages 1–7, Dec 2015.

[7] Ward Van Heddeghem, Filip Idzikowski, Willem Vereecken, Didier Colle, Mario Pickavet, and Piet Demeester. Power consumption modeling in optical multilayer networks. *Photonic Network Communications*, 24(2):86–102, 2012.

[8] Babak Behzad, Huong Vu Thanh Luu, Joseph Huchette, Surendra Byna, Prabhat, Ruth Aydt, Quincey Koziol, and Marc Snir. Taming parallel I/O complexity with auto-tuning. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC'13, pages 68:1–68:12, New York, NY, USA, 2013. ACM.

[9] ESnet-graphite. https://graphite.es.net/.

[10] George Porter, Richard Strong, Nathan Farrington, Alex Forencich, Pang Chen-Sun, Tajana Rosing, Yeshaiahu Fainman, George Papen, and Amin Vahdat. Integrating microsecond circuit switching into the data center. In *Proc. of the ACM SIGCOMM*, volume 40, pages 447–458. ACM, 2013.

[11] Guohui Wang, David G Andersen, Michael Kaminsky, Konstantina Papagiannaki, TS Ng, Michael Kozuch, and Michael Ryan. c-Through: Part-time optics in data centers. In *Proc. of ACM SIGCOMM '10*, volume 40, pages 327–338, 2010.

[12] Kai Chen, Ankit Singla, Atul Singh, Kishore Ramachandran, Lei Xu, Yueping Zhang, Xitao Wen, and Yan Chen. OSA: An optical switching architecture for data center networks with unprecedented flexibility. *IEEE/ACM Transactions on Networking*, 22(2):498–511, 2014.

[13] He Liu, Feng Lu, Alex Forencich, Rishi Kapoor, Malveeka Tewari, Geoffrey M. Voelker, George Papen, Alex C. Snoeren, and George Porter. Circuit switching under the radar with REACToR. In *11th USENIX NSDI*, pages 1–15, Seattle, WA, April 2014. USENIX Association.

[14] Hamid Hajabdolali Bazzaz, Malveeka Tewari, Guohui Wang, George Porter, TS Ng, David G Andersen, Michael Kaminsky, Michael A Kozuch, and Amin Vahdat. Switching the optical divide: Fundamental challenges for hybrid electrical/optical datacenter networks. In *Proceedings of the 2nd ACM Symposium on Cloud Computing*, page 30. ACM, 2011.

[15] X. Wang, X. Lin, W. Sun, and M. Veeraraghavan. Comparison of two sharing modes for a proposed optical enterprise-access SDN architecture. In *2018 IEEE International Telecomunication Networks and Applications Conference (ITNAC)*, pages 1–7, Nov 2018.

[16] Xiaoyu Wang, Malathi Veeraraghavan, Zongli Lin, and Eiji Oki. Optical Switch in the Middle (OSM) architecture for DCNs with Hadoop adaptations. In *Proceedings of the IEEE Conference on Communications (ICC'17)*, May 2017.

[17] X. Wang and M. Veeraraghavan. An evaluation study of a proposed hadoop for hybrid networks (hhn). In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'17)*, pages 1–7, Dec 2017.

[18] X. Wang, M. Veeraraghavan, and H. Shen. Evaluation study of a proposed Hadoop for data center networks incorporating optical circuit switches. *IEEE/OSA Journal of Optical Communications and Networking*, 10(8):50–63, August 2018.

[19] S. Maji, X. Wang, M. Veeraraghavan, J. Ros-Giralt, and A. Commike. A pragmatic approach of determining heavy-hitter traffic thresholds. In *2018 European Conference on Networks and Communications (EuCNC)*, pages 1–9, June 2018.

[20] A. M. Hendawi, F. Alali, X. Wang, Y. Guan, T. Zhou, X. Liu, N. Basit, and J. A. Stankovic. Hobbits: Hadoop and Hive based Internet traffic analysis. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 2590–2599, Dec 2016.

[21] I. Popescu, T. Miyazaki, M. Chino, X. Wang, S. Okamoto, A. Gravey, P. Gravey, M. Veeraraghavan, M. Brandt-Pearce, and N. Yamanaka. Application-centric energy-efficient Ethernet with quality of service support. *Electronics Letters*, 51(15):1165–1167, 2015.

[22] T. Miyazaki, I. Popescuy, M. Chino, X. Wang, K. Ashizawa, S. Okamotoz, M. Veeraraghavan, and N. Yamanaka. High speed 100GE adaptive link rate switching for

energy consumption reduction. In *2015 International Conference on Optical Network Design and Modeling (ONDM)*, pages 227–232, May 2015.

[23] Xiao Lin, Xiaoyu Wang, Malathi Veeraraghavan, Weiqiang Sun, and Weisheng Hu. Design of a time-space decoupled scheduling method for inter-DC optical networks. In *Submission of the IEEE Conference on Communications (ICC'19)*, 2019.

[24] Daniel Schien, Vlad C Coroama, Lorenz M Hilty, and Chris Preist. The energy intensity of the Internet: edge and core networks. In *ICT Innovations for Sustainability*, pages 157–170. Springer, 2015.

[25] Chankyun Lee, Junhyuk Kim, Yoontae Kim, and J.K.K. Rhee. Adaptive resource provisioning using traffic forecasting for energy efficient networks. In *Proceedings of International Conference on Information and Communication Technology Convergence (ICTC)*, pages 463–464, Nov. 2010.

[26] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J.A. Maestro. Ieee 802.3az: the road to energy efficient ethernet. *Communications Magazine, IEEE*, 48(11):50–56, November 2010.

[27] P. Reviriego, J.A. Hernandez, D. Larrabeiti, and J.A. Maestro. Performance evaluation of energy efficient Ethernet. *IEEE Communications Letters*, 13(9):697–699, Sept 2009.

[28] Chamara Gunaratne, Ken Christensen, and Stephen W Suen. Ethernet adaptive link rate (ALR): analysis of a buffer threshold policy. In *Proceedings of Global Telecommunications Conference (GLOBECOM'06).*, pages 1–6. IEEE, 2006.

[29] Ward Van Heddeghem and F. Idzikowski. Equipment power consumption in optical multilayer networks - source data. Technical report IBCN-12-001-01 (Jan. 2012). [Online]. Available: `http://powerlib.intec.ugent.be`.

[30] Ralf Huelsermann, Matthias Gunkel, Clara Meusburger, and Dominic A Schupke. Cost modeling and evaluation of capital expenditures in optical multilayer networks. *Journal of Optical Networking*, 7(9):814–833, 2008.

[31] F. Rambach, B. Konrad, L. Dembeck, U. Gebhard, M. Gunkel, M. Quagliotti, L. Serra, and V. Lopez. A multilayer cost model for metro/core networks. *IEEE/OSA Journal of Optical Communications and Networking*, 5(3):210–225, Mar. 2013.

[32] Eli Dart, Lauren Rotman, Brian Tierney, Mary Hester, and Jason Zurawski. The science DMZ: A network design pattern for data-intensive science. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, pages 85:1–85:10, New York, NY, USA, 2013. ACM.

[33] S. Gringeri, B. Basch, V. Shukla, R. Egorov, and T.J. Xia. Flexible architectures for optical transport nodes and networks. *Communications Magazine, IEEE*, 48(7):40–50, July 2010.

[34] Arnold O. Allen. *Probability, Statistics, and Queueing Theory with Computer Science Applications.* Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[35] RD Stevens and MC Sinclair. Finite-source analysis of traffic on private mobile radio systems. *Electronics letters*, 33(15):1292–1293, 1997.

[36] Xiaoyu Wang, Malathi Veeraraghavan, Takahiro Miyazaki, Naoaki Yamanaka, Satoru Okamoto, and Ion Popescu. Equipment power consumption and cost for enterprise access links - source data. Technical report (Apr. 2015). [Online]. Available: `http://venividiwiki.ee.virginia.edu/mediawiki/index.php/ACTION#Technical_reports`.

[37] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jon Zolla, Urs Hölzle, Stephen Stuart, and Amin Vahdat. B4: Experience with a globally-deployed software defined WAN. In *Proceedings of the ACM SIGCOMM 2013 Conference*, SIGCOMM '13, pages 3–14, New York, NY, USA, 2013. ACM.

[38] Arista 7280SRAM-48C6, price obtained through a conversation with Arista. https://goo.gl/LxQuSY.

[39] Cisco QSFP-100G-SR4-S QSFP28 850nm 100m transceiver module price. https://www.fs.com/products/48354.html.

[40] The IBM Power8 review: Challenging the Intel Xeon. https://goo.gl/EBYZwN.

[41] IBM Power systems S812L and S822L technical overview and introduction. https://www.redbooks.ibm.com/redpapers/pdfs/redp5098.pdf.

[42] Mellanox MCX516A-CDAT price from Mellanox official store. https://goo.gl/ePxS5E.

[43] Broadcom HBA 9405W series x16 PCIe tri-mode storage adapter price. https://goo.gl/y9y5oK.

[44] PAC Storage GS 3000 with dual redundant raid controllers. https://goo.gl/ty6gDz, price obtained by a quotation from PAC.

[45] PAC storage host board power consumption. Power consumption value obtained from communications with PAC.

[46] Dell 3.5" NL-SAS 1TB HDD price. https://goo.gl/akWnTp.

[47] Supermicro SuperStorage 2028R-NR48N specifications and price. https://goo.gl/1hi5Ai, https://goo.gl/hkUHt5.

[48] Supermicro SuperServer 2028U-TN24R4T+ specifications and price. https://goo.gl/6X2oht, https://goo.gl/KbD5uT.

[49] Supermicro AOC-S100G-M2C retail price. https://goo.gl/EJGeVo.

[50] Samsung 960 PRO Series NVMe SSD prices. https://goo.gl/ceCwg8.

[51] Nikhef 100G storage box. https://goo.gl/2L4atM.

[52] ESnet 100G data transfer nodes. https://goo.gl/zsKe6H.

[53] W. Lu and Z. Zhu. Malleable reservation based bulk-data transfer to recycle spectrum fragments in elastic optical networks. *Journal of Lightwave Technology*, 33(10):2078–2086, May 2015.

[54] Xin Jin, Yiran Li, Da Wei, Siming Li, Jie Gao, Lei Xu, Guangzhi Li, Wei Xu, and Jennifer Rexford. Optimizing bulk transfers with software-defined optical WAN. In *Proceedings of the 2016 ACM SIGCOMM Conference*, SIGCOMM '16, pages 87–100, New York, NY, USA, 2016. ACM.

[55] Payman Samadi, Ke Wen, Junjie Xu, and Keren Bergman. Software-defined optical network for metro-scale geographically distributed data centers. *Opt. Express*, 24(11):12310–12320, May 2016.

[56] A. Patel, M. Tacca, and J. P. Jue. Time-shift circuit switching. In *Proceedings of the 2008 Conference on Optical Fiber Communication/National Fiber Optic Engineers (OFC/NFOEC)*, pages 1–3, Feb 2008.

[57] N. Laoutaris, G. Smaragdakis, R. Stanojevic, P. Rodriguez, and R. Sundaram. Delay-tolerant bulk data transfers on the Internet. *IEEE/ACM Transactions on Networking*, 21(6):1852–1865, Dec 2013.

[58] Nikolaos Laoutaris, Michael Sirivianos, Xiaoyuan Yang, and Pablo Rodriguez. Inter-datacenter bulk transfers with Netstitcher. In *Proceedings of the ACM SIGCOMM 2011 Conference*, pages 74–85, New York, NY, USA.

[59] Y. Wu, Z. Zhang, C. Wu, C. Guo, Z. Li, and F. C. M. Lau. Orchestrating bulk data transfers across geo-distributed datacenters. *IEEE Transactions on Cloud Computing*, 5(1):112–125, Jan 2017.

[60] X. Lin, W. Sun, M. Veeraraghavan, and W. Hu. Time-shifted multilayer graph: A routing framework for bulk data transfer in optical circuit-switched networks with assistive storage. *IEEE/OSA J. Opt. Commun. Netw.*, 8(3):162–174, March 2016.

[61] X. Lin, W. Sun, M. Veeraraghavan, and W. Hu. Slotted store-and-forward optical circuit-switched networks: A performance study. *IEEE/OSA J. Opt. Commun. Netw.*, July 2017.

[62] Navid Hamedazimi, Zafar Qazi, Himanshu Gupta, Vyas Sekar, Samir R. Das, Jon P. Longtin, Himanshu Shah, and Ashish Tanwer. Firefly: A reconfigurable wireless data center fabric using free-space optics. In *Proc. of ACM SIGCOMM'14*, pages 319–330, New York, NY, 2014. ACM.

[63] Yong Cui, Shihan Xiao, Xin Wang, Zhenjie Yang, Shenghui Yan, Chao Zhu, Xiang-Yang Li, and Ning Ge. Diamond: Nesting the data center network with wireless rings in 3-D space. *IEEE/ACM Transactions on Networking*, 2017.

[64] Chaoli Zhang, Fan Wu, Xiaofeng Gao, and Guihai Chen. Free talk in the air: A hierarchical topology for 60 GHz wireless data center networks. *IEEE/ACM Transactions on Networking*, 2017.

[65] Monia Ghobadi, Ratul Mahajan, Amar Phanishayee, Nikhil Devanur, Janardhan Kulkarni, Gireeja Ranade, Pierre-Alexandre Blanche, Houman Rastegarfar,

Madeleine Glick, and Daniel Kilper. ProjecToR: Agile reconfigurable data center interconnect. In *Proc. of ACM SIGCOMM'16*, pages 216–229, New York, NY, 2016. ACM.

[66] Akira Yamashita, Wataru Muro, Masayuki Hirono, Takehiro Sato, Satoru Okamoto, Naoaki Yamanaka, and Malathi Veeraraghavan. Hadoop triggered opt/electrical data-center orchestration architecture for reducing power consumption. In *2017 19th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE, 2017.

[67] Matei Zaharia, Dhruba Borthakur, Joydeep Sen Sarma, Khaled Elmeleegy, Scott Shenker, and Ion Stoica. Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling. In *Proceedings of ACM EuroSys '10*, pages 265–278, New York, NY, 2010.

[68] Faraz Ahmad, Srimat T. Chakradhar, Anand Raghunathan, and T. N. Vijaykumar. Shufflewatcher: Shuffle-aware scheduling in multi-tenant MapReduce clusters. In *Proc. of USENIX ATC'14*, pages 1–13, Philadelphia, 2014. USENIX Association.

[69] Virajith Jalaparti, Peter Bodik, Ishai Menache, Sriram Rao, Konstantin Makarychev, and Matthew Caesar. Network-aware scheduling for data-parallel jobs: Plan when you can. In *Proc. of ACM SIGCOMM '15*, pages 407–420, New York, 2015. ACM.

[70] Jihe Wang, Danghui Wang, Meng Zhang, Meikang Qiu, and Bing Guo. Similarity-based node distance exploring and locality-aware shuffle optimization for Hadoop MapReduce. In *2017 IEEE International Conference on Smart Cloud (SmartCloud)*, pages 103–108. IEEE, 2017.

[71] Tom White. *Hadoop: The Definitive Guide*. O'Reilly Media, 2009.

[72] On-Demand Secure Circuits and Advance Reservation System (OSCARS). http://www.es.net/OSCARS/docs/index.html.

[73] X. Zhu and M. Veeraraghavan. Analysis and design of book-ahead bandwidth-sharing mechanisms. *IEEE Transactions on Communications*, 56(12):2156–2165, December 2008.

[74] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, kc claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. Named data networking. *SIGCOMM CCR*, 44(3):66–73, July 2014.

[75] Mark Berman, Jeffrey S. Chase, Lawrence Landweber, Akihiro Nakao, Max Ott, Dipankar Raychaudhuri, Robert Ricci, and Ivan Seskar. GENI: A federated testbed for innovative network experiments. *Computer Networks*, 61:5–23, 2014.

[76] Hadoop TeraSort application. `https://hadoop.apache.org/docs/r2.7.1/api/org/apache/hadoop/examples/terasort/package-summary.html`.

[77] Ganesh Ananthanarayanan, Sameer Agarwal, Srikanth Kandula, Albert Greenberg, Ion Stoica, Duke Harlan, and Ed Harris. Scarlett: Coping with skewed content popularity in MapReduce clusters. In *Proceedings of the sixth conference on Computer systems*, pages 287–300. ACM, 2011.

[78] SWIM workload repository. `https://github.com/SWIMProjectUCB/SWIM/wiki/Workloads-repository`.

[79] Calient S series optical circuit switch. https://goo.gl/hc4DWj, price obtained through a conversation with a Calient salesperson.

[80] Glimmerglass Intelligent Optical System 600. https://goo.gl/n1dByn.

[81] Arista 7160 series and Arista 7280R series comparisons. https://goo.gl/kZyeJE, https://goo.gl/LxQuSY, prices obtained through a conversation with Arista.

[82] Cisco Nexus 7700 switches environment data sheet. https://goo.gl/Qpmqeq.

[83] Juniper QFX10002 Ethernet switch data sheet. https://goo.gl/igaMvv.

[84] Huawei CloudEngine 12800 series data center switches. https://goo.gl/555Ffs.

[85] Arista 100G optics and cabling Q&A document. https://goo.gl/EpkTV3, price obtained through a conversation with Arista.

[86] Cisco 100GBASE QSFP-100G modules data sheet. https://goo.gl/SPKH1r.

[87] Fiber optic patch cables from fs.com. https://goo.gl/VCeMvt.

[88] Rivanna: UVA HPC system. `http://arcs.virginia.edu/rivanna`.