



ELSEVIER

Contents lists available at ScienceDirect

## Computer Networks

journal homepage: [www.elsevier.com/locate/comnet](http://www.elsevier.com/locate/comnet)

# Provisioning high-availability datacenter networks for full bandwidth communication

Wenda Ni <sup>a,\*</sup>, Changcheng Huang <sup>a</sup>, Jing Wu <sup>b</sup><sup>a</sup> Department of Systems and Computer Engineering, Carleton University, Ottawa, Ontario K1S 5B6, Canada<sup>b</sup> Communications Research Centre Canada, Ottawa, Ontario K2H 8S2, Canada

## ARTICLE INFO

## Article history:

Received 23 May 2013

Received in revised form 30 October 2013

Accepted 4 December 2013

Available online 12 March 2014

## Keywords:

Datacenter networks

Valiant load balancing

Full bandwidth communication

Service availability

Link failures

Capacity allocation

## ABSTRACT

One critical challenge in datacenter network design is full bandwidth communication. Recent advances have enabled this communication paradigm based on the notion of Valiant load balancing (VLB). In this paper, we target full bandwidth communication among all servers, for all valid traffic patterns, and under  $k$  arbitrary link failures. We focus on two typical datacenter topologies, VL2 and fat-tree, and propose a mechanism to perform VLB on fat-tree. We develop the minimum link capacity required on both topologies, where edge and core links are handled separately. These results can help datacenter providers to provision their networks with guaranteed availability. Based on the results, we evaluate the minimum total link capacity required on each topology and characterize the capacity increase trend with  $k$  and with the total number of supported servers. These studies are important for datacenter providers to project their capital expenditures on datacenter design, upgrade, and expansion. Next, we compare the total link capacity between the two topologies. We find that given the same server scale, fat-tree requires less total capacity than does VL2 for small  $k$ . For large  $k$ , there exists a turning point at which VL2 becomes more capacity-efficient.

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

Existing and emerging Internet applications, such as web search, video streaming, and social networking, are migrating towards the cloud computing paradigm, where user applications are run over a common datacenter infrastructure that consists of tens to hundreds of thousands of servers. In this new context, each user application job (e.g., MapReduce) is partitioned and assigned to various servers, far beyond the number a single server rack can hold. To enable local computation, extensive data exchanges are performed among servers that reside separately within a datacenter, contributing to a huge amount

of communication traffic. A recent survey conducted by IDC [1] regarding the challenges of cloud services has shown that performance and availability are among the top concerns expressed by potential cloud users. Accordingly, service availability is listed as a leading obstacle to the growth of cloud computing [2]. In this paper, we consider how to provision a high-availability communication system within a datacenter, which is an important factor in determining service availability.

Communication among servers is supported by a datacenter network, which typically consists of multiple tiers of switches and/or routers. Conventional datacenter networks have a tree-like topology designed using the *scale-up* method [3,4]. Higher-end non-commodity switches and/or routers with higher port speeds are required at higher tiers of the hierarchy to accommodate higher amounts of aggregate traffic. Ideally, the port speed

\* Corresponding author. Tel.: +1 613 790 6788.

E-mail addresses: [wendani@sce.carleton.ca](mailto:wendani@sce.carleton.ca) (W. Ni), [huang@sce.carleton.ca](mailto:huang@sce.carleton.ca) (C. Huang), [jingwu@ieee.org](mailto:jingwu@ieee.org) (J. Wu).

Notations		
$k$	Number of link failures	communication among all servers under $k$ arbitrary link failures
$\mathcal{N}_E$	Set of edge switches	Number of servers supported by one ToR switch on VL2
$\mathcal{N}_A$	Set of aggregation switches	$n_s$
$\mathcal{N}_C$	Set of core switches	$m$
$\mathcal{L}_E$	Set of edge links	$\delta(j)$
$\mathcal{L}_C$	Set of core links	$n$
$\mathcal{T}$	Set of all valid traffic matrices	$\tilde{n}_s \equiv n/2$
$\Lambda = \{\lambda_{ii'}\}_{ \mathcal{N}_E  \times  \mathcal{N}_E }$	Traffic matrix	$\mathcal{P}$
$\lambda_{ii'}$	Traffic demand from edge switch $i$ to edge switch $i' \neq i$	$\mathcal{N}_E^p$
$r$	Maximum sending/receiving rate of a server NIC	$\mathcal{N}_A^p$
$c_l(k)$	Minimum capacity required on link $l$ to enable full bandwidth communication among all servers under $k$ arbitrary link failures	$\mathcal{N}_C^j$
$C(k)$	Minimum total link capacity required on a network to enable full bandwidth	$\mathcal{F}_k$

moving up the hierarchy should be scaled up to the point that any server can communicate with any other server at the maximum rate of its network interface card (NIC). This situation is generally referred to as full bandwidth communication [4], which enables arbitrary traffic patterns among all of the servers as long as each server sends and receives traffic within the capacity limit of its NIC. To this end, the oversubscription ratios of all switches in the network should be maintained at 1:1, where a switch's oversubscription ratio is defined as the ratio of the maximum total traffic load on its downlinks under full bandwidth communication to the total capacity of its uplinks. Unfortunately, while it may be technically possible to provide full bandwidth communication using the scale-up method, the cost of such a communication network is prohibitively high due to the deployment of high-price non-commodity switches and/or routers at higher tiers to deliver the higher port speed required by traffic aggregation. Consequently, conventional datacenter networks are constructed with a significant oversubscription ratio, which is typically 5:1 to 20:1 at the lowest switch tier [3,5] and increases rapidly at higher tiers, reaching 240:1 at the highest switch tier [3]. This large oversubscription ratio fragments the server pool, meaning that under certain traffic patterns, a server can reach only part of its access limit due to the existence of bandwidth bottlenecks at higher levels of the switching hierarchy. Bandwidth bottlenecks can potentially lead to reduced server utilization, which in turn limits the performance and scale of cloud applications.

To address the oversubscription problem in a cost-efficient manner, novel datacenter network infrastructures have been proposed. Typical designs include VL2 [3], fat-tree [4], and BCube [6]. These new designs all use the *scale-out* approach, which takes full advantage of economies of scale by leveraging a large number of inexpensive commodity switches, in contrast to the *scale-up* method, which uses a small number of high-price non-commodity

switches. In VL2 and fat-tree, the equivalent of a high-speed port of non-commodity switches is built by bundling low-speed ports from several commodity switches to match the capacity. The use of high-end switches is thereby avoided completely. In BCube, the use of non-commodity switches is eliminated by introducing switching functionality into the servers, which further enables datacenter networks to scale out using low-end commodity switches with a small port count. Although a large number of commodity switches are required in the scale-out approach, the cost barrier associated with scaling the speed of a single port still renders the cost of scale-out solutions significantly lower than the scale-up counterpart [4,6]. Link capacity or, equivalently, port speed is dimensioned to enable full bandwidth communication among all of the servers (when there are no failures). Due to the topology scale and connectivity density, network failures are a part of daily life within a datacenter [5,7]. The rich connectivity inherent in the scale-out method provides multiple paths between any server pair, allowing resiliency against network failures [3,8] and allowing Valiant load balancing (VLB) to handle highly variable traffic without creating any "hot-spot" links [3,9]. However, despite the availability of redundant paths, capacity provisioning in current practice (e.g., [3,4,8]) does not support full bandwidth communication in the presence of network failures. In other words, any network failure can cause network congestion, which is manifested at the service level as increased service latency or, more severely, service unavailability [10].

Motivated by the fact that service availability has long been one of the top challenges in moving applications to the cloud [1,2], in this paper, we investigate how network failures can be tolerated with minimal impact on service availability. One option is to incorporate redundancy at the application level [5,11] based on the notion of virtual machine (VM) replication [12]. Specifically, a backup VM is allocated for each working VM. When a working VM

becomes slow or unresponsive as a result of network failures, job tasks continue seamlessly on the corresponding backup. To this end, two VMs must be running on physical servers residing in different failure domains [10] (availability zones [13]) so that they are not affected by the same network failures inside a datacenter. Additionally, VM replication must be performed at very high frequencies, e.g., every 25 ms [12], to ensure that the backup VM maintains a completely up-to-date copy of the corresponding working VM. While VM replication is effective in protecting against server failures, it is not effective in protecting against network link failures. When network links fail, a large number of physical servers may be simultaneously affected. These affected servers are thus underutilized due to the reduced traffic.

Therefore, the fundamental solution is to introduce redundancy at the underlying network level, particularly by allocating sufficient capacity to links. In this paper, we address the capacity allocation problem in datacenter networks. As a first step, we consider link failures, which are the predominant form of failures that occur approximately an order of magnitude more often than device failures in datacenter networks [7]. In this context, our design goal is to *provide full bandwidth communication among all the servers, for all valid traffic patterns, and under  $k$  arbitrary link failures* so that link failures can be masked at the network level with little or no impact on service level availability. Note that by “valid”, we mean traffic patterns that are compliant with the sending/receiving capacity limit on each server NIC. We focus on two typical datacenter network topologies, VL2 and fat-tree, for the following reasons. (1) Both topologies follow the switch-centric design principle, where servers only act as computation elements, which is consistent with conventional datacenter network design and thus enables a smooth upgrade path for existing datacenter networks. The counterpart approach is server-centric, where servers, apart from their original role of computing, act as switches/routers by relaying packets for each other. As traditional servers are not designed for fast packet switching/forwarding, specialized hardware and software are required on the server side. Such issues are yet to be solved and hence prevent server-centric topologies (e.g., BCube [6]) from real-life implementation. (2) In sharp contrast to server-centric topologies, both VL2 (folded Clos topology in the network core) and fat-tree are strongly promoted by leading vendors, such as Cisco [14], and are in the mature stage for commercial deployment. For simplicity of description, we only consider link failures in this paper. However, it should be noted that our method can be generalized to the case of node failures.

To cope with the highly dynamic traffic that characterizes datacenter environments [3,5,15–17], we employ VLB, a two-phase routing scheme capable of handling traffic variations in a congestion-free manner without the need for dynamic path adjustments. Specifically, in the first phase, a packet originating from a node is sent to a randomly chosen intermediate node, regardless of the packet destination. In the second phase, a node forwards packets received from the other nodes to their destinations. Consequently, the traffic load between any source-destination pair is equally split over all available paths [18,19]. In this

sense, two-phase routing is equivalent to load-balanced multipathing, which has recently been enabled at layer 2 (with limitation on path multiplicity) using industry standard protocols (e.g., IETF TRILL) and proprietary vendor alternatives (e.g., Cisco’s FabricPath) to meet the unique requirements of emerging switch-centric datacenter networks (e.g., VL2 and fat-tree) [20]. More importantly, to achieve the goal of congestion-free communication for all valid traffic matrices, the link capacity required under VLB is shown to be significantly lower than under traditional direct source-destination routing [18]. This significant capacity advantage encouraged us to choose VLB over its direct routing counterpart. When link failures occur, traffic carried by the failed paths is evenly assigned to the remaining paths to the same destinations. This protection mechanism is fundamentally different from conventional path protection in that there is no clear demarcation between working and backup paths. All paths between a source-destination pair are mutually protected in a global manner. To make sure that all remaining paths have enough capacity to carry the extra load, the links on these paths need to be engineered with enough spare capacity. Accordingly, in this paper, we answer two important questions:

1. For each topology, how much link capacity is needed at minimum to support full bandwidth communication for arbitrary valid traffic patterns among all the servers under  $k$  arbitrary link failures?
2. Given datacenters with the same server scale and failure tolerance level, which of the two topologies performs better in terms of the total link capacity required?

Our contributions can be summarized as follows.

- We derive the minimum link capacity required on VL2 to support full bandwidth communication for arbitrary valid traffic patterns among all servers. This derivation provides the basis to dimension VL2 networks for the desired availability. Due to the topology constraint, we are limited to considering  $k$  arbitrary link failures that do not partition the topology.
- We propose a mechanism to perform VLB on fat-tree. Based on this mechanism, we develop the minimum link capacity requirement on fat-tree to guarantee full bandwidth communication under  $k$  arbitrary link failures for arbitrary valid traffic patterns among all the servers. The results constitute the foundation for provisioning fat-tree networks with an appropriate guarantee of availability.
- We calculate the minimum total link capacity required on VL2 and fat-tree and characterize the trend in capacity increase with  $k$  and with the total number of supported servers on each topology. These studies provide insights for capital cost estimation in designing new datacenters as well as in upgrading and expanding existing datacenters.
- Given the same total number of supported servers, we compare the minimum total link capacity required on

VL2 and fat-tree and provide insightful observations. In particular, we characterize the results for  $1 \leq k \leq \frac{n}{4}$ , where  $n$  denotes the port count of homogeneous switches used to build fat-tree.

The remainder of the paper is organized as follows. In Section 2, we discuss the previous work on capacity allocation. In Section 3, we present background material on network topologies, traffic model, routing structure, and link capacity requirement with no failures. In particular, we propose a mechanism to carry out VLB over fat-tree in Section 3.2.3. Based on the foundation in Section 3, we develop our major contributions in Sections 4–6. Specifically, in Sections 4 and 5, we derive the required capacity to tolerate  $k$  link failures for VL2 and fat-tree, respectively. The numerical results for both topologies are presented in Section 6. We conclude the paper in Section 7. Part of the results in this paper were summarized in [29].

## 2. Previous work

The capacity allocation problem under link failures has been studied extensively for backbone networks (such as ATM, MPLS, and WDM). The design goal is typically to tolerate a predefined set of link failures for a given set of static traffic demands. Because the failure probabilities of links are very low in such a network context, tolerating single link failures is generally sufficient to guarantee service availability [21]. Therefore, the vast majority of works (e.g., [22]) consider single link failures. A small body of works (e.g., [23,24]) go up to dual link failures to accommodate connection requests with very high availability requirements [25]. The tolerance level in dual failure case can be partial [23] or full [24], depending on the specific availability target. In general, each connection takes the form of a working path protected by one or two backup paths. All paths are mutually link-disjoint, with each of them found through direct source-destination single-path routing. The primary backup path carries no traffic unless the working path fails, and the secondary backup path, if provisioned, is not used unless both the working and primary backup paths fail. Clearly, these works cannot be applied to our context, which, in contrast, is characterized by highly dynamic traffic at all timescales (and thus cannot be represented trivially by a deterministic, static traffic matrix), multiple concurrent link failures on a daily basis [7] (and thus considering only single or dual link failures is insufficient to guarantee service availability), and more importantly, VLB (a special form of multipath routing with unique protection features as discussed above). An exception to the aforementioned works is [26], which addresses capacity allocation for dynamic traffic demands. Link capacity is provisioned in such a way that all traffic patterns satisfying the nodal input/output capacity constraints can be accommodated in a strict-sense nonblocking manner. The traffic assumption is similar to ours. However, traffic is still routed directly from source to destination over a single path, and only single link failures are tolerated through the use of a conventional link protection scheme. These three factors prevent that work from being

applicable in our case. The most relevant work in backbone network design is [18], which introduced VLB (in parallel to [19]) to handle traffic volatility in a capacity-efficient manner yet only through an entirely static network configuration. The link capacity required to guarantee 100% throughput to any valid traffic matrix is investigated, which turns out to be significantly lower (provably the lowest when there are no network failures) than in traditional static networks with direct single-path routing. The link capacity in the presence of link failures is also derived. However, all of the results are developed over a full mesh logical topology and thus do not hold on the datacenter topologies under study (or on any topology other than full mesh). In the context of datacenter networks, a design algorithm was proposed in [17] that removes all topological restrictions placed on traditional practices. The network performance in terms of bisection bandwidth and end-to-end latency is thereby largely improved, but full bandwidth communication is not achieved. Moreover, because the design focus is on the topology aspect to enable arbitrary topology output, the algorithm is not optimized for link capacity and it does not consider network failures.

Therefore, to the best of our knowledge, there is no known work that can address the capacity allocation problem in datacenter networks, which exhibit unique traffic and failure characteristics, and thus dictate unique solutions in routing and protection.

## 3. Network models and capacity allocation under no-failure scenario

Both VL2 and fat-tree consist of three layers of switches: the edge layer, aggregation layer, and core layer. The switches at the corresponding layers are referred to as edge switches, aggregation switches, and core switches, respectively. Let  $\mathcal{N}_E$ ,  $\mathcal{N}_A$ , and  $\mathcal{N}_C$  denote the sets of edge switches, aggregation switches, and core switches, numbered from 1 to  $|\mathcal{N}_E|$ , from 1 to  $|\mathcal{N}_A|$ , and from 1 to  $|\mathcal{N}_C|$ , respectively. We define the links between the edge and the aggregation layers as edge links and define the links between the aggregation and the core layers as core links. Let  $\mathcal{L}_E$  and  $\mathcal{L}_C$  denote the sets of edge links and core links, respectively.

The server NICs are assumed to be homogeneous. Let  $r$  denote the capacity limit of each NIC. That is, each server sends and receives traffic within the maximum rate  $r$ . All servers are connected to the network via edge switches, each to one and only one edge switch (i.e., single homing). Thus, all traffic enters or leaves the network at edge switches. We represent the network traffic by matrix  $\Lambda = \{\lambda_{ii'}\}_{|\mathcal{N}_E| \times |\mathcal{N}_E|}$ , where element  $\lambda_{ii'}$  ( $i \neq i'$ ) denotes the traffic demand from edge switch  $i$  to edge switch  $i'$ . Element  $\lambda_{ii}$  denotes the traffic among servers hosted by the same edge switch  $i$ . This part of the traffic is fully handled by local edge switches and does not go through the network. Hence, we refer to such traffic as local traffic. Local traffic does not consume bandwidth on edge or core links and thus is not considered in the matrix or in our study. In other words, we let  $\lambda_{ii} \equiv 0$ . Due to the high variation of traffic, the traffic matrix is constantly changing at both

large and small timescales [3,5,15–17]. The set of all valid traffic matrices is denoted by  $\mathcal{T}$ , where the condition for a traffic matrix to be valid takes different forms on different topologies, as observed below.

### 3.1. VL2

#### 3.1.1. Topology

Fig. 1 shows the VL2 topology proposed in [3]. The interconnection between the aggregation and the core layers forms a complete bipartite graph, with each aggregation switch connected to each core switch by one port. Therefore, each aggregation switch requires a large number of ports, which is a major limitation of the VL2 topology. If we construct both layers using  $m$ -port switches,<sup>1</sup> a total of  $m$  switches are deployed at the aggregation layer, as each core switch has  $m$  ports and can thus accommodate at most  $m$  aggregation switches. Each aggregation switch uses half of its ports to connect to the core layer, leading to a total of  $m/2$  switches at the core layer. The other  $m/2$  ports of the aggregation switches are connected to the edge layer, which uses the top of rack (ToR) switches as edge switches. Each ToR switch has two links connected to the aggregation layer. In particular, two links are connected to two different aggregation switches for redundancy. This redundancy level, however, is not satisfactory in the sense that any edge switch can be disconnected from the topology by multiple ( $k \geq 2$ ) link failures. Consequently, it undermines our goal of full bandwidth communication under  $k$  arbitrary link failures for any  $k \geq 2$ . We can improve the redundancy level by increasing the number of uplinks from each edge switch at the cost of consuming more ports of the aggregation switches, which are, however, constrained by the complete bipartite connectivity with the core layer; this in turn limits the total number of edge switches and servers that can be supported. In other words, there exists a natural tradeoff between the uplink redundancy of edge switches and the scale of servers. In this study, we choose to retain the original topology in [3] and compromise our design goal, which is defined in Section 4. The VL2 topology has  $m^2/2$  edge links and  $m^2/2$  core links, i.e.,  $|\mathcal{L}_E| = |\mathcal{L}_C| = m^2/2$ .

#### 3.1.2. Traffic model

Let  $n_s$  denote the number of servers supported by one edge/ToR switch on VL2. The ingress/egress capacity of each edge switch is thus bounded by  $n_s r$ . By “ingress/egress”, we mean traffic that goes through the network, not including local traffic, which bounces off the edge switches. Any valid traffic matrix  $\Lambda \in \mathcal{T}$  satisfies the following constraints:

$$\sum_{i' \in \mathcal{N}_E, i' \neq i} \lambda_{ii'} \leq n_s r, \quad \forall i \in \mathcal{N}_E, \quad (1)$$

$$\sum_{i \in \mathcal{N}_E, i \neq i'} \lambda_{ii'} \leq n_s r, \quad \forall i' \in \mathcal{N}_E. \quad (2)$$

<sup>1</sup> Supporting the same total number of servers typically requires VL2 and fat-tree to use switches of different port counts. Hence, to facilitate the total link capacity comparison between them, we denote the switch port counts on VL2 and fat-tree by different notations, i.e.,  $m$  and  $n$ , respectively.

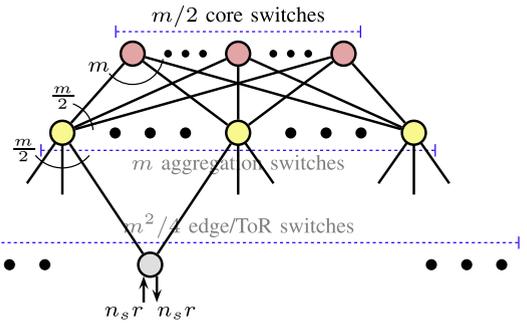


Fig. 1. VL2 topology.

#### 3.1.3. VLB

The VL2 topology provides two two-hop paths between an edge switch and a core switch. Ingress traffic from edge switch  $i$  to edge switch  $i'$  is first sent to a randomly chosen core switch over a path chosen at random (i.e., one of two paths) [3]. The core switch then forwards the traffic to destination  $i'$  over a randomly chosen path (one of two paths) [3]. All traffic is forwarded on a per-packet basis.

The above process can be equivalently viewed as follows. In the first phase, traffic from  $i$  to  $i'$  is evenly split over  $m$  two-hop paths that go to  $m/2$  core switches. In the second phase, traffic is forwarded from all core switches to destination  $i'$  over  $m$  two-hop paths with equal split. We can see that the routing in the two phases is symmetric, which can be better unraveled on the unfolded VL2 topology shown in Fig. 2.

#### 3.1.4. Link capacity with no failures

Because the VLB approach described above distributes traffic from an edge switch to another edge switch equally over all alternative paths, we can easily calculate the load on each edge and core link based on the total traffic originating and terminating at each edge switch given in (1) and (2), respectively. Specifically, each edge switch has two links connected to the aggregation layer. Thus, each edge link  $(i, j)$  carries half of the traffic sent from edge switch  $i$  to any other edge switch  $i'$ . It follows that the maximum load on link  $(i, j)$  is given by  $\max \left\{ \sum_{i' \in \mathcal{N}_E, i' \neq i} \frac{\lambda_{ii'}}{2} \right\} = \frac{n_s r}{2}$ , where we apply (1). The reverse direction of link  $(i, j)$ , i.e., link  $(j, i)$ , carries the traffic that terminates at edge switch  $i$ . Due to the routing symmetry, we know immediately that the maximum load on link  $(j, i)$  is  $\max \left\{ \sum_{i' \in \mathcal{N}_E, i' \neq i} \frac{\lambda_{i'i}}{2} \right\} = \frac{n_s r}{2}$  by applying (2). Therefore, the minimum capacity required on any edge link  $l$  under the no-failure scenario (i.e.,  $k = 0$ ) is given by

$$c_l(0) = \frac{n_s r}{2}, \quad \forall l \in \mathcal{L}_E. \quad (3)$$

Let  $\delta(j)$  denote the set of edge switches that connect to aggregation switch  $j$ . We have  $|\delta(j)| = m/2, \forall j \in \mathcal{N}_A$ . For each edge switch  $i \in \delta(j)$ , each core link  $(j, u)$  is on one and only one of the  $m$  paths from  $i$  to the core layer. Thus, the link load for traffic from  $i$  is given by  $\frac{1}{m} \sum_{i' \in \mathcal{N}_E, i' \neq i} \lambda_{ii'}$ , with a maximum value of  $\frac{n_s r}{m}$  considering (1). It immediately follows that the maximum load on link  $(j, u)$  is

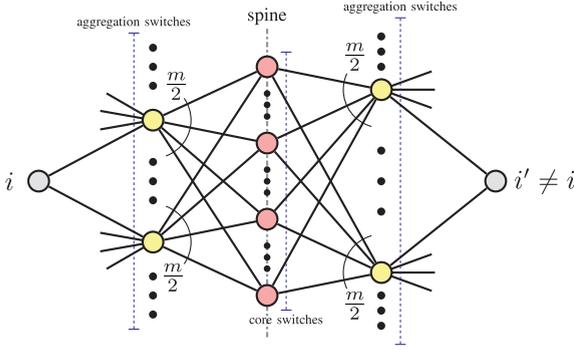


Fig. 2. Unfolded VL2 topology.

$\sum_{i \in \delta(j)} \frac{n_s r}{m} = |\delta(j)| \frac{n_s r}{m} = \frac{n_s r}{2}$ . The reverse direction of link  $(j, u)$  carries the traffic that terminates at edge switches in set  $\delta(j)$ . Due to the symmetry of the two routing phases, we know immediately that the maximum load on link  $(u, j)$  is  $\frac{n_s r}{2}$ . Therefore, the minimum capacity required on any core link  $l$  for  $k = 0$  is given by

$$c_l(0) = \frac{n_s r}{2}, \quad \forall l \in \mathcal{L}_C. \quad (4)$$

In (3) and (4), if we set  $n_s = 20$  and  $r = 1$  Gb/s, we have  $c_l(0) = 10$  Gb/s,  $\forall l \in \mathcal{L}_E \cup \mathcal{L}_C$ , which is consistent with the link capacity given in [3]. In other words, the minimum link capacity for  $k = 0$  in (3) and (4) generalizes the special design case in [3].

### 3.2. Fat-tree

#### 3.2.1. Topology

The fat-tree topology was originally proposed in [27] to interconnect the processors of a parallel supercomputer. Unlike the traditional tree topology, links in a fat-tree become “fatter” in terms of capacity as they move towards the root. To take advantage of economies of scale, rather than using one high-end switch with high-capacity links, “fat” links are constructed instead by the bundling of “thin” links from multiple commodity switches, leading to a multi-rooted tree topology. The edge and aggregation switches are arranged in the form of switching modules called pods, which are interconnected by core switches representing the multiple roots of a fat-tree. An illustrative topology with  $n = 4$  is shown in Fig. 3.

In general, if  $n$ -port switches are used to construct a fat-tree, each pod consists of  $n/2$  edge switches and  $n/2$  aggregation switches. Within a pod, each edge switch is connected to each aggregation switch by one port, thereby forming a complete bipartite graph. Externally, each pod is connected to each core switch by one of the other half  $(n/2)^2$  ports of its aggregation switches. Consequently, a total of  $(n/2)^2$  switches are required at the core layer. Conversely, because each core switch has  $n$  ports, a total of  $n$  pods are supported. Additionally, to make the topology regular, the interconnection between pods and core switches should satisfy the following condition: there exists a partition of the core switches into  $n/2$  equal-sized mutually disjoint sets such that each aggregation switch

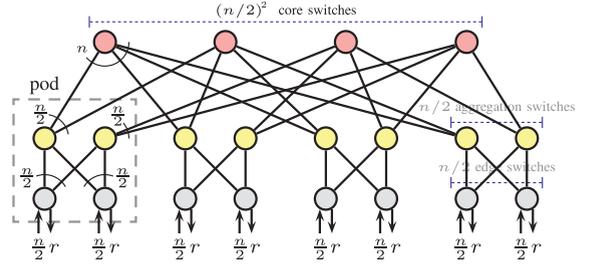


Fig. 3. Fat-tree topology with  $n = 4$ .

of each pod is connected to core switches belonging to the same set. In the case of  $n = 4$ , which is shown in Fig. 3, the core switches are partitioned into two sets, where the left two form one set and the right two form the other set. Each aggregation switch in each pod is connected to core switches in one and only one set. For each aggregation switch  $j$  ( $\forall j \in \mathcal{N}_A$ ), we denote the core switches to which it connects by set  $\mathcal{N}_C^j$ . Generally, each pod has  $(n/2)^2$  edge links and  $(n/2)^2$  core links. Thus, we have  $|\mathcal{L}_E| = |\mathcal{L}_C| = n^3/4$ . In contrast to the VL2 topology, the aggregation and the core layers in fat-tree do not form a complete bipartite graph, which allows the aggregation switches to have more ports connected to edge switches and therefore more redundancy on edge links.

#### 3.2.2. Traffic model

Let  $\tilde{n}_s$  denote the number of servers supported by one edge switch on fat-tree. Each edge switch has  $n/2$  ports connected to  $n/2$  servers. Thus, we have  $\tilde{n}_s \equiv n/2$ . Accordingly, the ingress/egress capacity limit of each edge switch is  $\tilde{n}_s r = \frac{n}{2} r$ , recalling that ingress/egress traffic refers to traffic that goes through the network, i.e., traffic that originates and terminates at different edge switches. Any valid traffic matrix  $\Lambda \in \mathcal{T}$  satisfies the following constraints:

$$\sum_{i' \in \mathcal{N}_E, i' \neq i} \lambda_{i i'} \leq \tilde{n}_s r = \frac{n}{2} r, \quad \forall i \in \mathcal{N}_E, \quad (5)$$

$$\sum_{i \in \mathcal{N}_E, i \neq i'} \lambda_{i i'} \leq \tilde{n}_s r = \frac{n}{2} r, \quad \forall i' \in \mathcal{N}_E. \quad (6)$$

#### 3.2.3. VLB

Each edge switch can communicate with another edge switch in the same pod or any edge switch in a remote pod. We refer to the traffic among edge switches of the same pod as intra-pod traffic and to the traffic destined for an edge switch in a remote pod as inter-pod traffic. To the best of our knowledge, how VLB is performed on fat-tree has not been discussed in the existing literature. We propose the following mechanisms for intra- and inter-pod traffic.

As the core layer connects a pod to a remote pod, intra-pod traffic does not go outside the pod. Specifically, a packet from an edge switch is first sent to a randomly chosen aggregation switch inside the pod. The aggregation switch then forwards the packet to the destination edge switch residing in the same pod. In both phases, packets are

delivered through direct links. From the end-to-end viewpoint, intra-pod traffic is evenly distributed over  $n/2$  two-hop paths between any two edge switches.

In contrast, inter-pod traffic goes through the core layer outside an individual pod. The core switches take the role of intermediate nodes in the two-phase routing. Routing in the two phases is symmetric, as can be easily observed by focusing on the actual routing topology from edge switch  $i$  to edge switch  $i'$  shown in Fig. 4. Because each core switch has only one two-hop path towards each edge switch, we can virtually concatenate the two paths in two phases for each core switch. Then, from the end-to-end viewpoint, inter-pod traffic is evenly split over  $(n/2)^2$  four-hop paths between any two edge switches.

As observed above, intra- and inter-pod traffic use different switching layers as intermediate destinations, which leads to entirely different path structures for these two types of traffic.

### 3.2.4. Link capacity with no failures

Below, we calculate the link load considering the contributions of intra- and inter-pod traffic, respectively. Because VLB requires traffic to be equally distributed among all alternative paths, the link load can be calculated based on the numbers of alternative paths discussed above. It should be noted that the edge and core links are different in the sense that edge links carry both intra- and inter-pod traffic, whereas core links carry only inter-pod traffic. We will discuss the contributions of intra- and inter-pod traffic individually. Let  $\mathcal{P}$  denote the set of pods, which are numbered from 1 to  $n/2$ . The edge and aggregation switches in pod  $p$  are denoted by sets  $\mathcal{N}_E^p$  and  $\mathcal{N}_A^p$ , respectively. We have  $\mathcal{N}_E^p \in \mathcal{N}_E$ ,  $\mathcal{N}_A^p \in \mathcal{N}_A$ , and  $|\mathcal{N}_E^p| = |\mathcal{N}_A^p| = \frac{n}{2}$ ,  $\forall p \in \mathcal{P}$ . Assume that edge switches  $i$  and  $i' \neq i$  and aggregation switch  $j$  are in pod  $p$ , and edge switch  $i'$  is in pod  $p' \neq p$ , as illustrated in Fig. 5.

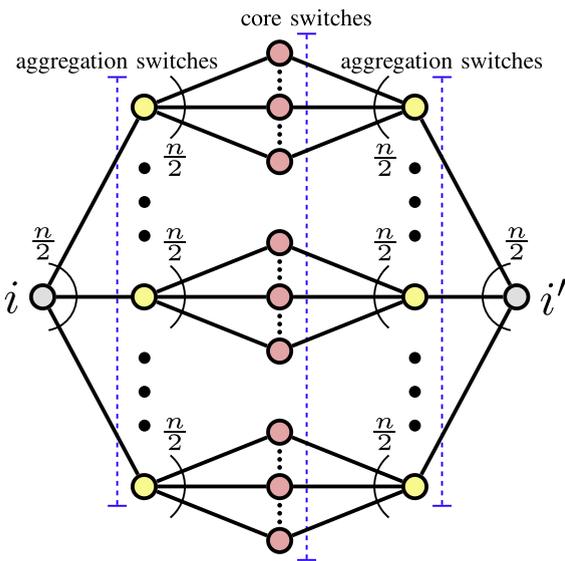


Fig. 4. Inter-pod routing topology between edge switch  $i$  and remote edge switch  $i'$  on fat-tree.

Each edge link  $(i, j)$  carries traffic sent from  $i$ . It is on one and only one of the  $n/2$  paths carrying intra-pod traffic from  $i$  to  $i'$ . Thus, the link load for traffic  $\lambda_{ii'}$  is  $\frac{\lambda_{ii'}}{n/2}$ . The inter-pod traffic from  $i$  to  $i'$  is split over  $(n/2)^2$  paths. Among them,  $n/2$  paths traverse link  $(i, j)$ . Thus, the link load for traffic  $\lambda_{ii'}$  is  $\frac{n/2}{(n/2)^2} \lambda_{ii'} = \frac{\lambda_{ii'}}{n/2}$ , which takes the same form as the one introduced by intra-pod traffic. Hence, we can say that link  $(i, j)$  carries  $\frac{1}{n/2}$  the amount of traffic from edge switch  $i$  to any other edge switch  $t$ . Consequently, the maximum load on link  $(i, j)$  is  $\max \left\{ \frac{1}{n/2} \sum_{t \in \mathcal{N}_E, t \neq i} \lambda_{it} \right\} = r$ , which follows from (5). The reverse direction of link  $(i, j)$  carries traffic that terminates at  $i$ . From the routing symmetry, we know immediately that the maximum load on link  $(j, i)$  is also  $r$ . Therefore, the minimum capacity requirement on any edge link  $l$  under the no-failure scenario (i.e.,  $k = 0$ ) is given by

$$c_l(0) = r, \quad \forall l \in \mathcal{L}_E. \quad (7)$$

Each core link  $(j, u)$  carries inter-pod traffic from each edge switch in pod  $p$ . Specifically, link  $(j, u)$  is on one and only one of the  $(n/2)^2$  paths from any edge switch  $i$  to any remote edge switch  $i'$ . Thus, the total load on link  $(j, u)$  is the sum over  $i$  and  $i'$  and is expressed as  $\sum_{i \in \mathcal{N}_E^p} \sum_{i' \in \mathcal{N}_E^{p'} \setminus \mathcal{N}_E^p} \frac{\lambda_{ii'}}{(n/2)^2}$ . Because the inter-pod traffic from each edge switch is bounded by  $\frac{n}{2}r$ , following from (5), i.e.,  $\sum_{i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p} \lambda_{ii'} \leq \frac{n}{2}r$ ,  $\forall i \in \mathcal{N}_E^p$ , the maximum load on link  $(j, u)$  is given by  $|\mathcal{N}_E^p| \frac{\frac{n}{2}r}{(n/2)^2} = r$ . The reverse direction of link  $(j, u)$  carries traffic that terminates at any edge switch in pod  $p$ . From the symmetry of the two routing phases, we know immediately that the maximum load on link  $(u, j)$  is  $r$ . Therefore, the minimum capacity requirement on any core link  $l$  for  $k = 0$  is given by

$$c_l(0) = r, \quad \forall l \in \mathcal{L}_C. \quad (8)$$

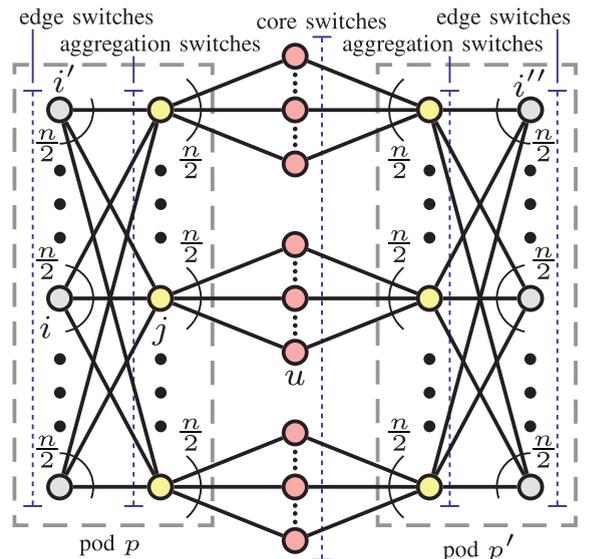


Fig. 5. Unfolding pods  $p$  and  $p'$  on fat-tree.

The minimum required capacity for  $k = 0$  on the edge and core links coincides with the original design in [4], where VLB is not employed. The fact that the edge and core links require the same capacity indicates that our VLB approach provides fair bandwidth allocation, and the network can be built using switches with homogeneous ports.

#### 4. Capacity allocation on VL2 with $k$ link failures

In this section, we derive the link capacity requirement on a VL2 topology that experiences  $k$  arbitrary link failures. The goal is to guarantee full bandwidth communication among all the servers. Given  $k$ , the basis to this end is that the topology remains connected in any  $k$  link failures. For a VL2 topology above the aggregation layer, it requires at least  $\frac{m}{2}$  link failures to disconnect an aggregation switch from any other aggregation switch, which allows us to consider  $k$  up to  $\frac{m}{2} - 1$ . However, each edge switch has only two links connecting it to the aggregation layer. Thus, any number of multiple link failures can disconnect any edge switch, which can limit the fault tolerance capability of the VL2 topology. Because the topology above the aggregation layer is guaranteed to be connected for  $k \leq \frac{m}{2} - 1$ , we consider  $k$  up to  $\frac{m}{2} - 1$ . For  $k \geq 2$ , we focus on failure scenarios where all of the edge switches remain connected, i.e., two edge links incident to an edge switch do not fail concurrently. From the probability viewpoint, this case holds when the edge links are engineered with high availability, such as 0.99999 (“five nines”), so that the probability that two edge links fail concurrently becomes negligible (e.g.,  $10^{-10}$  in the case of five-nine availability per edge link under independent link failures). Increasing the number of edge links per edge switch would certainly be helpful; however, this approach is constrained by the number of ports consumed by core switches, as discussed earlier.

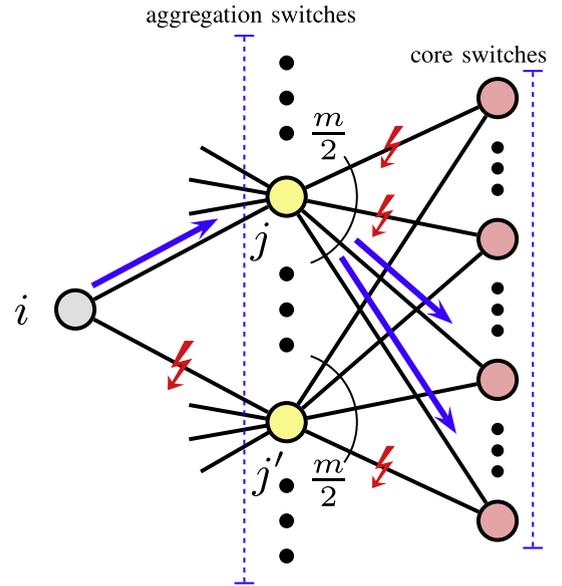
##### 4.1. Edge links

Due to the limited connectivity from each edge switch, the capacity requirement for edge links can be derived straightforwardly. We establish the following theorem.

**Theorem 1.** *Let  $k$  be a given integer with value  $1 \leq k \leq \frac{m}{2} - 1$ . To guarantee full bandwidth communication among all of the servers under  $k$  arbitrary link failures that do not partition the topology, the minimum capacity required on each edge link is*

$$c_l(k) = n_s r, \quad \forall l \in \mathcal{L}_E. \quad (9)$$

**Proof.** As the minimum link capacity is equal to the maximum link load, we find the maximum load experienced on an edge link. We first consider the link load in the direction from the edge layer to the aggregation layer, which is caused by traffic in the first routing phase or, equivalently, by traffic going from the edge layer to the core layer. Consider  $i$  to be a general edge switch. Let  $j$  and  $j'$  denote the two aggregation switches through which edge switch  $i$  is connected to the aggregation layer. Without loss of generality, we study the traffic load on edge link  $(i, j)$ . Link  $(i, j)$



**Fig. 6.** Failure on link  $(i, j')$ , which leads to the maximum traffic load on edge link  $(i, j)$ .

carries only traffic originating from  $i$ , which under the no-failure scenario is sent over either link  $(i, j)$  or link  $(i, j')$  in the first hop. Between these two links, at most one of them can fail under any failure scenario considered because otherwise edge switch  $i$  is disconnected from all other switches in the topology. In the case that edge link  $(i, j')$  fails, as illustrated in Fig. 6, link  $(i, j)$  carries all traffic originating from  $i$ . This situation leads to a maximum load of  $n_s r$  on link  $(i, j)$  when the traffic from  $i$  reaches the ingress capacity limit given by (1). Because in this case all traffic from  $i$  goes through link  $(i, j)$  in the first hop, it is clearly the maximum load that can be experienced on link  $(i, j)$  over all failure scenarios and all valid traffic matrices.

The reverse direction of link  $(i, j)$  carries traffic that terminates at  $i$ . From the routing symmetry, we know immediately that the maximum load on link  $(j, i)$  is also  $n_s r$ . Considering all link failure scenarios, the capacity allocation on all edge links is uniform, with a minimum value of  $n_s r$ .  $\square$

##### 4.2. Core links

In this subsection, we derive the capacity requirement for the core links. Similar to the edge link case, we obtain the minimum capacity on a core link by finding its maximum traffic load. Because the maximum link load is taken over all valid traffic matrices and all failure scenarios considered, one naive approach is to enumerate all possible failure scenarios and develop the load expression under each of them. Unfortunately, the number of possible scenarios, although finite, is intractable. This prohibits the use of the enumeration method. Instead, we reduce the search space by showing that the maximum load on a core link can be achieved within a limited set of failure scenarios. In doing so, we prune out the majority of failure sce-

narios and limit the failure scenarios of interest to a tractable size, over which the maximum link load can be found more easily. Below, we first characterize the failure scenarios that can potentially lead to the maximum load on a core link for any valid traffic pattern and then formulate the maximum link load over these candidate scenarios.

Let  $l$  be a general core link incident to aggregation switch  $j$  ( $\forall j \in \mathcal{N}_A$ ) and core switch  $u$  ( $\forall u \in \mathcal{N}_C$ ). Because the two directions of a core link carry traffic in different routing phases, we also refer to link  $l$  by the ordered pair  $(j, u)$  or  $(u, j)$  to distinguish its two directions. Specifically, link  $(j, u)$  is directed from aggregation switch  $j$  to core switch  $u$  and thus carries traffic in the first routing phase, which goes from the source edge switches to the core layer. Accordingly, the reverse direction of  $(j, u)$ , i.e., link  $(u, j)$ , carries traffic in the second routing phase, which goes from the core layer to the destination edge switches.

Because the symmetry of the two routing phases dictates the maximum load in both directions of any link to be the same, it is sufficient to consider only one direction of a core link. Specifically, we characterize the failure scenarios that can lead to the maximum load on link  $(j, u)$ . Intuitively, failures on edge links should be at locations where the affected traffic is directed towards aggregation switch  $j$  and is thus in part carried over link  $(j, u)$ . This indicates the failed edge links to be incident to edge switches that are connected to  $j$  via the other edge links. Additionally, failures on core links should be at locations where the affected traffic goes through link  $(j, u)$  as much as possible. This state can intuitively be achieved when all failed core links are incident to  $j$ . We establish [Propositions 1 and 2](#), which formally consider the failure settings on edge and core links, respectively. Based on these two propositions, we identify in [Theorem 2](#) the failure scenarios among which the maximum load on core link  $(j, u)$  can be found. Without loss of generality, we assume that among the  $k$  failed links,  $k_e$  failures are on edge links, and  $k_c$  failures are on core links. We have  $k_e, k_c \in \mathcal{K}(k)$  and  $k_e + k_c = k$ , where the set  $\mathcal{K}(k)$  is defined as

$$\mathcal{K}(k) \triangleq \{a \in \mathbb{Z} | 0 \leq a \leq k\}. \quad (10)$$

Note that from the routing symmetry, it immediately follows that both propositions hold for the reverse direction of link  $(j, u)$ , i.e., link  $(u, j)$ .

**Proposition 1.** *Given  $k_e$  and  $k_c$  failures on edge and core links, respectively, where  $k_e \geq 1$  and the failure locations on core links are chosen arbitrarily, for any valid traffic matrix, the maximum load on link  $(j, u)$  can be found over edge failures characterized by the following setting: all failed edge links are incident to edge switches that connect to aggregation switch  $j$  via the other (operating) edge links.*

**Proof.** See Appendix A.  $\square$

[Proposition 1](#) allows us to assume without loss of generality for our purpose (i.e., finding the maximum load on link  $(j, u)$ ) that all failed edge links follow the setting described above, i.e., are incident to edge switches that

connect to aggregation switch  $j$  via the other (operating) edge links. We denote such edge switches by set  $\gamma(j)$ . As  $|\gamma(j)| = k_e \leq k < \frac{m}{2} = |\delta(j)|$ , we have  $\gamma(j) \subset \delta(j)$ .

In the above failure setting for the edge links, we then characterize the failure locations on the core links. We establish the following proposition.

**Proposition 2.** *Given  $k_e$  and  $k_c$  failures on edge and core links, respectively, where  $k_c \geq 1$  and the failures on edge links follow the setting described in [Proposition 1](#), for any valid traffic matrix, the load on link  $(j, u)$  with all failed core links incident to aggregation switch  $j$  is no smaller than with any of the core link failures not incident to  $j$ .*

**Proof.** See Appendix B.  $\square$

Combining [Propositions 1 and 2](#), we establish the following theorem.

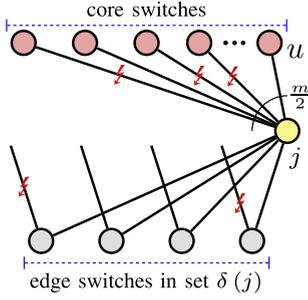
**Theorem 2.** *Let  $k$  be a given integer with value  $1 \leq k \leq \frac{m}{2} - 1$ . Let  $k_e$  and  $k_c$  be integer variables that take values in set  $\mathcal{K}(k)$  and satisfy  $k_e + k_c = k$ . The variables  $k_e$  and  $k_c$  denote the number of failures on edge and core links, respectively. Given any valid traffic matrix, the maximum load on link  $(j, u)$  over  $k$  arbitrary link failures can be found among the failure scenarios characterized by the following two features:*

1. if  $k_e \geq 1$ , all  $k_e$  failed edge links are incident to edge switches that are connected to aggregation switch  $j$  via the other (operating) edge links; and
2. if  $k_c \geq 1$ , all  $k_c$  failed core links are incident to aggregation switch  $j$ .

**Proof.** Given the number of failures on edge and core links, i.e., given the values of  $k_e$  and  $k_c$ , respectively, where  $k_e + k_c = k$ , [Propositions 1 and 2](#) ensure that for any valid traffic matrix, the maximum load on link  $(j, u)$  can be found among the failure scenarios where the edge and core link failures follow settings (1) and (2), respectively. Then, by considering all possible values of  $k_c$  (or equivalently,  $k_e$ , as  $k_c + k_e = k$ , where  $k$  is a constant), we actually cover the entire search space of  $k$  link failure scenarios, where the maximum link load over  $k$  arbitrary link failures can be found.  $\square$

We illustrate the failure scenarios characterized by [Theorem 2](#) in [Fig. 7](#) and denote them by set  $\mathcal{S}_F^{j,k}$ . Note that set  $\mathcal{S}_F^{j,k}$  is associated with aggregation switch  $j$  and  $k$  link failures. As we will see later, given  $k_c$  and  $k$ , the maximum load on link  $(j, u)$  is agnostic of more specific failure details, such as the exact failed edge or core links. Thus, we are only interested in failure scenarios with different values of  $k_c$ . Accordingly, we represent each element  $f$  in set  $\mathcal{S}_F^{j,k}$  by a 2-tuple  $(k_c, k)$  and do not differentiate among different combinations of failed links under a given  $(k_c, k)$ .

**Theorem 3.** *Let  $k$  be a given integer with value  $1 \leq k \leq \frac{m}{2} - 1$ . To guarantee full bandwidth communication among all of the servers under  $k$  arbitrary link failures that do not partition the topology, the minimum capacity required on any core link  $l$  can be formulated as*



**Fig. 7.** Characteristics of a failure scenario in set  $\mathcal{S}_F^{j,k}$ , where the maximum load on link  $(j, u)$  can be experienced.

$$c_l(k) = n_s r \cdot \max_{k_c \in \mathcal{K}(k)} f(k_c, k), \quad \forall l \in \mathcal{L}_C, \quad (11)$$

where  $k_c$  is an integer variable denoting the number of failed core links, with its permissible value given by the set

$$\mathcal{K}(k) = \{k_c \in \mathbb{Z} | 0 \leq k_c \leq k\}, \quad (12)$$

and function  $f(k_c, k)$  is defined as

$$f(k_c, k) \triangleq \frac{k - k_c}{\frac{m}{2} - k_c} + \frac{\frac{m}{2} - k + k_c}{m - k_c}. \quad (13)$$

**Proof.** We formulate the minimum capacity on any core link by formulating the maximum load on it, which by default is taken over all valid traffic matrices and all failure scenarios considered. For any core link  $(j, u)$  directed from aggregation switch  $j$  ( $\forall j \in \mathcal{N}_A$ ) to core switch  $u$  ( $\forall u \in \mathcal{N}_C$ ), Theorem 2 enables us to limit the search space of failure scenarios to set  $\mathcal{S}_F^{j,k}$ , where the maximum load on link  $(j, u)$  can be found. Under these scenarios, we derive the traffic load expression on link  $(j, u)$  with the ultimate goal of formulating its maximum load.

To model the traffic load under any failure scenario  $f$  in set  $\mathcal{S}_F^{j,k}$ , we first consider the traffic load on link  $(j, u)$  when there are no failures, and then add the load increase on link  $(j, u)$  incurred under  $f$ . Basically, link  $(j, u)$  carries one and only one path from each edge switch in set  $\delta(j)$ . These paths remain operating under any failure scenario in set  $\mathcal{S}_F^{j,k}$ . As discussed in Section 3.1.4, when there are no failures, the traffic load on link  $(j, u)$  is  $\sum_{i \in \delta(j)} \frac{1}{m} \sum_{i' \in \mathcal{N}_E, i' \neq i} \lambda_{i'i}$ . In the presence of any failure scenario in set  $\mathcal{S}_F^{j,k}$ , the load increase on link  $(j, u)$  is caused by traffic originating from all edge switches in set  $\delta(j)$ . Based on the number of disrupted paths from an edge switch, edge switches can be classified into two sets:  $\gamma(j)$  and  $\delta(j) \setminus \gamma(j)$ . As we will see next, for edge switches in the same set, their contributions to the load increase can be expressed identically, whereas edge switches in different sets have different expressions and thus need to be addressed separately.

Each edge switch in set  $\gamma(j)$  is on one end of a failed edge link, as shown in Fig. 7. Under a failure scenario with  $k_c$  failed core links, this situation leads to a total of  $\frac{m}{2} + k_c$  disrupted paths from each edge switch in  $\gamma(j)$ . Specifically, for each  $i$  in  $\gamma(j)$ , a failed edge link incident to  $i$  disrupts half of the paths (i.e.,  $\frac{m}{2}$  paths) from  $i$ , whereas each failed core

link disrupts one path from  $i$ . Because the traffic carried by each path from  $i$  is uniformly  $\frac{1}{m} \sum_{i' \in \mathcal{N}_E, i' \neq i} \lambda_{i'i}$  before failure, the total amount of disrupted traffic originating from  $i$  is  $\frac{\frac{m}{2} + k_c}{m} \sum_{i' \in \mathcal{N}_E, i' \neq i} \lambda_{i'i}$ ,  $\forall i \in \gamma(j)$ . The disrupted traffic is evenly distributed to the remaining  $\frac{m}{2} - k_c$  operating paths from  $i$ . Because link  $(j, u)$  is on one and only one remaining operating path from  $i$ , it carries  $\frac{1}{\frac{m}{2} - k_c}$  the amount of disrupted traffic from each  $i$  in  $\gamma(j)$  after the failure occurrence.

For edge switches in set  $\delta(j) \setminus \gamma(j)$ , the incident edge links do not fail. Therefore, the number of operating paths after failure is determined by the failed core links. Each failed core link disrupts one path from each edge switch in  $\delta(j) \setminus \gamma(j)$ . Thus, under failure scenario  $f = (k_c, k) \in \mathcal{S}_F^{j,k}$ , a total of  $k_c$  paths are disrupted from each edge switch in  $\delta(j) \setminus \gamma(j)$ . For each  $i$  in  $\delta(j) \setminus \gamma(j)$ , because the traffic carried by each path from  $i$  is uniformly  $\frac{1}{m} \sum_{i' \in \mathcal{N}_E, i' \neq i} \lambda_{i'i}$  before failure, the total amount of disrupted traffic originating from  $i$  is  $\frac{k_c}{m} \sum_{i' \in \mathcal{N}_E, i' \neq i} \lambda_{i'i}$ ,  $\forall i \in \delta(j) \setminus \gamma(j)$ . The disrupted traffic is evenly assigned to the remaining  $m - k_c$  operating paths from  $i$ , where link  $(j, u)$  is on one and only one remaining operating path from each  $i$  in  $\delta(j) \setminus \gamma(j)$ . Consequently, link  $(j, u)$  carries  $\frac{1}{m - k_c}$  the amount of disrupted traffic from each  $i$  in  $\delta(j) \setminus \gamma(j)$  after the failure.

Let  $\phi_{(j,u)}(k_c, k; \Lambda)$  denote the load on link  $(j, u)$  under traffic matrix  $\Lambda$  and failure scenario  $(k_c, k)$  in  $\mathcal{S}_F^{j,k}$ . Putting it all together, we have

$$\begin{aligned} \phi_{(j,u)}(k_c, k; \Lambda) &= \sum_{i \in \delta(j)} \frac{\sum_{i' \in \mathcal{N}_E, i' \neq i} \lambda_{i'i}}{m} + \sum_{i \in \gamma(j)} \frac{\sum_{i' \in \mathcal{N}_E, i' \neq i} \lambda_{i'i}}{m} \left( \frac{m}{2} + k_c \right) \\ &\quad + \sum_{i \in \delta(j) \setminus \gamma(j)} \frac{\sum_{i' \in \mathcal{N}_E, i' \neq i} \lambda_{i'i}}{m} \frac{k_c}{m - k_c}, \quad \forall \Lambda \in \mathcal{T}, j \in \mathcal{N}_A, \\ &\quad u \in \mathcal{N}_C, k_c \in \mathcal{K}(k), 1 \leq k \leq \frac{m}{2} - 1, \end{aligned} \quad (14)$$

where the first term denotes the traffic load before failure and the second and third terms denote the load increase after failure from edge switches in sets  $\gamma(j)$  and  $\delta(j) \setminus \gamma(j)$ , respectively.

So far, we have developed a load expression for any valid traffic matrix under any failure scenario in  $\mathcal{S}_F^{j,k}$ . Next, we maximize the load over all valid traffic matrices. We let

$$\begin{aligned} \rho_{(j,u)}(k_c, k) &= \max_{\Lambda \in \mathcal{T}} \phi_{(j,u)}(k_c, k; \Lambda), \quad \forall j \in \mathcal{N}_A, \\ &\quad u \in \mathcal{N}_C, k_c \in \mathcal{K}(k), 1 \leq k \leq \frac{m}{2} - 1. \end{aligned} \quad (15)$$

Introducing (1) into (14) yields

$$\begin{aligned} \phi_{(j,u)}(k_c, k; \Lambda) &\leq \sum_{i \in \delta(j)} \frac{n_s r}{m} + \sum_{i \in \gamma(j)} \frac{n_s r}{m} \left( \frac{m}{2} + k_c \right) + \sum_{i \in \delta(j) \setminus \gamma(j)} \frac{n_s r}{m} \frac{k_c}{m - k_c}, \\ &\quad \forall \Lambda \in \mathcal{T}, j \in \mathcal{N}_A, u \in \mathcal{N}_C, k_c \in \mathcal{K}(k), \\ &\quad 1 \leq k \leq \frac{m}{2} - 1. \end{aligned} \quad (16)$$

Equality is achieved when all edge switches send traffic at the maximum rate  $n_s r$  (with an infinite number of valid traffic matrices). It immediately follows that

$$\rho_{(j,u)}(k_c, k) = \sum_{i \in \delta(j)} \frac{n_s r}{m} + \sum_{i \in \gamma(j)} \frac{n_s r (\frac{m}{2} + k_c)}{\frac{m}{2} - k_c} + \sum_{i \in \delta(j) \setminus \gamma(j)} \frac{n_s r k_c}{m - k_c},$$

$$\forall j \in \mathcal{N}_A, u \in \mathcal{N}_C, k_c \in \mathcal{K}(k),$$

$$1 \leq k \leq \frac{m}{2} - 1. \quad (17)$$

Rearranging the right side by splitting the first term into sets  $\gamma(j)$  and  $\delta(j) \setminus \gamma(j)$  and combining them with the second and third terms, respectively, we have

$$\rho_{(j,u)}(k_c, k) = \sum_{i \in \gamma(j)} \frac{n_s r}{\frac{m}{2} - k_c} + \sum_{i \in \delta(j) \setminus \gamma(j)} \frac{n_s r}{m - k_c},$$

$$\forall j \in \mathcal{N}_A, u \in \mathcal{N}_C, k_c \in \mathcal{K}(k),$$

$$1 \leq k \leq \frac{m}{2} - 1. \quad (18)$$

Eq. (18) actually combines the traffic load before failure with the load increase after failure for each edge switch in  $\delta(j)$ . Introducing  $|\gamma(j)| = k_e, |\delta(j) \setminus \gamma(j)| = \frac{m}{2} - k_e$ , and  $k_e = k - k_c$  into (18), we obtain

$$\rho_{(j,u)}(k_c, k) = k_e \frac{n_s r}{\frac{m}{2} - k_c} + \left(\frac{m}{2} - k_e\right) \frac{n_s r}{m - k_c}$$

$$= n_s r \left( \frac{k - k_c}{\frac{m}{2} - k_c} + \frac{\frac{m}{2} - k + k_c}{m - k_c} \right)$$

$$= n_s r f(k_c, k), \quad \forall j \in \mathcal{N}_A, u \in \mathcal{N}_C,$$

$$k_c \in \mathcal{K}(k), 1 \leq k \leq \frac{m}{2} - 1. \quad (19)$$

Consequently, for any given integer  $k$  with its value in the interval  $[1, \frac{m}{2} - 1]$ , the maximum load on any core link  $(j, u)$  can be formulated as

$$\max_{k_c \in \mathcal{K}(k)} \rho_{(j,u)}(k_c, k) = n_s r \cdot \max_{k_c \in \mathcal{K}(k)} f(k_c, k), \quad \forall j \in \mathcal{N}_A,$$

$$u \in \mathcal{N}_C, 1 \leq k \leq \frac{m}{2} - 1, \quad (20)$$

where the maximum is taken over all failure scenarios in set  $\mathcal{S}_F^{j,k}$ .

The reverse direction of each core link  $(j, u), \forall j \in \mathcal{N}_A, u \in \mathcal{N}_C$  carries traffic in the second routing phase, which goes from the core layer to their destination edge switches. From the symmetry of the two routing phases, we know immediately that the maximum load on each  $(u, j)$  is the same as on link  $(j, u)$ , which is the optimal value in (20). Because the minimum capacity required on any core link  $l$  is equal to its maximum load, the theorem follows.  $\square$

The following lemma solves the problem formulated in (11)–(13).

**Lemma 1.** Let  $k$  be a given integer with value  $1 \leq k \leq \frac{m}{2} - 1$ . The optimal value of the problem in (11)–(13) is given by

$$c_l(k) = \begin{cases} \frac{n_s r}{2} + \frac{k n_s r}{m} & \text{if } 1 \leq k \leq \frac{m}{6}, \\ n_s r \cdot \max_{k_c \in \{\lfloor \hat{k}_c^1 \rfloor, \lceil \hat{k}_c^1 \rceil\}} f(k_c, k) & \text{if } \frac{m}{6} < k \leq \frac{m}{2} - 1, \end{cases} \quad \forall l \in \mathcal{L}_C, \quad (21)$$

where

$$\hat{k}_c^1 = \frac{k}{2} + \frac{m}{4} - \frac{1}{4} [(3m - 2k)(m - 2k)]^{\frac{1}{2}}. \quad (22)$$

**Proof.** See Appendix C.  $\square$

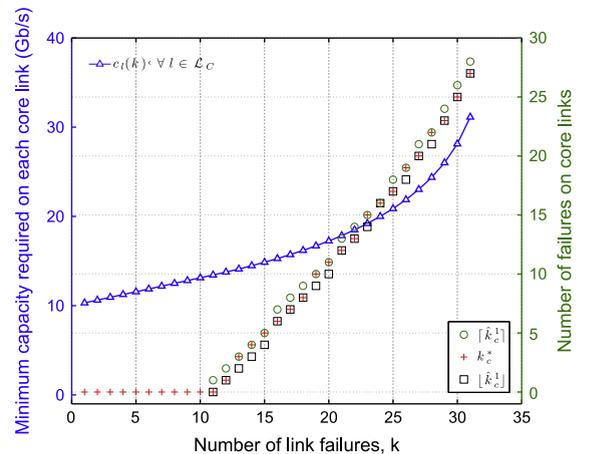
To illustrate how the minimum capacity requirement on a core link changes with the number of link failures  $k$ , we plot the result in (21) on the left axis of Fig. 8, where  $n_s, r$ , and  $m$  are set to 20, 1 Gb/s, and 64, respectively. We see that the link capacity increases linearly with  $k$  when  $k$  is within  $1 \leq k \leq \frac{m}{6}$ . However, when  $k$  is in the range  $\frac{m}{6} < k \leq \frac{m}{2} - 1$ , the link capacity increases super-linearly with  $k$ . On the right axis, we plot the corresponding optimal solution  $k_c^*$  to the problem. The value of  $k_c^*$  identifies the failure scenario in  $\mathcal{S}_F^{j,k}$  that leads to the maximum load/minimum capacity on a core link. The expression for  $k_c^*$  is given piecewise in the proof and is summarized here as

$$k_c^* = \begin{cases} 0 & \text{if } 1 \leq k \leq \frac{m}{6}, \\ \operatorname{argmax}_{k_c \in \{\lfloor \hat{k}_c^1 \rfloor, \lceil \hat{k}_c^1 \rceil\}} f(k_c, k) & \text{if } \frac{m}{6} < k \leq \frac{m}{2} - 1. \end{cases} \quad (23)$$

From (23), we know that when the value of  $k$  is in the interval  $1 \leq k \leq \frac{m}{6}$ , the maximum load on a core link is experienced when all failures are on edge links, i.e.,  $k_c^* = 0$ . Thus, we are more interested in the other case, where  $\frac{m}{6} < k \leq \frac{m}{2} - 1$ . We observe that  $k_c^*$ , the number of failures on core links, increases with  $k$  when  $k$  is in the range  $\frac{m}{6} < k \leq \frac{m}{2} - 1$ . The value of  $k_c^*$  can be either the floor or the ceiling of  $\hat{k}_c^1$ . It is not trivial to determine which of the two values  $k_c^*$  takes for a given  $k$ . Moreover, the values of  $\lfloor \hat{k}_c^1 \rfloor$  and  $\lceil \hat{k}_c^1 \rceil$  can be equal. This happens when  $\hat{k}_c^1$  is an integer (see  $k = 24$ ). In this case, we have  $k_c^* = \lfloor \hat{k}_c^1 \rfloor = \hat{k}_c^1 = \lceil \hat{k}_c^1 \rceil$ .

#### 4.3. Total link capacity

Because the minimum capacity requirement in both directions of each link is the same, we model the VL2 topology as an undirected graph. The total numbers of edge and core links are both  $m^2/2$ , as counted in Section 3.1.1. Let  $C(k)$  denote the minimum total link capacity required. Based on (3), (4), (9), and (21), we have



**Fig. 8.** Minimum capacity required on each core link in VL2 when  $n_s = 20, r = 1$  Gb/s, and  $m = 64$ .

$$C(k) = \begin{cases} \frac{m^2}{2} n_s r & \text{if } k=0, \\ \frac{m^2}{2} \left( \frac{3n_s r}{2} + \frac{kn_s r}{m} \right) & \text{if } 1 \leq k \leq \frac{m}{6}, \\ \frac{m^2}{2} n_s r \left[ 1 + \max_{k_c \in \{\hat{k}_c^1, \hat{k}_c^2\}} f(k_c, k) \right] & \text{if } \frac{m}{6} < k \leq \frac{m}{2} - 1, \end{cases} \quad \forall k \in \mathbb{Z}, \quad (24)$$

where  $f(k_c, k)$  and  $\hat{k}_c^1$  are given in (13) and (22), respectively.

## 5. Capacity allocation on fat-tree with $k$ link failures

In this section, we consider fat-tree subject to link failures. Fat-tree requires at least  $\frac{n}{2}$  link failures to disconnect the topology. Thus, with sufficient link capacity, we can guarantee full bandwidth communication among all servers up to  $\frac{n}{2} - 1$  arbitrary link failures. In the following section, we investigate the minimum link capacity requirement for this purpose.

### 5.1. Edge links

Each edge link only carries traffic that originates and terminates at the incident edge switch. This allows us to derive the capacity requirement on edge links for the general case straightforwardly. We establish the following theorem.

**Theorem 4.** *Let  $k$  be a given integer with value  $1 \leq k \leq \frac{n}{2} - 1$ . To guarantee full bandwidth communication among all of the servers under  $k$  arbitrary link failures, the minimum capacity required on each edge link is*

$$c_l(k) = r + \frac{kr}{\frac{n}{2} - k}, \quad \forall l \in \mathcal{L}_E. \quad (25)$$

**Proof.** We prove by first upper bounding the traffic load on any edge link and then show that the upper bound can be reached on an edge link under certain failure scenarios. In doing so, we show that the upper bound is actually the maximum load on any edge link, which is further equal to the minimum capacity requirement on any edge link.

We first consider the load on edge links caused by originating traffic, which all goes in the direction from the edge layer to the aggregation layer. Consider  $i$  to be a general edge switch. Let  $i$  reside in pod  $p$ . We derive the upper bound on the traffic load on any edge link incident to  $i$ . To this end, we first model the traffic load on such a link. Let  $\mathcal{F}_k$  denote the set of all possible failure scenarios with  $k$  failed links. Because the topology remains connected under  $k$  arbitrary link failures, there exists at least one edge link incident to  $i$  that is up. We denote the edge link by  $(i, j)$ , which is assumed to be incident to aggregation switch  $j$  at the other end. Additionally, there remain operating paths from  $i$  to any other edge switch  $i'$  ( $\forall i' \in \mathcal{N}_E \setminus \{i\}$ ) under any failure scenario  $f$  in  $\mathcal{F}_k$ . We let  $\hat{w}_{ii'}^f$  and  $\check{w}_{ii'}^f$  denote the number of remaining operating paths from  $i$  to edge switch  $i'$  in the same pod (i.e.,  $i' \in \mathcal{N}_E^p \setminus \{i\}$ ) and in a different pod (i.e.,  $i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p$ ), respectively, under failure scenario  $f$ . Among  $\hat{w}_{ii'}^f$  and  $\check{w}_{ii'}^f$ , we further denote the number of

operating paths that traverse edge link  $(i, j)$  by  $\hat{x}_{(ij)}^{f, ii'}$  and  $\check{x}_{(ij)}^{f, ii'}$ , respectively. Because traffic from  $i$  to any other edge switch  $i'$  is evenly distributed over the remaining operating paths from  $i$  to  $i'$ , the load on link  $(i, j)$  under traffic matrix  $\Lambda$  and failure scenario  $f$  can be expressed as

$$\begin{aligned} \varphi_{(ij)}(f; \Lambda) &= \sum_{i' \in \mathcal{N}_E^p \setminus \{i\}} \frac{\hat{x}_{(ij)}^{f, ii'} \lambda_{ii'}}{\hat{w}_{ii'}^f} + \sum_{i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p} \frac{\check{x}_{(ij)}^{f, ii'} \lambda_{ii'}}{\check{w}_{ii'}^f}, \\ &\forall i \in \mathcal{N}_E^p, j \in \mathcal{N}_A^p, p \in \mathcal{P}, f \in \mathcal{F}_k, \Lambda \in \mathcal{T}, \\ &1 \leq k \leq \frac{n}{2} - 1. \end{aligned} \quad (26)$$

Note that given  $f$  and  $\Lambda$ , the values of  $\hat{x}_{(ij)}^{f, ii'}$ ,  $\check{x}_{(ij)}^{f, ii'}$ ,  $\hat{w}_{ii'}^f$ , and  $\check{w}_{ii'}^f$  are all deterministic and fixed.

We establish the upper bound on  $\varphi_{(ij)}(f; \Lambda)$  through upper bounding  $\hat{x}_{(ij)}^{f, ii'}$  and  $\check{x}_{(ij)}^{f, ii'}$  and lower bounding  $\hat{w}_{ii'}^f$  and  $\check{w}_{ii'}^f$  over all failure scenarios and all destination edge switches. From the background discussions in Section 3.2.3, we know that when there are no failures, each edge link incident to  $i$  is on one and only one of the  $n/2$  paths carrying intra-pod traffic from  $i$  to local edge switch  $i'$  ( $\forall i' \in \mathcal{N}_E^p \setminus \{i\}$ ) and is on  $n/2$  of the  $(n/2)^2$  paths carrying inter-pod traffic from  $i$  to remote edge switch  $i'$  ( $\forall i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p$ ). Under any failure scenario  $f$ , the number of operating paths on edge link  $(i, j)$  is no greater than in the no-failure case. Hence, the upper bounds on  $\hat{x}_{(ij)}^{f, ii'}$  and  $\check{x}_{(ij)}^{f, ii'}$  are given by

$$\begin{aligned} \hat{x}_{(ij)}^{f, ii'} &\leq 1, \quad \forall i' \in \mathcal{N}_E^p \setminus \{i\}, \\ \check{x}_{(ij)}^{f, ii'} &\leq \frac{n}{2}, \quad \forall i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p, \\ &i \in \mathcal{N}_E^p, j \in \mathcal{N}_A^p, p \in \mathcal{P}, f \in \mathcal{F}_k, 1 \leq k \leq \frac{n}{2} - 1. \end{aligned} \quad (27)$$

Next, we develop the lower bounds on  $\hat{w}_{ii'}^f$  and  $\check{w}_{ii'}^f$ . To this end, we compare the impact of different failure locations on the number of remaining operating paths. For  $\hat{w}_{ii'}^f$ , because intra-pod traffic does not go beyond the aggregation switches, failures on the core links in  $p$  or links in remote pods do not decrease  $\hat{w}_{ii'}^f$ . Thus, we only need to compare failures on edge links incident to  $i$  and to local edge switches other than  $i$ . From the paths carried by an edge link under the no-failure scenario, it is easy to find that in the former case, a failed edge link maximally disrupts one path from  $i$  to every local edge switch other than  $i$ , while in the latter case, a failed edge link maximally disrupts only one path from  $i$ , which goes to the edge switch where the failed link is incident. We can see that the former case has a worse potential impact, which includes the impact of the latter one. Note that by “maximally” in both cases, we take into account the incremental impact of a failure over multiple existing failures, where paths traversing the new failure may have been disrupted due to the existing failures. In this case, the incremental paths to be disrupted by the new failure are fewer than the paths traversing the new failure. It follows that for  $k$  failures,  $\hat{w}_{ii'}^f$  is lower bounded by

$$\begin{aligned} \hat{w}_{i i'}^f &\geq \frac{n}{2} - k, \quad \forall i' \in \mathcal{N}_E^p \setminus \{i\}, \quad i \in \mathcal{N}_E^p, \quad p \in \mathcal{P}, \quad f \in \mathcal{F}_k, \\ 1 &\leq k \leq \frac{n}{2} - 1, \end{aligned} \quad (28)$$

where  $\frac{n}{2}$  is the number of operating paths from  $i$  to local edge switch  $i'$  in the no-failure case, as discussed in Section 3.2.3.

For  $\hat{w}_{i i'}^f$ , the failure locations of impact are those that carry paths from  $i$  to one or more remote edge switches. They are enumerated as follows: (1) edge links incident to  $i$ ; (2) edge links incident to remote edge switches; (3) core links in pod  $p$ ; and (4) core links in remote pods. In each case, we list for a general link all paths it carries that go from  $i$  to remote edge switches. The listed paths bound the number of paths to be disrupted when a link fails over multiple existing failures, as discussed above. Among the four cases, we find the worst case that bounds the minimum number of operating paths from  $i$  to a remote edge switch in  $k$  failures. Specifically, in case 1, an edge link incident to  $i$  carries, for every remote edge switch,  $n/2$  paths that go from  $i$  to that switch. In case 2, let  $i'$  denote the particular remote edge switch to which an edge link is incident. An edge link incident to  $i'$  carries  $n/2$  paths from  $i$  to  $i'$ . In case 3, a core link in pod  $p$  carries, for each remote edge switch, one and only one path that goes from  $i$  to that switch. In case 4, let  $p'$  denote the particular remote pod where a core link is located. A core link in  $p'$  carries, for every edge switch in  $p'$ , one and only one path that goes from  $i$  to the corresponding edge switch. Comparing the four cases, it is easy to find that one (additional) link failure can maximally disrupt  $n/2$  paths from  $i$  to any remote edge switch  $i'$ . It follows that in  $k$  failures, the total number of paths to be disrupted from  $i$  to any remote  $i'$  is no greater than  $k \frac{n}{2}$ . Therefore, the lower bound on  $\hat{w}_{i i'}^f$  can be written as

$$\begin{aligned} \hat{w}_{i i'}^f &\geq \frac{n}{2} \left( \frac{n}{2} - k \right), \quad \forall i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p, \quad i \in \mathcal{N}_E^p, \quad p \in \mathcal{P}, \quad f \in \\ &\mathcal{F}_k, \quad 1 \leq k \leq \frac{n}{2} - 1, \end{aligned} \quad (29)$$

where  $(n/2)^2$  is the number of operating paths from  $i$  to remote edge switch  $i'$  under the no-failure scenario, as discussed in Section 3.2.3.

Introducing (27)–(29) into (26), we obtain the upper bound on traffic load on any edge link  $(i, j)$  as

$$\begin{aligned} \varphi_{(i, j)}(f; \Lambda) &\leq \sum_{i' \in \mathcal{N}_E^p \setminus \{i\}} \frac{\lambda_{i i'}}{\frac{n}{2} - k} + \sum_{i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p} \frac{\frac{n}{2} \lambda_{i i'}}{\frac{n}{2} - k} \\ &= \sum_{i' \in \mathcal{N}_E \setminus \{i\}} \frac{\lambda_{i i'}}{\frac{n}{2} - k} \leq \frac{\frac{n}{2} r}{\frac{n}{2} - k} = r + \frac{kr}{\frac{n}{2} - k}, \\ &\forall i \in \mathcal{N}_E^p, \quad j \in \mathcal{N}_A^p, \quad p \in \mathcal{P}, \quad f \in \mathcal{F}_k, \\ &\Lambda \in \mathcal{T}, \quad 1 \leq k \leq \frac{n}{2} - 1, \end{aligned} \quad (30)$$

where the second inequality follows from (5).

Last, we show that the upper-bound load given in (30) can be experienced on edge link  $(i, j)$  under the failure scenario where  $k$  link failures are on  $k$  of the  $n/2$  edge links incident to  $i$  (other than link  $(i, j)$ ). We illustrate such a scenario in Fig. 9, where there remain  $\frac{n}{2} - k$  operating edge links incident to  $i$ , with link  $(i, j)$  being one of them. Each operating link carries, for each local edge switch  $i' \neq i$ , one

and only one operating path that transfers intra-pod traffic from  $i$  to  $i'$ . Moreover, for each remote edge switch  $i'$ , each operating link carries  $n/2$  operating paths that transfer inter-pod traffic from  $i$  to  $i'$ . Hence, for every traffic demand  $\lambda_{i i'}$ ,  $\forall i' \in \mathcal{N}_E \setminus \{i\}$ , all operating links incident to  $i$  carry the same number of operating paths from  $i$  to  $i'$ . Considering that each traffic demand  $\lambda_{i i'}$  is evenly balanced over the corresponding operating paths, all traffic from  $i$  is thus evenly split over the  $\frac{n}{2} - k$  operating links incident to  $i$ . In other words, the traffic load on link  $(i, j)$  (and on each of the  $\frac{n}{2} - k$  operating links incident to  $i$ ) is  $\sum_{i' \in \mathcal{N}_E \setminus \{i\}} \frac{\lambda_{i i'}}{\frac{n}{2} - k}$ , which takes the maximum value  $\frac{\frac{n}{2} r}{\frac{n}{2} - k} = r + \frac{kr}{\frac{n}{2} - k}$  when traffic from  $i$  reaches the ingress capacity limit given by (5).

The reverse direction of edge link  $(i, j)$  carries traffic that terminates at  $i$ . From the symmetry of the two routing phases, it immediately follows that the reverse direction of link  $(i, j)$ , i.e., link  $(j, i)$ , experiences the same maximum load as link  $(i, j)$ . Considering all failure scenarios, the maximum load on each edge link is uniform. Consequently, the minimum capacity required on each edge link  $l$  is  $r + \frac{kr}{\frac{n}{2} - k}$  and the theorem follows.  $\square$

### 5.2. Core links

Each core link in pod  $p$  carries inter-pod traffic that originates and terminates at every edge switch in  $p$ . The following theorem generalizes the capacity requirement on the core links for  $k$  arbitrary link failures.

**Theorem 5.** Let  $k$  be a given integer with value  $1 \leq k \leq \frac{n}{2} - 1$ . To guarantee full bandwidth communication among all of the servers under  $k$  arbitrary link failures, the minimum capacity required on each core link is

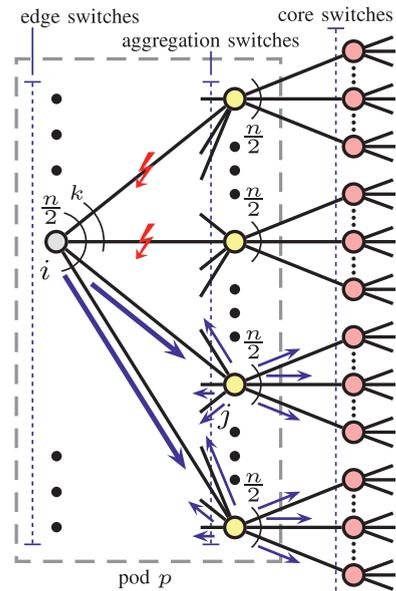


Fig. 9. Failure scenario with  $k$  failed links that leads to the maximum traffic load on edge links incident to edge switch  $i$ .

$$c_l(k) = r + \frac{kr}{\left(\frac{n}{2} - k\right)\frac{n}{2}}, \quad \forall l \in \mathcal{L}_C. \quad (31)$$

**Proof.** Similar to the case of edge links on fat-tree, we first establish the upper bound on the traffic load on any core link and then show the failure scenarios where the upper-bound load can be experienced on a core link. In doing so, we obtain the maximum load on any core link, which is equal to the minimum link capacity requirement.

We first consider the load on core links in the direction from the aggregation layer to the core layer, which is caused by traffic in the first routing phase, i.e., outgoing inter-pod traffic. Note that intra-pod traffic does not go through core links and thus is not considered in this proof. Similar to the edge link case, we upper bound the traffic load on any core link under any failure scenario by first formulating its traffic load. In contrast to the edge link case, we model the link load by decomposing every contributing traffic demand in the form of the load before failure and the load increase after failure. Specifically, consider  $(j, u)$  to be a general core link that is up under failure scenario  $f$  ( $\forall f \in \mathcal{F}_k$ ). Link  $(j, u)$  is incident to aggregation switch  $j$  and core switch  $u$ . Let  $p$  denote the pod where link  $(j, u)$  resides. We refer to edge switches inside and outside  $p$  as local and remote edge switches, respectively. For each local edge switch  $i$  (i.e.,  $i \in \mathcal{N}_E^p$ ) and each remote edge switch  $i'$  (i.e.,  $i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p$ ), link  $(j, u)$  carries one and only one path from  $i$  to  $i'$ . Under a failure scenario, paths traversing link  $(j, u)$  can fail. We denote the status of each traversing path by binary indicator  $y_{(j,u)}^{f,ii'}$ . Indicator  $y_{(j,u)}^{f,ii'}$  equals one if the one path from local  $i$  to remote  $i'$  carried by link  $(j, u)$  is up under failure scenario  $f$  and zero otherwise. If  $y_{(j,u)}^{f,ii'} = 1$ , the traffic demand  $\lambda_{ii'}$  introduces a load on link  $(j, u)$  under failure scenario  $f$ , which can be partitioned into load before failure and load increase after failure. From the discussions in Section 3.2.4, we know that the load incurred before failure is  $\frac{\lambda_{ii'}}{(n/2)^2}$ . In the event of any failure scenario, the disrupted paths from  $i$  to  $i'$  are deterministic and known. The disrupted traffic (i.e., traffic carried by the disrupted paths) is evenly assigned to the remaining operating paths from  $i$  to  $i'$ . We let  $\mu_{ii'}(f; \Lambda)$  denote the total amount of disrupted traffic from  $i$  to  $i'$  under failure scenario  $f$  and traffic matrix  $\Lambda$ . Because link  $(j, u)$  is on one and only one operating path from  $i$  to  $i'$ , the load increase after failure can be expressed as  $\mu_{ii'}(f; \Lambda) / \check{w}_{ii'}^f$ , recalling that  $\check{w}_{ii'}^f$  denotes the number of remaining operating paths from local edge switch  $i$  to remote edge switch  $i'$  under failure scenario  $f$ . Conversely, if  $y_{(j,u)}^{f,ii'} = 0$ , the traffic demand  $\lambda_{ii'}$  does not incur any load on link  $(j, u)$  under failure scenario  $f$ . It is easy to find that both cases can be uniformly written as

$$\frac{y_{(j,u)}^{f,ii'} \lambda_{ii'}}{\left(\frac{n}{2}\right)^2} + \frac{y_{(j,u)}^{f,ii'} \mu_{ii'}(f; \Lambda)}{\check{w}_{ii'}^f},$$

which thus holds for traffic from any local edge switch  $i$  to any remote edge switch  $i'$ . Taking the sum over  $i$  and  $i'$ , we

obtain the total load on link  $(j, u)$  under failure scenario  $f$  and traffic matrix  $\Lambda$  as

$$\begin{aligned} \varphi_{(j,u)}(f; \Lambda) &= \sum_{i \in \mathcal{N}_E^p} \sum_{i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p} \left( \frac{y_{(j,u)}^{f,ii'} \lambda_{ii'}}{\left(\frac{n}{2}\right)^2} + \frac{y_{(j,u)}^{f,ii'} \mu_{ii'}(f; \Lambda)}{\check{w}_{ii'}^f} \right), \\ &\quad \forall j \in \mathcal{N}_A^p, \quad u \in \mathcal{N}_C^j, \quad p \in \mathcal{P}, \quad f \in \mathcal{F}_k, \\ &\quad \Lambda \in \mathcal{T}, \quad 1 \leq k \leq \frac{n}{2} - 1, \end{aligned} \quad (32)$$

where for each given  $f$  and  $\Lambda$ , the values of  $y_{(j,u)}^{f,ii'}$ ,  $\mu_{ii'}(f; \Lambda)$ , and  $\check{w}_{ii'}^f$  are all deterministic and fixed.

Next, we develop the upper bound on  $\varphi_{(j,u)}(f; \Lambda)$ . The approach is similar to the edge link case, which is achieved through the bounding of all parameters involved in the link load model. In the core link case, this process requires upper bounds on  $y_{(j,u)}^{f,ii'}$  and  $\mu_{ii'}(f; \Lambda)$  and a lower bound on  $\check{w}_{ii'}^f$ . Because  $y_{(j,u)}^{f,ii'}$  is a binary indicator, the upper bound on  $y_{(j,u)}^{f,ii'}$  can be written straightforwardly as

$$\begin{aligned} y_{(j,u)}^{f,ii'} &\leq 1, \quad \forall i \in \mathcal{N}_E^p, \quad i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p, \quad j \in \mathcal{N}_A^p, \quad u \\ &\quad \in \mathcal{N}_C^j, \quad p \in \mathcal{P}, \quad f \in \mathcal{F}_k, \quad 1 \leq k \leq \frac{n}{2} - 1. \end{aligned} \quad (33)$$

The lower bound on  $\check{w}_{ii'}^f$  is given in (29). Thus, we focus on upper bounding  $\mu_{ii'}(f; \Lambda)$ . Because with no network failures, the maximum load on any edge or core link is uniformly  $r$  according to (7) and (8), it follows that the failure of  $k$  links can maximally disrupt an amount  $kr$  of inter-pod traffic originating from  $p$ , i.e.,

$$\begin{aligned} \sum_{i \in \mathcal{N}_E^p} \sum_{i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p} \mu_{ii'}(f; \Lambda) &\leq kr, \quad \forall p \in \mathcal{P}, \quad f \in \mathcal{F}_k, \quad \Lambda \\ &\quad \in \mathcal{T}, \quad 1 \leq k \leq \frac{n}{2} - 1. \end{aligned} \quad (34)$$

Introducing (5), (29), (33), and (34) into (32), we establish the upper bound on the traffic load on any core link  $(j, u)$  to be

$$\begin{aligned} \varphi_{(j,u)}(f; \Lambda) &\leq \sum_{i \in \mathcal{N}_E^p} \sum_{i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p} \frac{\lambda_{ii'}}{\left(\frac{n}{2}\right)^2} + \sum_{i \in \mathcal{N}_E^p} \sum_{i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p} \frac{\mu_{ii'}(f; \Lambda)}{\left(\frac{n}{2} - k\right)\frac{n}{2}} \\ &\leq |\mathcal{N}_E^p| \frac{\frac{n}{2}r}{\left(\frac{n}{2}\right)^2} + \frac{kr}{\left(\frac{n}{2} - k\right)\frac{n}{2}} \\ &= r + \frac{kr}{\left(\frac{n}{2} - k\right)\frac{n}{2}}, \quad \forall j \in \mathcal{N}_A^p, \quad u \in \mathcal{N}_C^j, \quad p \\ &\quad \in \mathcal{P}, \quad f \in \mathcal{F}_k, \quad \Lambda \in \mathcal{T}, \quad 1 \leq k \leq \frac{n}{2} - 1, \end{aligned} \quad (35)$$

where the first inequality follows from (29) and (33) and the second inequality follows from (5) and (34).

Last, we show that the upper-bound load in (35) can be reached on a core link in  $p$  under one of the following failure scenarios: (1) all  $k$  failed links are incident to local edge switch  $i_1$  ( $\forall i_1 \in \mathcal{N}_E^p$ ); (2) some of the failed links are incident to local edge switch  $i_2$  ( $\forall i_2 \in \mathcal{N}_E^p$ ), the rest are incident to remote edge switch  $i'_2$  ( $\forall i'_2 \in \mathcal{N}_E \setminus \mathcal{N}_E^p$ ), and no two failed links are on the same path connecting  $i_2$  and  $i'_2$  so that all failed links in total disrupt  $k\frac{n}{2}$  paths from  $i_2$  to  $i'_2$ ; and (3) all  $k$  failed links are incident to remote edge switch

$i_3$ . We illustrate these scenarios in Fig. 10. In particular, we denote the scenarios in Figs. 10(a), 10(b), and 10(c) by  $f_1$ ,  $f_2$ , and  $f_3$ , respectively, and discuss them individually below. In general, given failure scenario  $f$ , the load on any core link  $(j, u)$  under traffic matrix  $\Lambda$  can be expressed more straightforwardly as

$$\begin{aligned} \varphi_{(j,u)}(f; \Lambda) &= \sum_{i \in \mathcal{N}_E^p} \sum_{i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p} \frac{y_{(j,u)}^{f,ii'} \lambda_{ii'}}{\tilde{w}_{ii'}^f}, \\ \forall j \in \mathcal{N}_A^p, u \in \mathcal{N}_C, p \in \mathcal{P}, f \in \mathcal{F}_k, \\ \Lambda \in \mathcal{T}, 1 \leq k \leq \frac{n}{2} - 1. \end{aligned} \quad (36)$$

Formula (36) is equivalent to the formula in (32) but takes a different modeling viewpoint, where the link load is derived directly under the failure scenario rather than decomposed into the load before and after failure as in (32). We calculate the link load based on (36) for all three scenario cases.

In scenario 1, as illustrated in Fig. 10(a), there exist  $\frac{n}{2} - k$  aggregation switches in  $p$  that are not incident to failed links. Among all core links incident to such an aggregation switch, we randomly choose one link and denote it by  $(j_1, u_1)$ , as shown in Fig. 10(a). Because all failed links only affect traffic originating (and terminating) at  $i_1$ , traffic from any local edge switches other than  $i_1$  is routed as it is in the no-failure case. Thus, for any  $i \in \mathcal{N}_E^p \setminus \{i_1\}$ , we have

$$\begin{aligned} \tilde{w}_{ii'}^{f_1} &= (n/2)^2 \quad \text{and} \quad y_{(j_1,u_1)}^{f_1,ii'} = 1, \quad \forall i \in \mathcal{N}_E^p \setminus \{i_1\}, i' \\ &\in \mathcal{N}_E \setminus \mathcal{N}_E^p, 1 \leq k \leq \frac{n}{2} - 1. \end{aligned} \quad (37)$$

Conversely, for edge switch  $i_1$ , each failed link disrupts  $n/2$  paths from  $i_1$  to any remote edge switch  $i'$ . Hence, from  $i_1$  to any remote  $i'$ , the number of operating paths is reduced

to  $(\frac{n}{2} - k) \frac{n}{2}$ , among which one and only one path traverses link  $(j_1, u_1)$ . In other words, we have

$$\begin{aligned} \tilde{w}_{i_1 i'}^{f_1} &= \left(\frac{n}{2} - k\right) \frac{n}{2} \quad \text{and} \quad y_{(j_1,u_1)}^{f_1,i_1 i'} = 1, \\ \forall i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p, 1 \leq k \leq \frac{n}{2} - 1. \end{aligned} \quad (38)$$

Introducing (37) and (38) into (36), we obtain the load on link  $(j_1, u_1)$  under failure scenario  $f_1$  and traffic matrix  $\Lambda$  as

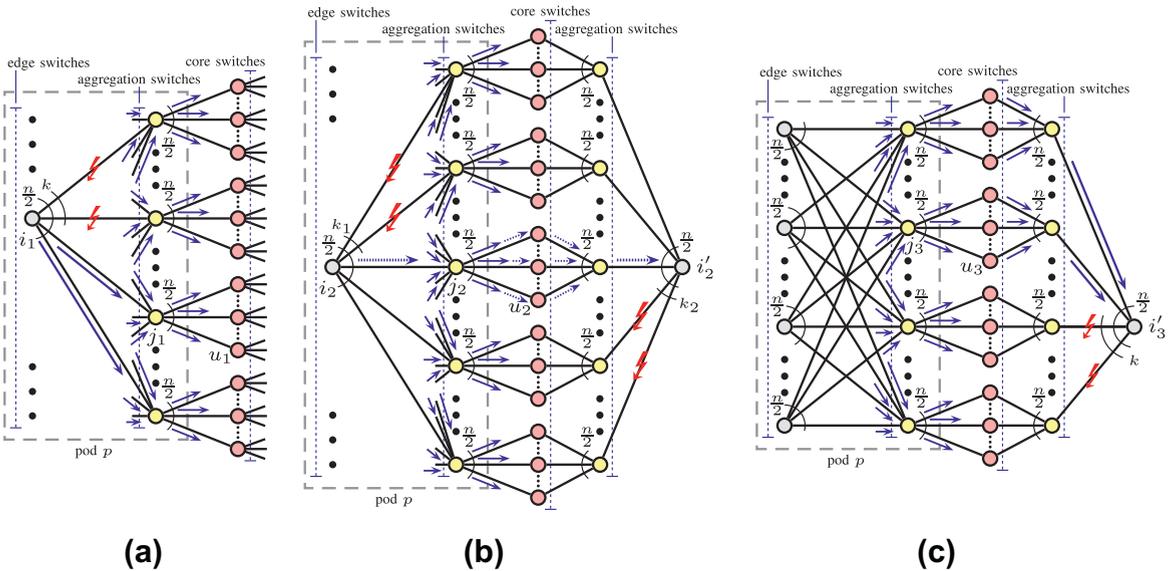
$$\begin{aligned} \varphi_{(j_1,u_1)}(f_1; \Lambda) &= \sum_{i \in \mathcal{N}_E^p \setminus \{i_1\}} \sum_{i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p} \frac{\lambda_{ii'}}{\left(\frac{n}{2}\right)^2} + \sum_{i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p} \frac{\lambda_{i_1 i'}}{\left(\frac{n}{2} - k\right) \frac{n}{2}}, \\ \forall \Lambda \in \mathcal{T}, 1 \leq k \leq \frac{n}{2} - 1. \end{aligned} \quad (39)$$

Applying (5) to (39) yields

$$\begin{aligned} \varphi_{(j_1,u_1)}(f_1; \Lambda) &\leq \sum_{i \in \mathcal{N}_E^p \setminus \{i_1\}} \frac{\frac{n}{2} r}{\left(\frac{n}{2}\right)^2} + \frac{\frac{n}{2} r}{\left(\frac{n}{2} - k\right) \frac{n}{2}} \\ &= \left(\frac{n}{2} - 1\right) \frac{\frac{n}{2} r}{\left(\frac{n}{2}\right)^2} + \frac{\frac{n}{2} r}{\left(\frac{n}{2} - k\right) \frac{n}{2}} \\ &= r + \frac{kr}{\left(\frac{n}{2} - k\right) \frac{n}{2}}, \quad \forall \Lambda \in \mathcal{T}, 1 \leq k \leq \frac{n}{2} - 1, \end{aligned}$$

which implies the maximum load on link  $(j_1, u_1)$  to be  $r + \frac{kr}{(\frac{n}{2}-k)\frac{n}{2}}$ . This situation is achieved when all edge switches in  $p$  send inter-pod traffic at their maximum rates  $\frac{n}{2} r$ . Note that the destinations of traffic from  $i_1$  can vary among all remote edge switches.

In scenario 2, as illustrated in Fig. 10(b), let  $k_1$  and  $k_2$  denote the number of failed links incident to  $i_2$  and  $i_2'$ , respectively. We have  $k_1 \geq 1, k_2 \geq 1$ , and  $k_1 + k_2 = k$ . There remain  $(\frac{n}{2} - k) \frac{n}{2}$  operating paths from  $i_2$  to  $i_2'$ , each corresponding to one core link in  $p$ . Among these  $(\frac{n}{2} - k) \frac{n}{2}$  core links, we randomly choose one and denote it by  $(j_2, u_2)$ , as shown in Fig. 10(b). We consider the traffic load



**Fig. 10.** Failure scenarios with  $k$  failed links that lead to the maximum traffic load on core link  $(j_a, u_a)$ ,  $\forall a \in \{1, 2, 3\}$ . (a) Scenario  $f_1$ : all  $k$  failed links are incident to local edge switch  $i_1$ . (b) Scenario  $f_2$ :  $k_1$  failed links are incident to local edge switch  $i_2$ , the rest (i.e.,  $k_2 = k - k_1$ ) are incident to remote edge switch  $i_2'$ , and no two failed links are on the same path connecting  $i_2$  and  $i_2'$ . (c) Scenario  $f_3$ : all  $k$  failed links are incident to remote edge switch  $i_3'$ .

on link  $(j_2, u_2)$ . Specifically, link  $(j_2, u_2)$  carries one and only one operating path from any local edge switch in  $p$  to any remote edge switch, i.e.,

$$y_{(j_2, u_2)}^{f_2, i i'} = 1, \quad \forall i \in \mathcal{N}_E^p, \quad i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p, \quad 1 \leq k \leq \frac{n}{2} - 1. \quad (40)$$

Moreover, it is easy to find that for each local edge switch  $i \neq i_2$ , there remain  $(\frac{n}{2} - k_2) \frac{n}{2}$  operating paths from  $i$  to  $i_2$  and  $(n/2)^2$  operating paths from  $i$  to any remote edge switch other than  $i_2$ . Conversely, from local edge switch  $i_2$ , the numbers of remaining operating paths to  $i_2$  and to any remote edge switch other than  $i_2$  are  $(\frac{n}{2} - k) \frac{n}{2}$  and  $(\frac{n}{2} - k_1) \frac{n}{2}$ , respectively. Hence, we have

$$\tilde{w}_{i i'}^{f_2} = \begin{cases} (\frac{n}{2} - k_2) \frac{n}{2} & \text{if } i \neq i_2, i' = i_2 \\ (\frac{n}{2})^2 & \text{if } i \neq i_2, i' \neq i_2 \\ (\frac{n}{2} - k) \frac{n}{2} & \text{if } i = i_2, i' = i_2 \\ (\frac{n}{2} - k_1) \frac{n}{2} & \text{if } i = i_2, i' \neq i_2 \end{cases} \quad (41)$$

$$\forall i \in \mathcal{N}_E^p, \quad i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p, \quad 1 \leq k \leq \frac{n}{2} - 1.$$

Introducing (40) and (41) into (36), we obtain the load on link  $(j_2, u_2)$  under failure scenario  $f_2$  and traffic matrix  $\Lambda$  as

$$\begin{aligned} \varphi_{(j_2, u_2)}(f_2; \Lambda) &= \sum_{i \in \mathcal{N}_E^p \setminus \{i_2\}} \sum_{i' \in \mathcal{N}_E \setminus \{\mathcal{N}_E^p \cup \{i_2\}\}} \frac{\lambda_{i i'}}{(\frac{n}{2})^2} + \sum_{i \in \mathcal{N}_E^p \setminus \{i_2\}} \frac{\lambda_{i i_2}}{(\frac{n}{2} - k_2) \frac{n}{2}} \\ &+ \sum_{i' \in \mathcal{N}_E \setminus \{\mathcal{N}_E^p \cup \{i_2\}\}} \frac{\lambda_{i_2 i'}}{(\frac{n}{2} - k_1) \frac{n}{2}} + \frac{\lambda_{i_2 i_2}}{(\frac{n}{2} - k) \frac{n}{2}} \\ &= \sum_{i \in \mathcal{N}_E^p \setminus \{i_2\}} \sum_{i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p} \frac{\lambda_{i i'}}{(\frac{n}{2})^2} + \sum_{i \in \mathcal{N}_E^p \setminus \{i_2\}} \frac{k_2 \lambda_{i i_2}}{(\frac{n}{2} - k_2) (\frac{n}{2})^2} \\ &+ \sum_{i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p} \frac{\lambda_{i_2 i'}}{(\frac{n}{2} - k_1) \frac{n}{2}} + \frac{k_2 \lambda_{i_2 i_2}}{(\frac{n}{2} - k) (\frac{n}{2} - k_1) \frac{n}{2}}, \\ &\forall \Lambda \in \mathcal{T}, \quad 1 \leq k \leq \frac{n}{2} - 1. \end{aligned} \quad (42)$$

Next, we find the maximum value of  $\varphi_{(j_2, u_2)}(f_2; \Lambda)$  among all valid traffic matrices. Applying (5) to (42) yields

$$\begin{aligned} \varphi_{(j_2, u_2)}(f_2; \Lambda) &\leq |\mathcal{N}_E^p \setminus \{i_2\}| \frac{\frac{n}{2} r}{(\frac{n}{2})^2} + \frac{k_2}{(\frac{n}{2} - k_2) (\frac{n}{2})^2} \sum_{i \in \mathcal{N}_E^p \setminus \{i_2\}} \lambda_{i i_2} \\ &+ \frac{\frac{n}{2} r}{(\frac{n}{2} - k_1) \frac{n}{2}} + \frac{k_2}{(\frac{n}{2} - k) (\frac{n}{2} - k_1) \frac{n}{2}} \lambda_{i_2 i_2}, \\ &\forall \Lambda \in \mathcal{T}, \quad 1 \leq k \leq \frac{n}{2} - 1. \end{aligned} \quad (43)$$

We compare the second and fourth terms on the right side of the inequality. Because  $\sum_{i \in \mathcal{N}_E^p \setminus \{i_2\}} \lambda_{i i_2}$  and  $\lambda_{i_2 i_2}$  satisfy

$$\sum_{i \in \mathcal{N}_E^p \setminus \{i_2\}} \lambda_{i i_2} + \lambda_{i_2 i_2} \leq \frac{n}{2} r,$$

which follows from the egress capacity limit given in (6), for the highest link load, the one with larger weight factor takes the maximum value  $\frac{n}{2} r$ , while the other takes the minimum value 0. Because  $k_1 > 0$ ,  $k_2 > 0$ , and  $k_1 + k_2 = k \leq \frac{n}{2} - 1$ , we have

$$\frac{n}{2} - k_2 > \frac{n}{2} - k > 0 \quad \text{and} \quad \frac{n}{2} > \frac{n}{2} - k_1 > 0.$$

It follows that the weight factor of  $\lambda_{i_2 i_2}$  is greater than the weight factor of  $\sum_{i \in \mathcal{N}_E^p \setminus \{i_2\}} \lambda_{i i_2}$ , i.e.,

$$\frac{k_2}{(\frac{n}{2} - k) (\frac{n}{2} - k_1) \frac{n}{2}} > \frac{k_2}{(\frac{n}{2} - k_2) (\frac{n}{2})^2}.$$

Hence, the highest link load is achieved when  $\lambda_{i_2 i_2} = \frac{n}{2} r$  and  $\sum_{i \in \mathcal{N}_E^p \setminus \{i_2\}} \lambda_{i i_2} = 0$ . Making these substitutions into (43), we have

$$\begin{aligned} \varphi_{(j_2, u_2)}(f_2; \Lambda) &\leq \left(\frac{n}{2} - 1\right) \frac{\frac{n}{2} r}{(\frac{n}{2})^2} + \frac{\frac{n}{2} r}{(\frac{n}{2} - k_1) \frac{n}{2}} + \frac{k_2 \frac{n}{2} r}{(\frac{n}{2} - k) (\frac{n}{2} - k_1) \frac{n}{2}} \\ &= r + \frac{kr}{(\frac{n}{2} - k) \frac{n}{2}}, \quad \forall \Lambda \in \mathcal{T}, \quad 1 \leq k \leq \frac{n}{2} - 1. \end{aligned}$$

The maximum load is experienced when each edge switch in  $p$  sends inter-pod traffic at its maximum rate  $\frac{n}{2} r$ , and all traffic from  $i_2$  is destined for  $i_2$ , i.e.,  $\lambda_{i_2 i_2} = \frac{n}{2} r$ .

In scenario 3, as illustrated in Fig. 10(c), there exist  $(\frac{n}{2} - k) \frac{n}{2}$  core links in  $p$  such that each of them carries, for each local edge switch  $i$  in  $p$ , one and only one operating path from  $i$  to  $i_3$ . Among these core links, we randomly select one link and denote it by  $(j_3, u_3)$ , as shown in Fig. 10(c). We study the traffic load on link  $(j_3, u_3)$ . Because, in fact, link  $(j_3, u_3)$  carries one and only one operating path from any local edge switch to any remote edge switch, we have

$$y_{(j_3, u_3)}^{f_3, i i'} = 1, \quad \forall i \in \mathcal{N}_E^p, \quad i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p, \quad 1 \leq k \leq \frac{n}{2} - 1. \quad (44)$$

Moreover, it is easy to find that, for each local edge switch  $i$ , there remain  $(n/2)^2$  operating paths from  $i$  to any remote edge switch other than  $i_3$ , while the number of operating paths from  $i$  to  $i_3$  is reduced to  $(\frac{n}{2} - k) \frac{n}{2}$ , i.e.,

$$\tilde{w}_{i i'}^{f_3} = \begin{cases} (\frac{n}{2})^2 & \text{if } i' \neq i_3 \\ (\frac{n}{2} - k) \frac{n}{2} & \text{if } i' = i_3 \end{cases} \quad \forall i \in \mathcal{N}_E^p, \quad i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p, \quad 1 \leq k \leq \frac{n}{2} - 1. \quad (45)$$

Introducing (44) and (45) into (36), we obtain the load on link  $(j_3, u_3)$  under failure scenario  $f_3$  and traffic matrix  $\Lambda$  to be

$$\begin{aligned} \varphi_{(j_3, u_3)}(f_3; \Lambda) &= \sum_{i \in \mathcal{N}_E^p} \sum_{i' \in \mathcal{N}_E \setminus \{\mathcal{N}_E^p \cup \{i_3\}\}} \frac{\lambda_{i i'}}{(\frac{n}{2})^2} + \sum_{i \in \mathcal{N}_E^p} \frac{\lambda_{i i_3}}{(\frac{n}{2} - k) \frac{n}{2}} \\ &= \sum_{i \in \mathcal{N}_E^p} \sum_{i' \in \mathcal{N}_E \setminus \mathcal{N}_E^p} \frac{\lambda_{i i'}}{(\frac{n}{2})^2} + \sum_{i \in \mathcal{N}_E^p} \frac{k \lambda_{i i_3}}{(\frac{n}{2} - k) (\frac{n}{2})^2}, \\ &\forall \Lambda \in \mathcal{T}, \quad 1 \leq k \leq \frac{n}{2} - 1. \end{aligned} \quad (46)$$

Considering the ingress capacity limit at all local edge switches (given in (5)) and the egress capacity limit at  $i_3$  (given in (6)), we have

$$\begin{aligned} \varphi_{(j_3, u_3)}(f_3; \Lambda) &\leq |\mathcal{N}_E^p| \frac{\frac{n}{2} r}{(\frac{n}{2})^2} + \frac{k \frac{n}{2} r}{(\frac{n}{2} - k) (\frac{n}{2})^2} \\ &= r + \frac{kr}{(\frac{n}{2} - k) \frac{n}{2}}, \quad \forall \Lambda \in \mathcal{T}, \quad 1 \leq k \leq \frac{n}{2} - 1. \end{aligned} \quad (47)$$

The maximum load reaches  $r + \frac{kr}{(\frac{n}{2}-k)^{\frac{3}{2}}}$  when all edge switches in  $p$  send inter-pod traffic at their maximum rates  $\frac{n}{2}r$ , and the total traffic destined for  $i'_3$  is  $\frac{n}{2}r$ , i.e.,  $\sum_{i \in \mathcal{N}_E^p} \lambda_{ii'_3} = \frac{n}{2}r$ . Note that the sources of traffic to  $i'_3$  can vary within  $p$ .

Considering all failure scenarios and the symmetry of the two routing phases, we know that the maximum load in both directions of each core link  $l$  is  $r + \frac{kr}{(\frac{n}{2}-k)^{\frac{3}{2}}}$  and the theorem follows immediately.  $\square$

### 5.3. Total link capacity

Similar to the case of VL2, the minimum required capacity in both directions of each link is the same on fat-tree. We thus model the fat-tree topology as an undirected graph. The total numbers of edge and core links are both  $n^3/4$ , as calculated in Section 3.2.1. Considering (7), (8), (25), and (31), the minimum total link capacity required on fat-tree is computed as

$$C(k) = \frac{n^3}{4} \left( 2r + \frac{kr}{\frac{n}{2}-k} + \frac{kr}{(\frac{n}{2}-k)^{\frac{3}{2}}} \right), \quad \forall k \in \mathbb{Z}, \quad 0 \leq k \leq \frac{n}{2} - 1. \quad (48)$$

## 6. Numerical results

In this section, we evaluate the minimum capacity required per link on VL2 and fat-tree as well as the minimum total link capacity required. We assume that the maximum rate of each server NIC is 1 Gb/s, i.e.,  $r = 1$  Gb/s. For VL2, we consider  $n_s$  to be 20 and/or 80, the typical numbers of servers supported by a ToR switch [17].

### 6.1. VL2

Fig. 11 shows the minimum capacity requirement on each individual link, where  $n_s$  is set to 20. We see that for  $k \geq 1$ , the minimum capacity on edge links is

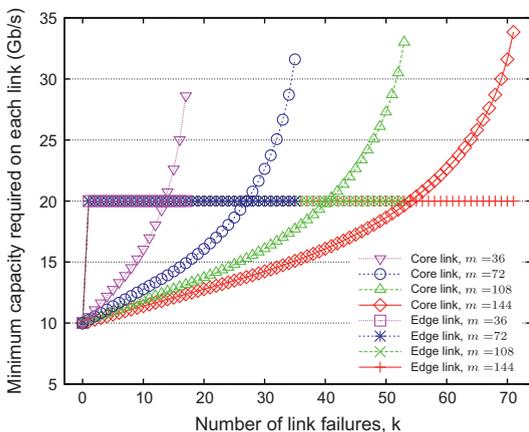


Fig. 11. Minimum capacity required per link on VL2 when  $n_s = 20$ .

independent of  $k$  and  $m$ , i.e., the number of link failures and the port count of aggregation and core switches. This property is dictated by (9). For core links, the capacity increase with  $k$  is dictated by (21) and (22) and thus follows the same trend illustrated in Fig. 8. For a common  $k$ , we see that less capacity is required per core link with larger  $m$  and that the capacity gap between different  $m$  increases as  $k$  gets larger. Because the link capacity increases with  $k$ , there can be a point from which the required capacity exceeds the speed that can be provided by a single port of a commodity switch. In such cases, we can build the link capacity by bundling two or more ports of a commodity switch to match the required capacity.

In Fig. 12, we show how the total link capacity increases with the total number of hosted servers (i.e., the size of a datacenter). Note that given  $n_s$  and  $m$ , the total number of servers supported by VL2 is  $\frac{m^2}{4}n_s$ . We present results for  $n_s = 20$  and  $n_s = 80$  in Figs. 12(a) and 12(b), respectively. For each  $n_s$ , we increase  $m$  to scale the size of a datacenter from tens of thousands of servers to hundreds of thousands of servers. We consider a selected number of  $k$ . Because the value range of  $k$  is dependent on  $m$ , we choose  $k$  to be proportional to  $m$  in most cases, i.e.,  $k = 0, \frac{m}{8}, \frac{m}{4}$ , and  $\frac{3m}{8}$ . We also consider  $k = 1$  and  $k = \frac{m}{2} - 1$ . We see that for all cases, the total link capacity increases linearly with the total number of supported servers, which suggests that the slope of the curves is constant. In the case of  $0 \leq k \leq \frac{m}{6}$  (i.e.,  $k = 0, 1$ , and  $\frac{m}{8}$ ), the explanation is straightforward. Dividing the total link capacity by the total number of supported servers, we obtain the ratio  $2r$  for  $k = 0$  and  $3r + \frac{2kr}{m}$  for  $1 \leq k \leq \frac{m}{6}$ . This immediately explains the linear capacity increase for  $k = 0$  and for  $1 \leq k \leq \frac{m}{6}$  with values in proportion to  $m$  (i.e.,  $k = \frac{m}{8}$ ), where the ratio becomes constant. For  $k = 1$ , the ratio changes with  $m$  due to the term  $\frac{2r}{m}$ . However, because  $m$  is large in the datacenter context, the impact of the term is negligible. Thus, the capacity growth for  $k = 1$  can be approximated as linear in relation to the total number of supported servers as well.

We also observe in Fig. 12 that given the total number of supported servers, the capacity gap between  $k = 0$  and  $k = 1$  is large. This is because the capacity on all edge links is doubled when we move from the case of no failures in (3) to the case of single link failures in (9). This part of the capacity remains constant among all  $k \geq 1$ . For  $k \geq 1$ , the capacity increase comes from core links. In Fig. 12, we observe that for a given total number of servers, the capacity increase becomes larger gracefully when  $k$  is incremented by an almost constant step of  $\frac{m}{8}$  starting from  $k = 1$ . This observation is validated in Fig. 13, which plots the total link capacity against the number of link failures  $k$ . Fig. 13 shows clearly that for a given total number of servers, the total capacity increases linearly with  $k$  when  $k$  is within  $1 \leq k \leq \frac{m}{6}$  and increases super-linearly with  $k$  when  $k$  is in the range  $\frac{m}{6} < k \leq \frac{m}{2} - 1$ . This immediately accounts for the larger capacity increase with larger  $k$  at near-regular intervals observed in Fig. 12. The increasing trend observed in Fig. 13 is dictated by the minimum capacity requirement on core links given in (21) and (22). Accordingly, for  $k \geq 1$ , all curves take a similar shape to the one illustrated in Fig. 8.

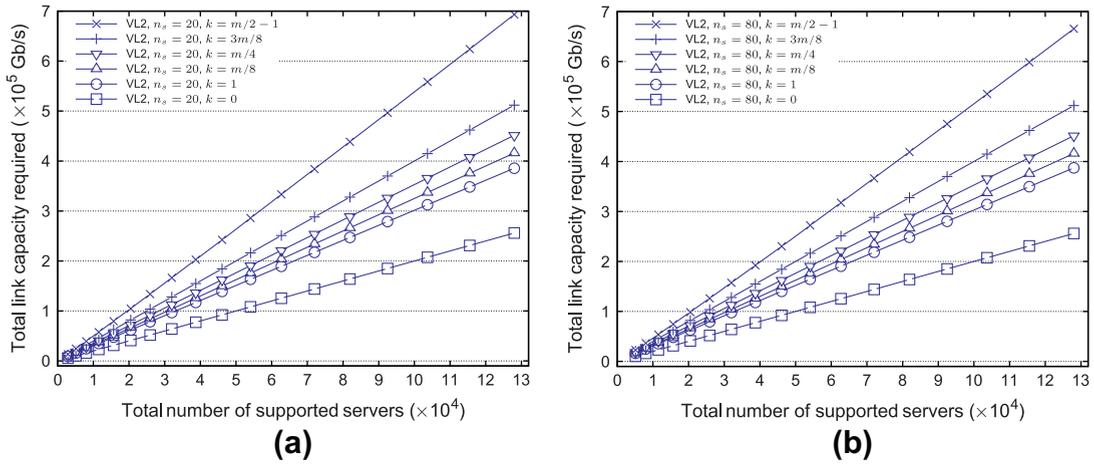


Fig. 12. Total link capacity required on VL2. (a)  $n_s = 20$ . (b)  $n_s = 80$ .

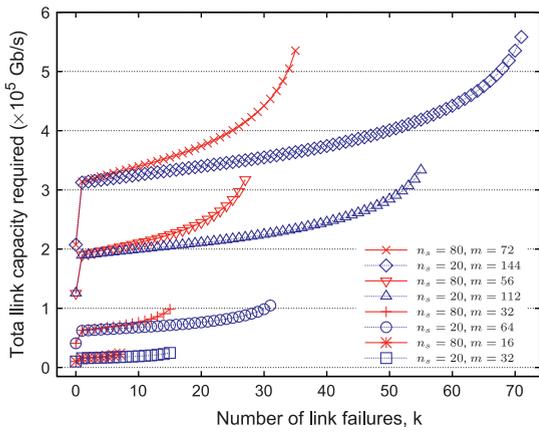


Fig. 13. Total link capacity required on VL2.

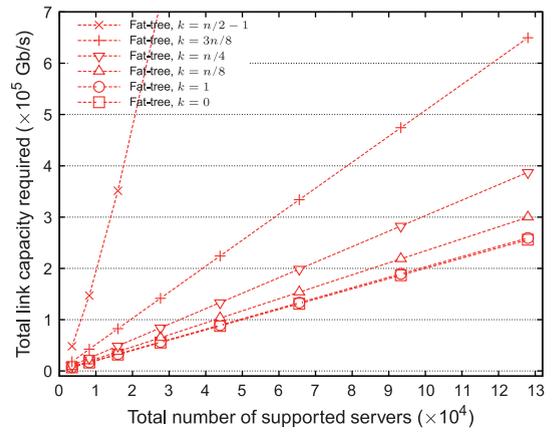


Fig. 15. Total link capacity required on fat-tree.

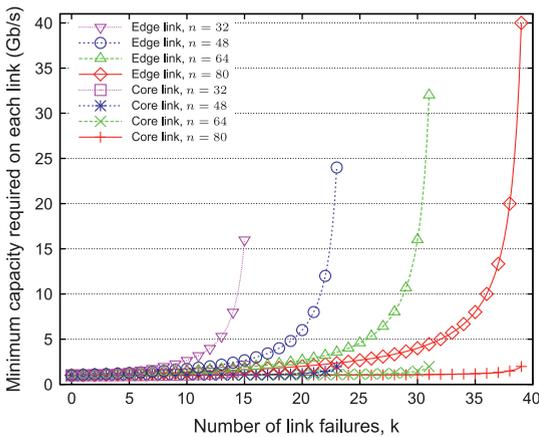


Fig. 14. Minimum capacity required per link on fat-tree.

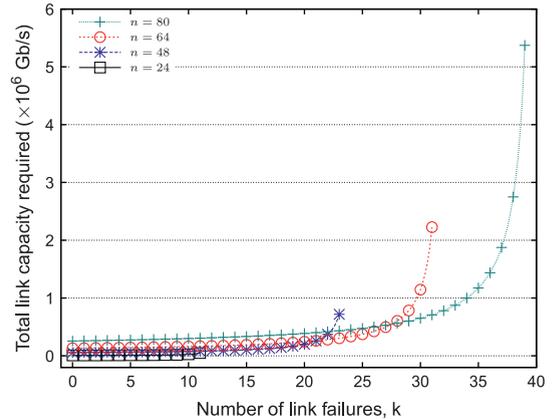


Fig. 16. Total link capacity required on fat-tree.

Fig. 13 also compares the total link capacity between  $n_s = 20$  and  $n_s = 80$  given that the total number of supported servers is the same. We see that for a common

$k \geq 1$ , the link capacity with  $n_s = 20$  is smaller than with  $n_s = 80$ , and the capacity gap becomes larger as  $k$  gets larger or the total number of servers increases.

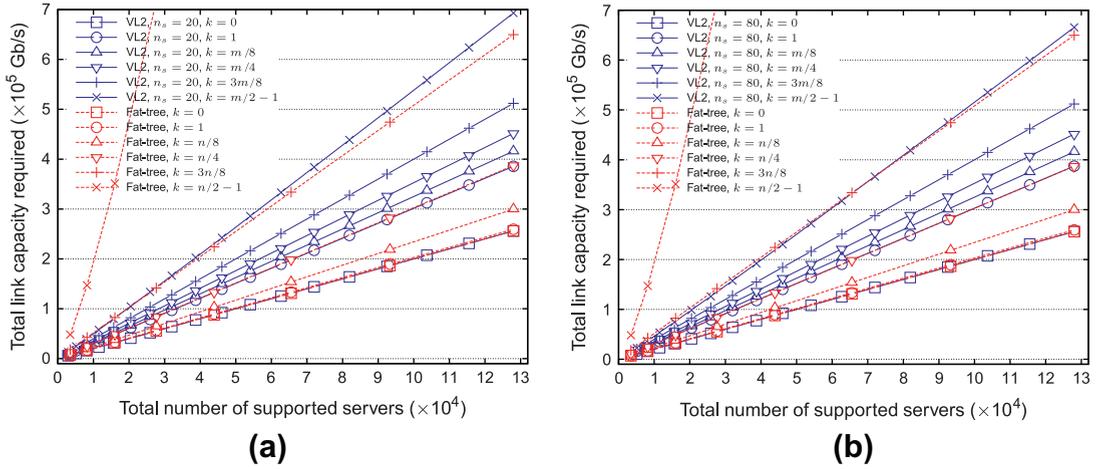


Fig. 17. Total link capacity required on VL2 and fat-tree. (a)  $n_s = 20$ . (b)  $n_s = 80$ .

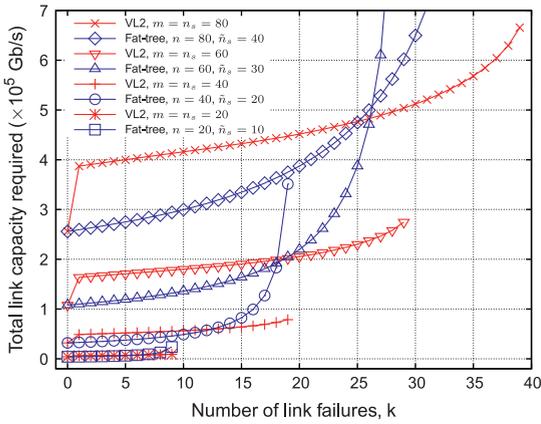


Fig. 18. Total link capacity comparison between VL2 and fat-tree.

### 6.2. Fat-tree

Fig. 14 shows the minimum capacity requirement per link. We see that given  $n$ , a core link requires less capacity than an edge link for all  $k \geq 1$ . This is because the path diversity for inter-pod traffic is  $\frac{n}{2}$  times as much as that for intra-pod traffic to significantly reduce the capacity requirement on core links in the event of link failures. Moreover, we observe that given a common  $k$ , less capacity is required per edge/core link for larger  $n$ , which shows the benefit of VLB. With increasing  $n$ , more alternative paths are available. The spare capacity allocated on each link is therefore reduced. Similar to the case discussed in VL2, for a link capacity that goes beyond the speed a single port of a commodity switch can provide, we can achieve it by using several ports of a commodity switch that work together to obtain the required capacity.

Similar to the case of VL2, we study how the total link capacity changes with the total number of supported servers. The curves are shown in Fig. 15. Note that for fat-tree with  $n$ -port switches, the total number of supported serv-

ers is  $\tilde{n}_s \cdot \frac{n}{2} \cdot n = \frac{n^3}{4}$ . We choose  $k$  to be in proportion to the switch port count in most cases, similar to the case of VL2. We observe that other than  $k = \frac{n}{2} - 1$ , the total link capacity increases linearly with the total number of servers. In general, the ratio between the total link capacity and the total number of servers is  $2r + \frac{kr}{\frac{n}{2}-k} + \frac{kr}{(\frac{n}{2}-k)^2}$ , which is dependent on  $n$ . When  $k$  is 0 or in proportion to  $n$ , the ratio becomes constant. For  $k = 1$ , the ratio can be safely approximated as constant based on the fact that  $n$  is large. Thus, in all cases other than  $k = \frac{n}{2} - 1$ , the ratio turns out to be constant, which explains the linear capacity increase with the total number of supported servers.

Given the total number of supported servers, we also observe in Fig. 15 that when  $k$  progresses in near-constant steps (i.e.,  $\approx \frac{n}{8}$ ) starting from  $k = 1$ , the total capacity increases much more sharply as  $k$  gets larger, showing an “exponential-like” increase trend. This trend is clearly captured in Fig. 16. In particular, all curves are very flat for small values of  $k$ , which indicates that a small extra capacity investment can support full bandwidth communication to the regime of multiple link failures. For example, in the case of  $n = 64$  (which supports 65,536 servers), by allocating 10% extra capacity compared to  $k = 0$ , the network can guarantee full bandwidth communication in the presence of 5 arbitrary link failures.

### 6.3. Capacity comparison between VL2 and fat-tree

We compare the total link capacity between VL2 and fat-tree for the same total number of supported servers. To this end, we combine the results in Figs. 12 and 15 and replot them in Fig. 17. In Fig. 17, it is interesting to observe that the curve of  $k = 1$  for VL2 overlaps with the curve of  $k = \frac{n}{4}$  for fat-tree. The observation is general: given the same total number of supported servers, the total link capacity required on VL2 for  $k = 1$  is approximately the same as on fat-tree for  $k = \frac{n}{4}$ . It immediately follows that for  $1 \leq k \leq \frac{n}{4}$ , fat-tree outperforms VL2 in terms of the total link capacity. We provide a proof of this observation in Appendix D.

The graceful capacity growth of VL2 with  $k \geq 1$  and the “exponential-like” capacity increase on fat-tree suggest a cross-point between the two curves, as indicated in Fig. 17. Clearly, the cross-point is no smaller than  $\frac{n}{4}$ , based on the observation above. We find the exact point in Fig. 18, which plots the total link capacity against the number of link failures. We set  $n_s = m = n$  so that both networks support the same total number of servers and are with the same range of  $k$ . We see that for  $n_s = m = n = 20, 40, 60,$  and  $80$ , the total link capacity of fat-tree is smaller than the total link capacity of VL2 when  $k$  is no greater than 6, 12, 18, and 25, respectively. In all cases, the value of the cross-point is not significantly larger than  $\frac{n}{4}$ .

## 7. Conclusion

We studied the capacity allocation problem in datacenter networks, where VLB is employed to handle highly variable traffic. Our design goal is to guarantee full bandwidth communication among all servers, for all valid traffic matrices, and under  $k$  arbitrary link failures. From the connectivity viewpoint,  $k$  is supported up to  $\frac{n}{2} - 1$  on fat-tree, whereas  $k$  is considered up to  $\frac{n}{2} - 1$  on VL2 for failure scenarios that do not partition the topology. We also proposed a mechanism to run VLB over fat-tree. In this context, we derived the minimum link capacity required on both topologies, where edge and core links are considered separately. Based on the results, we first evaluated the minimum total link capacity required on each individual topology and characterized the capacity increase trend with  $k$  and with the total number of supported servers. Then, we compared the total link capacity between the two topologies. We find that given the same total number of supported servers, there exists a turning point beyond which VL2 is better in terms of the total capacity requirement due to the sharp capacity increase on fat-tree in this regime.

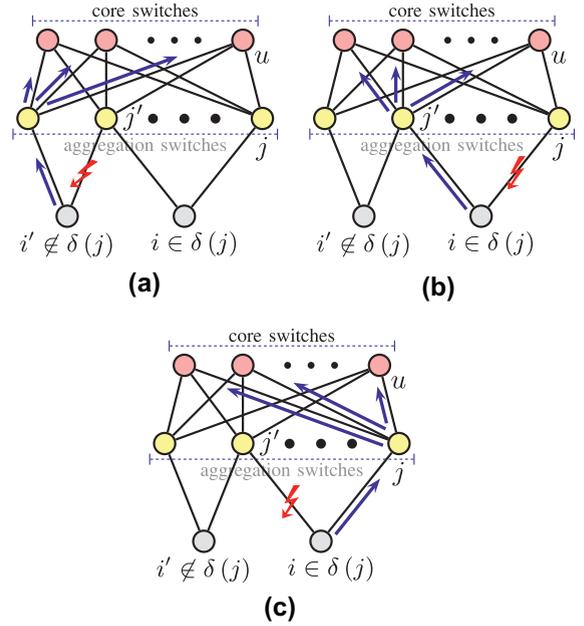
Future work includes two topics. The first one is capacity allocation for full bandwidth communication under node failures and under general shared risk link group failures. Note that a node failure is equivalent to the failure of all its incident links; thus, even a single node failure goes beyond the value range of  $k$  considered in this paper. The second one is capacity allocation for full bandwidth communication with heterogeneous ingress/egress capacity at edge switches.

## Acknowledgment

The authors would like to thank anonymous reviewers for their valuable comments to improve the technical quality of the paper, and Editor-in-Chief Dr. Harry Rudin for his careful reading that ensures the language quality of the paper.

## Appendix A. Proof of Proposition 1

In general, failures on edge links can occur at three types of location with respect to aggregation switch  $j$  and edge switches in  $\delta(j)$ , as shown in Fig. A.19: (1) links



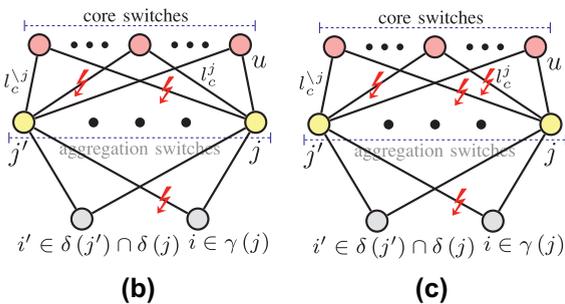
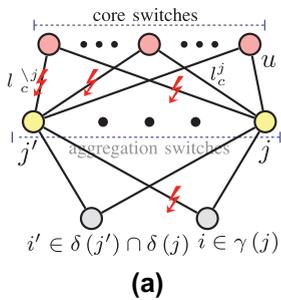
**Fig. A.19.** Failures on edge links with respect to aggregation switch  $j$  and edge switches in set  $\delta(j)$ . (a) Case 1: a failed edge link is incident to an edge switch not connected to  $j$ . (b) Case 2: a failed edge link is incident to  $j$ . (c) Case 3: a failed edge link is not incident to  $j$  but is incident to an edge switch connected to  $j$ .

incident to an edge switch not connected to  $j$  (and thus not in  $\delta(j)$ ), e.g., link  $(i', j')$ ; (2) links incident to  $j$  (and thus also incident to an edge switch in  $\delta(j)$ ), e.g., link  $(i, j)$ ; and (3) links not incident to  $j$  but incident to an edge switch connected to  $j$  (i.e., an edge switch in  $\delta(j)$ ), e.g., link  $(i, j')$ . We study how each failure case affects the traffic load on link  $(j, u)$ . Recall that link  $(j, u)$  carries one and only one path from each edge switch in  $\delta(j)$  to the core layer (see the VLB described in Section 3.1.3). In case 1, as the edge switch is not connected to  $j$ , a failed link does not affect the load on link  $(j, u)$ . In case 2, a failed link reduces the traffic load on link  $(j, u)$ . Specifically, if a failed link is incident to edge switch  $i$  in  $\delta(j)$  at the other end, it removes the traffic originating at  $i$  from the traffic load on link  $(j, u)$ . In case 3, let  $j'$  denote the aggregation switch incident to a failed link. After the failure event, the edge switch incident to the failed link directs all traffic that originally went through  $j'$  towards aggregation switch  $j$ , which introduces extra load on link  $(j, u)$ . Considering all three cases, we can find that to generate the maximum traffic load on link  $(j, u)$ , all  $k_e$  failures on edge links should be of case 3. Note that the above discussion is true regardless of the failure occurrence on core links and thus holds for arbitrary failure locations on the core links.

## Appendix B. Proof of Proposition 2

Clearly, core link  $(j, u)$  does not fail because otherwise there will be no traffic load on it. Other than link  $(j, u)$ , core links can fail at two types of location with respect to aggregation switch  $j$ , as shown in Fig. B.20(a): (1) links that are

incident to  $j$  and (2) links that are not. Accordingly, we denote failed core links by set  $\mathcal{F}_C$ , where core links incident to aggregation switch  $j$  are denoted by set  $\mathcal{F}_C^j$ , while core links incident to aggregation switches other than  $j$  are denoted by set  $\mathcal{F}_C^{\setminus j}$ . We have  $|\mathcal{F}_C| = k_c$ ,  $\mathcal{F}_C^j \cap \mathcal{F}_C^{\setminus j} = \emptyset$ , and  $\mathcal{F}_C = \mathcal{F}_C^j \cup \mathcal{F}_C^{\setminus j}$ . We show that given any valid traffic matrix, the maximum load on link  $(j, u)$  can be achieved when all  $k_c$  failed core links are incident to aggregation switch  $j$ . To this end, we compare the impact of the two failure location cases enumerated above on the traffic load on link  $(j, u)$ . Note that in the failure setting considered for edge links, which ensures that at most one edge link incident to an edge switch fails, the failure locations on core links can be chosen arbitrarily without disconnecting the topology. In other words, there exists at least one operating path from any edge switch to the core layer. Specifically, for  $k$  failed links, the minimum number of operating paths from any edge switch  $i$  to the core layer is  $\frac{m}{2} - (k - 1)$ , which occurs when one failure is on one edge link incident to  $i$  and the rest  $k - 1$  failures are on core links incident to the aggregation switch to which edge switch  $i$  is connected via the other (operating) edge link (e.g., aggregation switch  $j$  in Fig. B.21). We illustrate such a failure scenario in Fig. B.21, where the failed edge link disrupts half the paths (i.e.,  $\frac{m}{2}$  paths) from  $i$  to the core layer, while each failed core link disrupts one additional path from  $i$  to the core layer. Hence, the number of remaining operating paths from  $i$  to the core layer is  $m - [\frac{m}{2} + (k - 1)] = \frac{m}{2} - (k - 1)$  in the worst case. Considering  $k \leq \frac{m}{2} - 1$ , we have  $\frac{m}{2} - (k - 1) \geq \frac{m}{2} - (\frac{m}{2} - 1 - 1) = 2$ , i.e., the number of remaining operating paths from any edge switch  $i$  is in fact lower bounded by 2.



**Fig. B.20.** Failures on core links. (a) Failure setting  $\mathcal{F}_C$ , where some failed core links are incident to  $j$  while the others are not. (b) Failure setting  $\mathcal{F}_C \setminus \{l_c^j\}$ . (c) Failure setting  $\mathcal{F}_C \cup \{l_c^j\} \setminus \{l_c^{\setminus j}\}$ .

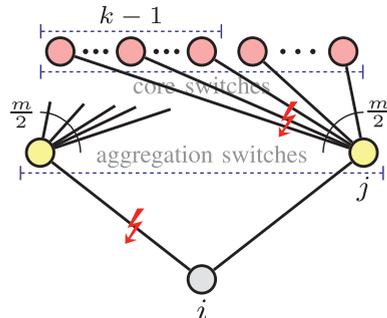
Given  $k_c$  failed core links, whose locations are chosen arbitrarily, a failed core link can be either incident to aggregation switch  $j$  or not. If  $\mathcal{F}_C^j \neq \emptyset$ , i.e., not all failed core links are incident to  $j$ , then  $|\mathcal{F}_C^j| < |\mathcal{F}_C| = k_c \leq k \leq \frac{m}{2} - 1$  or, simply,  $|\mathcal{F}_C^j| < \frac{m}{2} - 1$ , which implies that there exists a core link incident to  $j$  (other than link  $(j, u)$ ) that is up. We denote this link by  $l_c^j$ , as illustrated in Fig. B.20(a). We compare the load increase on link  $(j, u)$  between a failure on link  $l_c^j$  and on any link  $l_c^{\setminus j}$  in  $\mathcal{F}_C^{\setminus j}$ . By “load increase”, we refer to the baseline load to be incurred in failure setting  $\mathcal{F}_C \setminus \{l_c^{\setminus j}\}$ , where the “failed” core link  $l_c^{\setminus j}$  is assumed to be up with its impact removed, while all other link failures stay the same, as illustrated in Fig. B.20(b). In failure setting  $\mathcal{F}_C \setminus \{l_c^{\setminus j}\}$ , let  $R_i$  denote the number of remaining operating paths from edge switch  $i$  ( $\forall i \in \mathcal{N}_E$ ). Because the total number of failed links in the network is reduced to  $k - 1$ , it follows from the result in  $k$  failures presented above that  $R_i \geq 3$  for any edge switch  $i$ .

In the case that link  $l_c^j$  fails over failure setting  $\mathcal{F}_C \setminus \{l_c^j\}$ , as illustrated in Fig. B.20(c), as all edge switches in set  $\delta(j)$  remain connected to  $j$ , a failure on link  $l_c^j$  affects all edge switches in  $\delta(j)$ . Specifically, it disrupts one and only one of the  $R_i$  operating paths from edge switch  $i$  in  $\delta(j)$ , and thus  $\frac{1}{R_i}$  the amount of traffic from  $i$ . For each  $i$  in  $\delta(j)$ , the traffic carried by the failed path is evenly assigned to the remaining  $R_i - 1$  operating paths from  $i$ . Because  $R_i \geq 3$ ,  $\forall i \in \mathcal{N}_E$ , we have  $R_i - 1 \geq 2 > 0$ ,  $\forall i \in \delta(j)$ . Considering that link  $(j, u)$  carries one and only one remaining operating path for each edge switch in  $\delta(j)$ , the load increase on link  $(j, u)$  under traffic matrix  $\Lambda$  can be expressed as

$$\Delta\psi_{(j,u)}(l_c^j; \Lambda) = \sum_{i \in \delta(j)} \frac{\sum_{i' \in \mathcal{N}_E, i' \neq i} \lambda_{i'i}}{R_i - 1}, \quad \forall j \in \mathcal{N}_A, u \in \mathcal{N}_C, \Lambda \in \mathcal{T}.$$

Note that all terms on the right side of the equality are valid due to the fact that  $R_i - 1 > 0$ ,  $\forall i \in \delta(j)$ .

In the second case, where link  $l_c^{\setminus j}$  fails over failure setting  $\mathcal{F}_C \setminus \{l_c^{\setminus j}\}$ , as illustrated in Fig. B.20(a), let  $j' \neq j$  denote the aggregation switch to which link  $l_c^{\setminus j}$  is incident. Unlike the first case, only edge switches that are connected to



**Fig. B.21.** Failure scenario that produces the minimum number of operating paths from edge switch  $i$  among  $k$  arbitrary link failures that do not partition the topology.

both  $j'$  and  $j$  affect the traffic load on link  $(j, u)$ . Note that due to the failures on edge links, not all edge switches in set  $\delta(j')$  remain connected to  $j'$ . Hence, the edge switches with impact are limited to the set  $\delta(j') \cap \delta(j) \setminus \gamma(j)$ . For each edge switch in  $\delta(j') \cap \delta(j) \setminus \gamma(j)$ , the impact of additional failure is exactly the same as in the first case discussed above. Specifically, failure on link  $l_c^j$  disrupts one and only one of the  $R_i$  operating paths from edge switch  $i$  in  $\delta(j') \cap \delta(j) \setminus \gamma(j)$ , and thus  $\frac{1}{R_i}$  the amount of traffic originating from  $i$ . For each  $i$  in  $\delta(j') \cap \delta(j) \setminus \gamma(j)$ , the traffic carried by the failed path is evenly assigned to the remaining  $R_i - 1$  operating paths from  $i$ , where  $R_i - 1 \geq 2 > 0$ ,  $\forall i \in \delta(j') \cap \delta(j) \setminus \gamma(j)$ . Moreover, link  $(j, u)$  is on one and only one remaining operating path for each edge switch in  $\delta(j') \cap \delta(j) \setminus \gamma(j)$  to carry the disrupted traffic. Hence, the load increase on link  $(j, u)$  under traffic matrix  $\Lambda$  can be written as

$$\Delta\psi_{(j,u)}(l_c^j; \Lambda) = \sum_{i \in \delta(j') \cap \delta(j) \setminus \gamma(j)} \frac{\sum_{i' \in \mathcal{N}_E, i' \neq i} \lambda_{i'j}}{R_i - 1}, \quad \forall j \in \mathcal{N}_A, u \in \mathcal{N}_C, l_c^j \in \mathcal{F}_C^j, \Lambda \in \mathcal{T},$$

where all terms on the right side of the equality are valid because  $R_i - 1 > 0$ ,  $\forall i \in \delta(j') \cap \delta(j) \setminus \gamma(j)$ .

Now, we compare the load increase on link  $(j, u)$  between the above two incremental failure cases. Because  $\delta(j') \cap \delta(j) \setminus \gamma(j) \subseteq \delta(j)$ , we have

$$\Delta\psi_{(j,u)}(l_c^j; \Lambda) - \Delta\psi_{(j,u)}(l_c^j; \Lambda) \geq 0, \quad \forall j \in \mathcal{N}_A, u \in \mathcal{N}_C, l_c^j \in \mathcal{F}_C^j, \Lambda \in \mathcal{T}. \quad (\text{B.1})$$

This statement suggests that given an arbitrary set  $\mathcal{F}_C$  on failed core links, if there exists a failed core link  $l_c^j$  that is not incident to aggregation switch  $j$ , then by moving the link failure from  $l_c^j$  to an operating link  $l_c^j$  incident to  $j$ , the load on link  $(j, u)$  does not decrease for any valid traffic matrix and is most likely to increase. Because we do not make any assumption about the core links in set  $\mathcal{F}_C$ , if in the new failure setting there exists a failed core link  $l_c^j$  not incident to aggregation switch  $j$ , the above discussion (and thus Eq. (B.1)) still holds. This allows us to repeat the operation process (moving failure on  $l_c^j$  to an operating link incident to  $j$ ) without decreasing the load on link  $(j, u)$  until all failed core links become incident to aggregation switch  $j$ , i.e.,  $\mathcal{F}_C^j = \emptyset$  and  $\mathcal{F}_C = \mathcal{F}_C^j$ . It follows that the load on link  $(j, u)$  with all failed core links incident to  $j$  (i.e.,  $\mathcal{F}_C^j = \emptyset$ ) is no smaller than the load with any core link failure not incident to  $j$  (i.e.,  $\mathcal{F}_C^j \neq \emptyset$ ).

### Appendix C. Proof of Lemma 1

The problem in (11)–(13) is a discrete optimization problem that maximizes the function  $f(k_c, k)$  over the single integer variable  $k_c$ , which takes values in the discrete set  $\mathcal{K}(k) = \{0, 1, \dots, k\}$ . We first relax  $k_c$  to a real variable and find the optimal point of  $k_c$  at which the value of function  $f(k_c, k)$  is maximized on the interval  $k_c \in [0, k]$ . We denote this optimal point of  $k_c$  by  $\tilde{k}_c^*$ . Next, we restore the

integer constraint on  $k_c$  and find the optimal solution  $k_c^*$  to the original problem in (11)–(13) as well as the optimal value of the problem.

To find the optimal point  $\tilde{k}_c^*$  in the interval  $[0, k]$ , we use the method of observing the sign change of the derivative function [28]. Specifically, the derivative function of  $f(k_c, k)$  (with respect to variable  $k_c$ ) is given by

$$f'(k_c, k) = \frac{df}{dk_c} = \frac{mg(k_c, k)}{\left(\frac{m}{2} - k_c\right)(m - k_c)},$$

where

$$g(k_c, k) = (k_c)^2 - \left(\frac{m}{2} + k\right)k_c + \frac{3}{4}mk - \frac{m^2}{8}.$$

Because  $k_c \leq k \leq \frac{m}{2} - 1 < \frac{m}{2}$ , we have  $\frac{m}{2} - k_c > 0$  and  $m - k_c > 0$ . Hence, the sign of  $f'(k_c, k)$  is consistent with the sign of  $g(k_c, k)$ . Below, we study function  $g(k_c, k)$  to determine the sign of  $f'(k_c, k)$ .

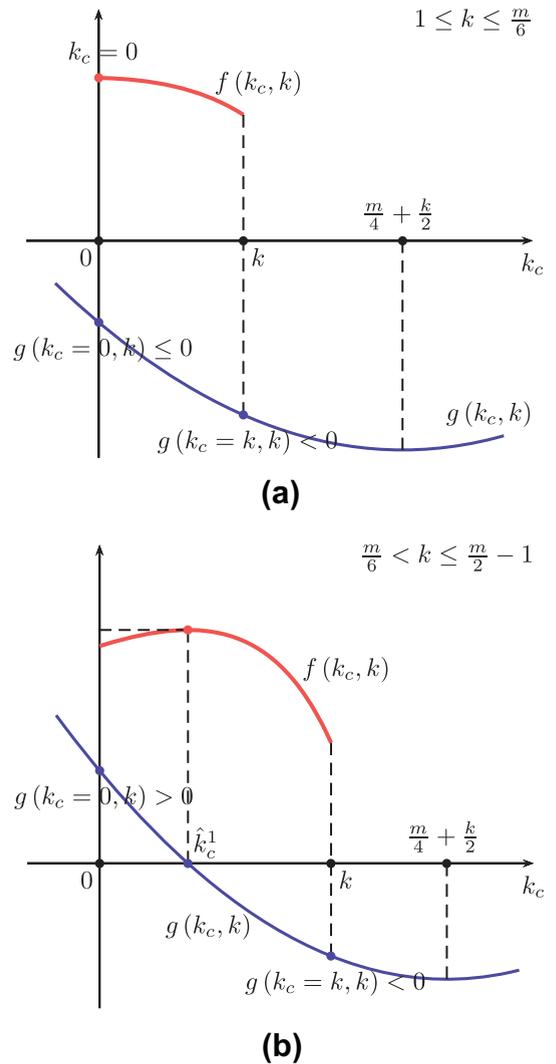


Fig. C.22. Functions  $f(k_c, k)$  and  $g(k_c, k)$  under different values of  $k$ . (a)  $1 \leq k \leq \frac{m}{6}$ . (b)  $\frac{m}{6} < k \leq \frac{m}{2} - 1$ .

Function  $g(k_c, k)$  is a parabola that opens upward. The vertical axis of symmetry is at  $k_c = \frac{m}{4} + \frac{k}{2}$ . Because  $k - (\frac{m}{4} + \frac{k}{2}) = \frac{k}{2} - \frac{m}{4} \leq \frac{1}{2}(\frac{m}{2} - 1) - \frac{m}{4} = -\frac{1}{2} < 0$ , we have

$$k < \frac{m}{4} + \frac{k}{2}, \quad \forall k \leq \frac{m}{2} - 1, \quad (\text{C.1})$$

which implies that the value interval of  $k_c$  (i.e.,  $[0, k]$ ) is on the left side of the vertical axis of symmetry, as shown in Fig. C.22. Considering the fact that the parabola opens upward, function  $g(k_c, k)$  is thus strictly monotonically decreasing on the closed interval  $[0, k]$ . This allows us to characterize the sign of  $g(k_c, k)$  in general by checking the two ends of the interval, i.e.,  $k_c = 0$  and  $k_c = k$ . At the right end, we obtain

$$g(k_c = k, k) = \frac{m}{4} \left( k - \frac{m}{2} \right) \leq \frac{m}{4} \left( \frac{m}{2} - 1 - \frac{m}{2} \right) < 0.$$

At the left end, we have

$$g(k_c = 0, k) = \frac{3}{8} m (6k - m) \begin{cases} \leq 0 & \text{if } 1 \leq k \leq \frac{m}{6}, \\ > 0 & \text{if } \frac{m}{6} < k \leq \frac{m}{2} - 1. \end{cases}$$

Consequently, when  $1 \leq k \leq \frac{m}{6}$ , the sign of  $f'(k_c, k)$  is “−” on the interval  $(0, k]$ . Accordingly, because function  $f(k_c, k)$  is continuous on the closed interval  $[0, k]$ ,  $f(k_c, k)$  is strictly monotonically decreasing on the interval  $[0, k]$ , as illustrated in Fig. C.22(a). Hence, at point  $\tilde{k}_c^* = 0$ , we obtain the maximum value of function  $f(k_c, k)$  on the interval  $[0, k]$ . The maximum value is given by

$$f(\tilde{k}_c^* = 0, k) = \frac{1}{2} + \frac{k}{m}, \quad \forall k \in \mathbb{Z}, \quad 1 \leq k \leq \frac{m}{6}. \quad (\text{C.2})$$

Conversely, when  $\frac{m}{6} < k \leq \frac{m}{2} - 1$ , the sign of the function  $g(k_c, k)$  changes from “+” at  $k_c = 0$  to “−” at  $k_c = k$ . Because the function  $g(k_c, k)$  is strictly monotonically decreasing on the interval  $[0, k]$ , as discussed above, it has one and one only real root in the interior of the interval, i.e., there exists  $\hat{k}_c \in (0, k)$  such that  $g(\hat{k}_c, k) = 0$ . Accordingly, and more specifically, the sign of the function  $f'(k_c, k)$  is “+” on the interval  $[0, \hat{k}_c)$  and changes to “−” on the interval  $(\hat{k}_c, k]$ , as shown in Fig. C.22(b). It follows from the method of sign change that the maximum value of the function  $f(k_c, k)$  on the interval  $[0, k]$  is obtained at point  $\tilde{k}_c^* = \hat{k}_c$ . To derive the explicit expression for  $\hat{k}_c$ , we solve the equation  $g(\hat{k}_c, k) = 0$ . We obtain two real roots  $\hat{k}_c^1$  and  $\hat{k}_c^2$  given, respectively, by

$$\hat{k}_c^1 = \frac{k}{2} + \frac{m}{4} - \frac{1}{4} [(3m - 2k)(m - 2k)]^{\frac{1}{2}}, \quad (\text{C.3})$$

and

$$\hat{k}_c^2 = \frac{k}{2} + \frac{m}{4} + \frac{1}{4} [(3m - 2k)(m - 2k)]^{\frac{1}{2}}. \quad (\text{C.4})$$

Because

$$\hat{k}_c^2 > \frac{k}{2} + \frac{m}{4} > k, \quad (\text{C.5})$$

where the last inequality follows from (C.1), we know immediately that  $\hat{k}_c^1$  is the root in the interior of the interval  $[0, k]$ , i.e.,  $\tilde{k}_c^* = \hat{k}_c^1 \in (0, k)$ . The corresponding maximum

value of  $f(k_c, k)$  on the interval  $[0, k]$  is thus  $f(\tilde{k}_c^* = \hat{k}_c^1, k)$ ,  $\forall \frac{m}{6} < k \leq \frac{m}{2} - 1$ .

So far, we have found the optimal point  $\tilde{k}_c^*$  at which the value of  $f(k_c, k)$  on the continuous interval  $[0, k]$  is maximized. Next, we restore the integer constraint on  $k_c$  by enforcing  $k_c \in \mathcal{K}(k)$  and derive the optimal solution  $k_c^*$  to the original problem in (11)–(13) as well as the optimal value of the problem. We discuss the cases  $1 \leq k \leq \frac{m}{6}$  and  $\frac{m}{6} < k \leq \frac{m}{2} - 1$ .

When  $1 \leq k \leq \frac{m}{6}$ , because  $\tilde{k}_c^*$  takes the integer value 0, which is also an element in set  $\mathcal{K}(k)$ , it immediately follows that  $k_c^* = 0$  is the optimal solution to the original problem. The optimal value of the original problem is (C.2) scaled by  $n_s r$ .

When  $\frac{m}{6} < k \leq \frac{m}{2} - 1$ ,  $\tilde{k}_c^*$  takes the value  $\hat{k}_c^1$ , which is not necessarily an integer. Because the sign of the function  $f'(k_c, k)$  is “+” on the interval  $[0, \hat{k}_c^1)$  and becomes “−” on the interval  $(\hat{k}_c^1, k]$ , the function  $f(k_c, k)$  is strictly monotonically increasing on the interval  $[0, \hat{k}_c^1]$  and is strictly monotonically decreasing on the interval  $[\hat{k}_c^1, k]$ . Consequently, the integer point that maximizes  $f(k_c, k)$  on the interval  $[0, k]$  is among the points nearest to  $\hat{k}_c^1$ , i.e.,  $\lfloor \hat{k}_c^1 \rfloor$  and  $\lceil \hat{k}_c^1 \rceil$ . Note that because  $\hat{k}_c^1$  is an interior point of the interval  $[0, k]$ , i.e.,  $\hat{k}_c^1 \in (0, k)$ , we have  $0 < \lfloor \hat{k}_c^1 \rfloor \leq \lceil \hat{k}_c^1 \rceil \leq k$ . In other words, both  $\lfloor \hat{k}_c^1 \rfloor$  and  $\lceil \hat{k}_c^1 \rceil$  are in the closed interval  $[0, k]$  and thus are also elements in set  $\mathcal{K}(k)$ . Therefore, the optimal solution to the original problem is given by

$$k_c^* = \operatorname{argmax}_{k_c \in \{\lfloor \hat{k}_c^1 \rfloor, \lceil \hat{k}_c^1 \rceil\}} f(k_c, k), \quad \forall k \in \mathbb{Z}, \quad \frac{m}{6} < k \leq \frac{m}{2} - 1.$$

The corresponding optimal value of the original problem is

$$n_s r \cdot \max_{k_c \in \{\lfloor \hat{k}_c^1 \rfloor, \lceil \hat{k}_c^1 \rceil\}} f(k_c, k), \quad \forall k \in \mathbb{Z}, \quad \frac{m}{6} < k \leq \frac{m}{2} - 1.$$

Combining the results of the two cases, we obtain the optimal value of the original problem for any given integer  $k$  with its value in the range  $1 \leq k \leq \frac{m}{2} - 1$ .

## Appendix D. Proof of observation in Section 6.3

For VL2 and fat-tree supporting the same total number of servers, we have  $\frac{m^2 n_s}{4} = \frac{n^3}{4}$ . On VL2, the total capacity required for  $k = 1$  is  $\frac{m^2 n_s}{4} (3r + \frac{2r}{m})$ . On fat-tree, the total required capacity for  $k = \frac{n}{4}$  is  $\frac{n^3}{4} (3r + \frac{2r}{n})$ . Dividing the total capacity on VL2 by the total capacity on fat-tree yields  $\frac{3 + \frac{2}{m}}{3 + \frac{2}{n}} \approx 1$ , where the approximate equality follows from the fact that both  $n$  and  $m$  are large integers in the datacenter context.

## References

- [1] F. Gens, New IDC IT Cloud Services Survey: Top Benefits and Challenges, December 2009. <<http://blogs.idc.com/ie/?p=730>>.
- [2] M. Armbrust et al., A view of cloud computing, *ACM Commun. ACM* 53 (4) (2010) 50–58.
- [3] A. Greenberg et al., VL2: a scalable and flexible data center network, in: Proceedings of ACM SIGCOMM, August 2009, pp. 51–62.

- [4] M. Al-Fares, A. Loukissas, A. Vahdat, A scalable, commodity data center network architecture, in: Proceedings of ACM SIGCOMM, August 2008, pp. 63–74.
- [5] L.A. Barroso, U. Hölzle, The datacenter as a computer: an introduction to the design of warehouse-scale machines, Morgan Claypool Publishers Synthesis Lect. Comput. Archit. 4 (1) (2009) 1–108.
- [6] C. Guo et al., BCube: a high performance, server-centric network architecture for modular data centers, in: Proceedings of ACM SIGCOMM, August 2009, pp. 63–74.
- [7] P. Gill, N. Jain, N. Nagappan, Understanding network failures in data centers: measurement, analysis, and implications, in: Proceedings of ACM SIGCOMM, August 2011, pp. 350–361.
- [8] R.N. Mysore et al., PortLand: a scalable fault-tolerant layer 2 data center network fabric, in: Proceedings of ACM SIGCOMM, August 2009, pp. 39–50.
- [9] W.J. Dally, B. Towles, Principles and Practices of Interconnection Networks, Morgan Kaufman, San Francisco, CA, 2004.
- [10] P. Bodík, I. Menache, M. Chowdhury, P. Mani, D.A. Maltz, I. Stoica, Surviving failures in bandwidth-constrained datacenters, in: Proceedings of ACM SIGCOMM, August 2012, pp. 431–442.
- [11] W.-L. Yeow, C. Westphal, U.C. Kozat, Designing and embedding reliable virtual infrastructures, ACM SIGCOMM Comput. Commun. Rev. 41 (2) (2011) 57–64.
- [12] B. Cully, G. Lefebvre, D. Meyer, M. Feeley, N. Hutchinson, A. Warfield, Remus: high availability via asynchronous virtual machine replication, in: Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI), April 2008, pp. 161–174.
- [13] Amazon, Amazon Elastic Compute Cloud (Amazon EC2) Features. <<http://aws.amazon.com/ec2/#features>>.
- [14] Cisco, Massively Scalable Data Center (MSDC) Design and Implementation Guide, January 2013. <[http://www.cisco.com/en/US/docs/solutions/Enterprise/Data\\_Center/MSDC/1.0/MSDC1.pdf](http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/MSDC/1.0/MSDC1.pdf)>.
- [15] T. Benson, A. Anand, A. Akella, M. Zhang, Understanding data center traffic characteristics, ACM SIGCOMM Comput. Commun. Rev. 40 (1) (2010) 92–99.
- [16] M. Alizadeh et al., Data center TCP (DCTCP), in: Proceedings of ACM SIGCOMM, September 2010, pp. 63–74.
- [17] A.R. Curtis, T. Carpenter, M. Elsheikh, A. López-Ortiz, S. Keshav, REWIRE: an optimization-based framework for unstructured data center network design, in: Proceedings of IEEE International Conference on Computer Communications (INFOCOM), March 2012, pp. 1116–1124.
- [18] R. Zhang-Shen, N. McKeown, Designing a predictable internet backbone network, in: Proceedings of ACM 3rd Workshop on Hot Topics in Networks (HotNets-III), November 2004.
- [19] M. Kodialam, T.V. Lakshman, S. Sengupta, Efficient and robust routing of highly variable traffic, in: Proceedings of ACM 3rd Workshop on Hot Topics in Networks (HotNets-III), November 2004.
- [20] C.J.S. Decusatis, A. Carranza, C.M. Decusatis, Communication within clouds: open standards and proprietary protocols for data center networking, IEEE Commun. Mag. 50 (9) (2012) 26–33.
- [21] J. Zhang, K. Zhu, H. Zang, N.S. Matloff, B. Mukherjee, Availability-aware provisioning strategies for differentiated protection services in wavelength-convertible WDM mesh networks, IEEE/ACM Trans. Netw. 15 (5) (2007) 1177–1190.
- [22] Y. Liu, D. Tipper, P. Siripongwutikorn, Approximating optimal spare capacity allocation by successive survivable routing, IEEE/ACM Trans. Netw. 13 (1) (2005) 198–211.
- [23] D.A. Schupke, R.G. Prinz, Capacity efficiency and restorability of path protection and rerouting in WDM networks subject to dual failures, Springer Photonic Netw. Commun. 8 (2) (2004) 191–207.
- [24] V.Y. Liu, D. Tipper, Spare capacity allocation using shared backup path protection for dual link failures, Elsevier Comput. Commun., September 2012. <http://dx.doi.org/10.1016/j.comcom.2012.09.007>.
- [25] J.M. Simmons, Optical Network Design and Planning, Springer, New York, NY, 2008.
- [26] K.S. Ho, K.W. Cheung, Generalized survivable network, IEEE/ACM Trans. Netw. 15 (4) (2007) 750–760.
- [27] C.E. Leiserson, Fat-trees: universal networks for hardware-efficient supercomputing, IEEE Trans. Comput. 34 (10) (1985) 892–901.

- [28] I.N. Bronshtein, K.A. Semendiaev, G. Musiol, H. Muehlig, Handbook of Mathematics, Springer-Verlag, New York, NY, 2007.
- [29] W. Ni, C. Huang, J. Wu, On capacity provisioning in datacenter networks for full bandwidth communication, in: Proceedings of IEEE High Performance Switching and Routing (HPSR), July 2013, pp. 62–67.



**Wenda Ni** received the B.Eng. (with excellence) and Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China in 2005 and 2010, respectively. He is currently a researcher with the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada. His research interests include reliable network design and analysis, passive optical networks, datacenter networking, and decomposition methods for large-scale network optimization. He was a semi-finalist in the 2010 Corning Outstanding

Student Paper Competition for his paper "Survivable mapping with maximal physical-layer failure-localization potential in IP over transparent optical networks" (10–12 papers out of over 430 student submissions), and a runner-up of the 2013 Fabio Neri Best Paper Award for his paper "Availability of survivable Valiant load balancing (VLB) networks over optical networks".



**Changcheng Huang** received his B. Eng. in 1985 and M. Eng. in 1988 both in Electronic Engineering from Tsinghua University, Beijing, China. He received a Ph.D. degree in Electrical Engineering from Carleton University, Ottawa, Canada in 1997. From 1996 to 1998, he worked for Nortel Networks, Ottawa, Canada where he was a systems engineering specialist. He was a systems engineer and network architect in the Optical Networking Group of Tellabs, Illinois, USA during the period of 1998–2000. Since July 2000, he has

been with the Department of Systems and Computer Engineering at Carleton University, Ottawa, Canada where he is currently a professor. He won the CFI new opportunity award for building an optical network laboratory in 2001. He was an associate editor of IEEE Communications Letters from 2004 to 2006.



**Jing Wu** obtained a B.Sc. degree in information science and technology in 1992, and a Ph.D. degree in systems engineering in 1997, both from Xian Jiao Tong University, China. He is now a Research Scientist at the Communications Research Centre Canada (Ottawa, Canada), an Agency of Industry Canada. Currently, he is also appointed as an Adjunct Professor at the University of Ottawa, School of Information Technology and Engineering. He has contributed over 70 conference and journal papers. He holds three patents on

Internet congestion control, and one patent on control plane failure recovery. His research interests include control and management of optical networks, protocols and algorithms in networking, optical network performance evaluation and optimization. He is a co-chair of the sub-committee on network architecture, management and applications of the Asia Communications and Photonics Exhibit and Conference (ACP) 2009 and 2010.