FC+: Near-optimal Deadlock-free Expander Data Center Networks

Xiao Zhang[1], Peirui Cao[1], Yongxi Lyu[1], Qizhou Zhang[1], Shizhen Zhao[1], Xinbing Wang[1], Chenghu Zhou[1,2]
[1]*Shanghai Jiao Tong university,* [2]*Chinese Academy of Sciences*

***Abstract*—Expander networks have gained attention as a cost-efficient alternative to expensive Clos networks in data centers. However, they face challenges with deadlocks caused by the widespread deployment of PFC-enabled RoCE networks. Unfortunately, current methods to address deadlocks in expander networks often come with drawbacks that either compromise performance or fail to completely eliminate deadlocks.**

After identifying path diversity as the performance bottleneck in FC (Flatten Clos), we present FC+ (Flatten Clos Plus), a topology-routing co-design to eliminate deadlocks and achieve near-optimal performance. Similar to FC, FC+ also maps its topology to a multi-layered virtual topology and performs up-down routing to eliminate deadlocks. Based on this, FC+ introduces 2 new designs that can effectively improve path diversity. First, FC+ adopts a non-uniform virtual multi-layer design, which greatly increases the number of deadlock-free paths. Second, FC+ uses deadlock-free K-Shortest Paths (DF-KSP) for routing, utilizing the path diversity better. We perform throughput evaluation under different traffic patterns. With 1 lossless priority, FC+ consistently outperforms FC and the performance enhancement reaches 1.4x to 2x under near-worst case. Another advantage of FC+ over FC is that FC+'s DF-KSP routing allows using more than 1 lossless priorities to further improve performance. Compared to Tagger, the state-of-the-art lossless priority management method to avoid deadlocks, FC+ reduces the number of lossless priorities from 3 to 2 in order to achieve near-optimal performance.

I. Introduction

As the demand for larger bandwidth increases due to growing traffic in data centers, the cost of building larger-scale Clos networks has become prohibitive [2], [4], [5], [29]. To reduce the cost, flattened expander topologies such as Jellyfish [23], Slimfly [3], Xpander [26], etc., have been proposed.

RDMA over Commodity Ethernet (RoCE) has gained significant traction in data centers, including Microsoft Azure [1] and Alibaba [7], driven by its high performance and low CPU overheads. Particularly, in Microsoft Azure, RDMA traffic constitutes around 70% of the total traffic and Priority-based Flow Control (PFC) remains a popular solution to avoid congestion packet losses [1]. However, the use of PFC in RoCE can potentially introduce deadlocks, which pose a challenge for expander networks due to the flattened topology. In expander topologies, K-shortest Paths (KSP) is commonly employed to achieve high throughput [23], [26]. However, KSP routing can introduce deadlocks into networks [30].

Existing approaches to handle deadlocks caused by KSP all fall short. (1) The Deadlock Recovery methods introduce significant latency when operate on control plane [15], [22]. Method operating on the data plane relies on specialized hardware and its deadlock resolution techniques may face challenges in handling concurrent deadlocks or result in packet loss [27]. (2) Lossy RDMA like Lossy RoCE [10], [16] impairs the performance of mice flows and requires hardware support such as Mellanox ConnectX-4 [14], [18]. Additionally, IWarp [21], another Lossy RDMA technology, exhibits poor performance due to its reliance on TCP for lossless delivery. (3) Lossless Priority Management [6], [13] typically requires a minimum of 3 lossless priorities, even with compression algorithms like Tagger [12]. However, RoCE only supports 2 or 3 lossless priorities [9].

Some deadlock avoidance routing approaches, such as FC [30] and EDST [24], [25], have also been proposed. However, these routing methods lead to performance degradation. So we are struggling to weigh the pros and cons of using KSP versus using deadlock avoidance routing.

Motivated by the design of virtual multi-layer topology and up-down routing in FC [30], we propose **Flattened Clos Plus (FC+)**, a novel topology-routing co-design free of deadlocks with near-optimal performance. Through our observations, we identify that the throughput losses in FC are primarily caused by the significant reduction in path diversity when handling deadlocks. As indicated in § II-C, only an average of 4 to 6 paths are used among the available paths for each switch pair in FC. In FC+, we introduce 2 new designs aimed at maximizing path diversity during deadlock handling:

1) We propose a non-uniform virtual multi-layer topology that involves creating maximum number of virtual layers and reducing the number of virtual switches. This innovative topology leads to the introduction of a much larger number of deadlock-free paths compared to FC.
2) We employ DF-KSP routing as a replacement for edge-disjoint routing in FC. With DF-KSP, we can choose K deadlock-free shortest paths, which provides the flexibility to select a specific number of paths based on requirements. Besides, another advantage of DF-KSP is that it allows us to use more than 1 lossless priority, while FC is limited to using only 1 due to the constraints of edge disjoint routing.

We evaluate FC+ against FC, equal-cost Clos, KSP for various traffic metrics (all-to-all, uniform random, near-worst case). With 1 lossless priority, FC+ consistently outperforms FC and equal-cost Clos across all three traffic patterns. Notably, under near-worst case, FC+ achieves a throughput that is 1.4x to 2x higher than FC. Besides, compared to the average

performance gap between FC and KSP, the gap between FC+ and KSP has narrowed. Under the all-to-all metric, it reduced from 17% to 9%. In the uniform random scenario, the gap decreased from 23% to 10%. And in the near-worst case, it dropped from 53% to 16%.

When utilizing 1 lossless priority, FC+ still exhibits a small performance gap compared to KSP. So we evaluate the minimum number of lossless priorities required for FC+ to attain near-optimal performance. The findings reveal that with just 2 lossless priorities, the average performance gap relative to KSP diminishes to less than 1% across all traffic metrics. Intriguingly, much like KSP, FC+ demonstrates the capacity to converge to the performance upper bound as the network's size escalates. This proves that FC+ can achieve near-optimal results with a mere 2 lossless priorities. When juxtaposed with Tagger [12] (the contemporary benchmark in lossless priority management for deadlock avoidance without sacrificing performance) , FC+ reduces the required number of lossless priorities from 3 to 2 for near-optimal performance.

II. Background and Motivation

Existing deadlock-handling methods for expander networks can be categorized into two classes. The first focuses on handling deadlocks caused by routing methods like KSP, while the second aims to design deadlock avoidance routing methods. Both of them cannot handle deadlocks efficiently without performance loss. **Therefore, based on our observation, we aim to develop a new deadlock avoidance routing method without compromising performance in expander networks.**

A. Drawbacks of existing methods handling deadlocks

1) Deadlock recovery: Deadlock recovery is a method that involves detecting and then resolving deadlocks. Conventional approaches for deadlock recovery operate on the control plane [15], [22], which can be problematic due to the large latency between the data plane and the control plane. As a result, deadlocks can not be resolved quickly, leading to network performance degradation.

Even if a recent approach, ITSY [27], detects and resolves the deadlocks on data plane with lower latency, its three methods to resolve deadlocks fall short. The proposed methods either introduce packet loss or struggle to handle concurrent deadlocks. Additionally, ITSY relies on programmable switch hardware, such as P4, which is still not widely deployed in data center networks.

2) Lossy RDMA: One of lossy RDMA approches is lossy RoCE. The lossy RoCE is to disable the PFC and redesign the RoCE to work under lossy network [10], [16]. While lossy RoCE effectively eliminates deadlocks, it negatively impacts the latency performance of mice flows. Additionally, the implementation of lossy RoCE often requires specific hardware support, such as Mellanox ConnectX-4 NICs [19]. IWarp [21] is another lossy RDMA approach. But it relies on TCP for lossless delivery, which leads to poor performance.

3) Lossless Priority Management: This approach requires to switch the lossless priorities of packets hop-by-hop [6], [13]. It has been widely adopted in HPC environments that utilize Infiniband networks supporting up to 15 lossless priorities. However, the RoCE networks can only support at most 2 to 3 lossless priorities due to the constrained buffer space [9].

Besides, Tagger [12] proposes generic strategies to reduce the number of lossless priorities required for deadlock prevention. However, even expander networks with several hundred of switches typically require at least 3 lossless priorities. Additionally, to prevent packet loss, switches need to allocate sufficient buffer space (headroom) during the transmission of PFC PAUSE frames. However, with the trend towards using relatively shallow buffers in modern data center networks [8], [9], [28], the challenge of reserving headroom for at least 3 lossless priorities becomes even more pronounced.

B. Drawbacks of existing deadlock avoidance routing

This approach aims to design routing methods that eliminate Cyclic Buffer Dependencies (CBDs) by restricting the routing choices. FC (Flattened Clos) [30] is the state-of-the-art deadlock avoidance routing which outperforms EDST routing [24], [25] significantly (EDST is another deadlock-free routing). However, due to the restriction on routing choices, FC still has a noticeable performance gap compared to KSP [30].

C. Obervation: poor path diversity of FC [30] hinders its performance

We provide a toy example to introduce FC topology and virtual up-down routing first. FC splits each ToR switch into k virtual switches and assign them to k virtual layers accordingly. As shown in Fig. 1, there are 3 virtual layers. FC enforces the up-down constraint for routing to eliminate deadlocks. It allows paths in the up direction from a lower layer to a higher layer, such as $S_1^2 \rightarrow S_2^3$ in Fig. 1. Similarly, paths in the down direction from a higher layer to a lower layer are permitted, like $S_2^3 \rightarrow S_2^2 \rightarrow S_3^1$. Additionally, paths that first go up and then go down, such as $S_1^2 \rightarrow S_2^3 \rightarrow S_2^2 \rightarrow S_3^1$, are also allowed in FC. Among all up-down paths between each switch pair, FC's routing only uses edge disjoint up-down paths. (The edge disjoint constraint ensures that each path cannot share any common links.)

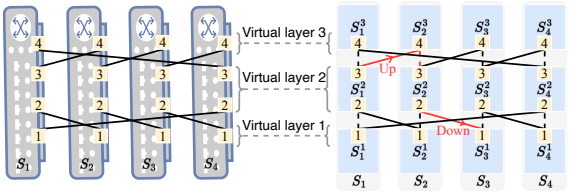


Fig. 1: A toy example of FC.

In Tab. I, we consider switches with 32 ports and connect them to other switches using 18 ports. Under FC's topology, we calculate both the average number of up-down paths and the average number of edge-disjoint paths between all switch pairs. It is observed that FC's topology has small number of

up-down paths. Furthermore, the edge disjoint routing in FC utilizes only a small fraction of these available paths.

The Num of Switches	Average Num of Up-down Paths	Average Num of Edge Disjoint Up-down Paths
100	31.67	6.48
200	16.07	5.01
300	10.78	4.22
400	22.83	4.92
500	18.35	4.55

TABLE I: Average num of paths (FC's topology and routing).

In conclusion, the lack of path diversity in both FC's topology and routing significantly impairs its performance. **Therefore, by enhancing path diversity, FC+ aims to achieve near-optimal performance.**

III. FC+ (Flattened Clos Plus)

Building upon the design principles of FC, we introduce a new expander topology-routing co-design called Flattened Clos Plus (FC+).

A. Design of topology (non-uniform virtual topology)

Assuming the data center network consists of N ToR switches denoted as $S = S_1, S_2, ..., S_N$, where each switch has p ports (s of which are connected to other switches and h of which are connected to hosts). The construction of FC+'s topology involves three key steps, which are outlined below:

STEP 1: Generating virtual switches. To construct the virtual topology with k layers, we split each ToR switch into v virtual switches ($2 < v \leq k$). These virtual switches, labeled as $S_i^1, S_i^2, ..., S_i^v$, represent the j-th virtual switch of switch S_i. Each virtual switch S_i^j, is connected to l_j virtual switches from other ToR switches, ensuring the desired connectivity. l_j must satisfy the following equation:

$$l_j = \begin{cases} 1, & j = 1, v \\ \dfrac{s-2}{v-2}, & 1 < j < v \end{cases} \tag{1}$$

STEP 2: Assign virtual switches to virtual layers. We represent the virtual layers as $L = \{L_1, L_2, ..., L_k\}$. The value of k is determined by the equation:

$$k = \frac{s-2}{2} + 2 \tag{2}$$

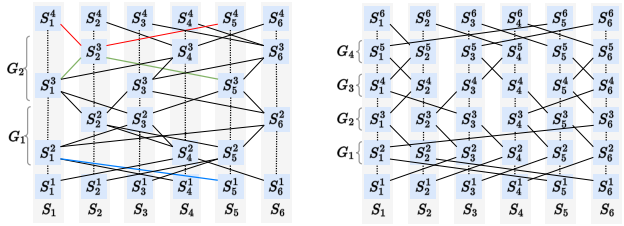
Initially, we assign the virtual switches S_i^1 and S_i^v to L_1 and L_k respectively. The remaining $k-2 = \frac{s-2}{2}$ virtual layers are divided into $v-2$ groups labeled as $G_1, G_2, ..., G_{v-2}$. Each group G_l ($1 \leq l \leq v-2$) contains the same number of virtual layers, denoted as N_{gl}, given by:

$$N_{gl} = \frac{s-2}{2(v-2)} \tag{3}$$

Each virtual switch S_i^j ($1 < j < v$) is assigned to the $(j-1)$-th group G_{j-1}. All the virtual layers within each group should have an equal number of virtual switches.

STEP 3: Random Wiring. We randomly generate $k-1$ bipartite graphs between the adjacent virtual layers for FC+ topology. Each link in bipartite graphs should be created between 2 virtual switches from different ToR switches. Since S_i^j and S_i^{j+1} belong to the same switch, and there is no need to create a link in between.

To illustrate the generation of FC+ topology, we provide two simple examples using 6 switches with $s = 10$, as shown in Fig. 2. In both examples, we choose v to be 4 and 6 respectively. Furthermore, k should be 6.



(a) FC+ topology 1. (b) FC+ topology 2.

Fig. 2: Simple FC+ topology.

In Fig. 2(a), when $v = 4$, there is 2 virtual layer groups (G_1, G_2) with 2 layers in each group ($N_{gl} = 2$). We assign S_i^1 to L_1 and S_i^4 to L_6. The S_i^2 and S_i^3 can be randomly assigned to G_1 and G_2 (ensuring that L_2 to L_5 have the same number of virtual switches). Once the assignments are made, we can generate the bipartite graphs for wiring randomly.

In Fig. 2(b), when $v = 6$, we have 4 groups of virtual layers, each with 1 layer. Similarly, We assign S_i^1 to L_1 and S_i^6 to L_6. The $S_i^j (1 < j < v)$ is separately assigned to G_{j-1}. Notably, it is also a FC topology with the maximum number of layers.

Different from the constrain of FC ($v = k$), FC+ allows $v \leq k$. Therefore, we call FC+'s virtual topology as **non-uniform virtual multi-layer topology**.

B. Deadlock-free KSP routing (DF-KSP)

In order to enhance the path diversity of FC+'s routing, we propose the **Deadlock-free K-shortest Paths Routing**. We follow 2 steps to compute the paths:

STEP1: Construct the undirected virtual multi-layer topology. Following the guidelines outlined in § III-A, we can construct the FC+ topologies. These FC+ topologies can then be virtualized into multi-layer topologies, as illustrated in the example shown in Fig. 2.

STEP2: Compute DF-KSP paths. We first introduce a metric called **down-up times** (T_{du}) to quantify the number of times a path traverses from a higher virtual layer to a lower virtual layer and then returns to a higher virtual layer.

For each pair of source and destination ToR switches (S_i, S_j), our objective is to obtain K paths for routing. So we first calculate the shortest path between S_i and S_j in the undirected virtual multi-layer topology. Subsequently, we determine T_{du} for the computed path. If $T_{du} = 0$, we store the path and proceed to compute the next shortest path. Otherwise, we discard the path and move on to the next shortest path. This process is repeated until we have stored K paths.

Toy Example. As shown in Fig. 2(a), we aim to calculate $K = 2$ deadlock-free shortest paths between S_1 and S_5. Starting with the shortest path $S_1^2 \rightarrow S_5^1$ (Blue Line), we observe that T_{du} for this path is 0. Hence, we store this path as one of the desired up-down shortest paths. Next, we examine the second shortest path $S_1^4 \rightarrow S_2^3 \rightarrow S_5^4$ (Red Line). This path descends from L_6 to L_5 and then ascends from L_5 back to L_6. Consequently, its T_{du} value is 1, and we drop this path. Moving forward, we compute the next path $S_1^3 \rightarrow S_2^3 \rightarrow S_5^3$ (Green Line), and upon evaluation, we find that T_{du} for this path is 0. As a result, we obtain the desired 2 deadlock-free shortest paths between S_1 and S_5.

Due to the utilization of only 1 lossless priority, we refer to the routing method as **FC+ (1L-K)**, where K represents the number of paths used for routing.

C. Improvement on path diversity by 2 new designs

FC+'s non-uniform virtual multi-layer topology. We maximize the number of virtual layers (k) and reduce the number of virtual switches (v). In FC topology, k is set to be the smallest or second smallest value to guarantee each switch pair has at least 1 up-down path. In FC+, we directly set k to be the largest value of FC virtual topology. Besides, in FC, v must be equal to k, while FC+ allows $v \leq k$. These designs improve the path diversity within FC+'s topology, offering more efficient routing options. The benefits of these designs are further analyzed in § IV.

DF-KSP rather than edge disjoint routing. We utilize the DF-KSP to enhance the path diversity of routing. As shown in Tab. I, edge disjoint routing utilizes only 4 to 6 paths due to the constraint that no two paths for a given pair of switches can share the same link, leading to poor path diversity. By DF-KSP, we can select a specific number of paths without deadlocks based on requirements, enabling the selection of a larger number of diverse and efficient paths for routing.

D. DF-KSP can use more than 1 lossless priorities

Another edge of DF-KSP over FC's routing is its ability to support multiple lossless priorities. In DF-KSP, we denote the number of such priorities as N_{lp}.

The steps for computing routing paths are similar to the method with 1 lossless priority. The difference lies in the criteria for storing paths. After computing the T_{du} of a path, we store it only if $T_{du} \leq N_{lp} - 1$. This is because each time a down-up condition occurs, we can switch the lossless priority of the packets from n to $n + 1$. Hence, we can perform up to $N_{lp} - 1$ priority transitions to avoid deadlocks. For instance, in the case of utilizing 2 lossless priorities, consider the path $S_1^4 \rightarrow S_2^3 \rightarrow S_5^4$ depicted in Fig. 2(a) (Red Line). At S_2, where the down-up condition occurs, we transition the packets' lossless priority from 1 to 2.

In FC+, using only 2 lossless priorities can achieve excellent performance. We name the routing method with 2 lossless priorities as **FC+ (2L-K)**. The lossless priority can be switched by matching the InPorts and OutPorts, like Tagger [12].

Remark*: Paths determined by FC+ (2L-K) with $T_{du} = 1$ can be split into two up-down paths. Each path is assigned a unique lossless priority, ensuring deadlock-free paths within the same priority. Thus, FC+ (2L-K) remains free from deadlocks.

E. Computational complexity of FC+ routing

In FC+ routing, the main computational task is computing K-shortest paths. The worst-case computational complexity of KSP is $\mathcal{O}(KV(E+V\log V))$, where E is the number of edges and V is the number of nodes. In an expander graph, where the number of edges is proportional to the number of nodes (E = CV, where C is a constant), the computational complexity of KSP can be simplified to $\mathcal{O}(KV^2 \log V)$.

FC+ (2L-K): When we find K-shortest paths on the undirected virtual topology, $V = vN$. However, if we want to find K paths, we need to pick K paths from K_2 shortest paths ($K_2 \geq K$). The computational complexity is $\mathcal{O}(K_2 v^2 N^2 \log N)$. Note that when each pair of switches in network has at most 1 link, we can directly compute the K-shortest paths on FC+ topology with N nodes because the paths in the FC+ topology are one-to-one mapped with the paths in the virtual topology. So the computational complexity can be reduced to $\mathcal{O}(K_2 N^2 \log N)$ under such condition.

To illustrate the relationship between K_2 and K, we provide Tab. II to show average value of K_2 respect to K and N, for network composed of 32-port switches (s = 18). As depicted in Tab. II, even as the network expands, the ratio between K_2 and K typically ranges from 1.5x to 3x.

N	K = 32	K = 64	K = 100
100	36.67	82.67	144.69
500	50.07	104.08	165.45
1000	51.53	119.41	231.86
1500	53.69	145.83	257.91

TABLE II: Average values of K_2 with respect to K and N.

FC+ (1L-K): In DF-KSP routing, when $N_{lp} = 1$, we need to drop many paths to find the K up-down paths. So $K_2 \gg K$. To reduce computational complexity, we adopt an alternative approach used by FC [30]. In this approach, we construct a directed virtual up-down topology based on the undirected multi-layer topology. The directed topology ensures all paths are up-down paths. Consequently, we can directly compute the K-shortest paths on this directed graph. The number of virtual nodes under directed graph is $V = (2v - 1)N$. Thus, the computational complexity becomes $\mathcal{O}(K(2v-1)^2 N^2 \log N)$.

IV. Numerical Analysis of FC+'s Topology

In FC+, the topology is determined by two parameters: k and v. We have provided Eq. (2) to determine the value of k. In this section, we focus on the numerical analysis of our choice of k and explain the benefits of reducing v. Additionally, we provide our strategy for selecting the appropriate value of v.

A. Numerical analysis of k.

We draw inspiration from FC topology to determine our choice of k for FC+. So we first establish a relationship between k and the number of up-down paths of FC. The

Num of Switches	k = 4			k = 6			k = 8			k = 10		
	N_P	N_{P6}	N_{P4}	N_P	N_{P6}	N_{P4}	N_P	N_{P6}	N_{P4}	N_P	N_{P6}	N_{P4}
400	8.13	8.13	4.13	37.67	23.76	6.08	133.90	43.59	7.72	411.75	63.41	8.95
600	5.43	5.43	2.76	25.33	15.93	4.06	90.51	29.22	5.16	279.42	42.51	5.98
800	4.08	4.08	2.07	19.07	11.97	3.05	68.15	21.96	3.87	211.89	31.96	4.48
1000	3.27	3.27	1.65	15.30	9.59	2.44	54.78	17.59	3.10	170.42	25.61	3.59

TABLE III: Average number of paths affected by k.

Num of Switches	v = 3 (N_{gl} = 8)			v = 4 (N_{gl} = 4)			v = 6 (N_{gl} = 2)			v = 10 (N_{gl} = 1)		
	P_{ave}^{max}	P_{ave}^{min}	N_{P4}	P_{ave}^{max}	P_{ave}^{min}	N_{P4}	P_{ave}^{max}	P_{ave}^{min}	N_{P4}	P_{ave}^{max}	P_{ave}^{min}	N_{P4}
400	6.00	3.00	37.56	5.03	3.34	20.82	5.38	3.63	12.14	5.47	4.15	8.95
600	6.15	3.00	30.83	5.43	3.56	13.87	5.81	3.69	8.11	6.03	4.41	5.98
800	6.49	3.06	24.24	5.75	3.59	10.42	5.91	4.01	6.08	6.28	4.72	4.48
1000	6.97	3.26	19.46	5.81	3.68	8.35	5.96	4.03	4.86	6.44	5.00	3.59

TABLE IV: Average number of paths and average length of paths affected by v ($k = 10$).

number of up-down paths of FC can be characterized by the expression [30]: $(1 + \frac{s}{2(k-1)})^{k-1}$. Therefore, when k satisfies Eq. (2) (maximum value of k), FC can obtain the maximum number of up-down paths.

We also use numerical results in Tab. III to verify the relationship between k and path diversity in FC. In Tab. III, we observe three metrics: N_P, representing the average number of up-down paths for all switch pairs; N_{P6}, denoting the average number of up-down paths within 6 hops for each pair; and N_{P4}, indicating the average number of up-down paths within 4 hops. These results are obtained using a network configuration consisting of 32-port switches with $s = 18$. We can see as k becomes larger, N_P, N_{P4} and N_{P6} all increases. This indicates that with a higher value of k, both the number of short paths and the number of long paths increase, resulting in improved path diversity of the topology.

In FC, we are concerned that a larger value of k may result in longer paths. Therefore, we tend to choose a small value for k to limit path length. However, this choice, along with the edge disjoint routing, leads to poor path diversity of FC. In contrast, in FC+, we aim to enhance path diversity, so we select the maximum value of k as indicated in Eq. (2).

B. Numerical analysis of v

In Tab. III, even when k reaches its maximum value, the average number of short paths (N_{P4}) remains small. To further enhance path diversity in FC+, we reduce the number of virtual switches (v) to increase the forwarding capacity of virtual switches. In Fig. 2(b), when k reaches maximum value and is equal to v, each virtual switch can only connect to at most 2 different ToR switches. By reducing v, virtual switches S_i^j ($1 < j < v$) can connect to more ToR switches. As depicted in Fig. 2(a), when v decreases from 6 to 4, the virtual switches can connect to at most 4 different ToR switches, which provides more options for packet transit.

We compute N_{P4} with different values of v, as shown in Tab. IV. It can be observed that as v decreases, the average number of short paths increases. This indicates that reducing the number of virtual switches enhances the path diversity.

However, using the smallest value of v ($v = 3$) may not be the optimal choice. As shown in Tab. V. We compute the near-worst case throughput of FC+ (1L-32) for a network with 400 switches ($N = 400$), where each switch connects to other 18 switches. The performance is notably subpar when $v = 3$.

v ($k = 10$)	10	6	4	3
N_{gl}	1	2	4	8
Throughput	0.278	0.287	0.297	0.142

TABLE V: Near-worst case throughput affected by v.

In Tab. IV, we separately compute the P_{ave}^{max} and P_{ave}^{min} to figure out the reason for the significant performance degradation (P_{ave}^{max} represents the maximum average length of the first 32 paths among all pairs of switches, and P_{ave}^{min} represents the minimum value). As shown in Tab. IV, as v decreases and N_{gl} increases, both P_{ave}^{max} and P_{ave}^{min} decrease at first. However, when v reaches 3 and N_{gl} reaches 8, P_{ave}^{max} suddenly increases to 6. We further analyze the path length of such pairs with $P_{ave}^{max} = 6$ and find that all the paths are at least 6 hops. This indicates that some pairs of switches only have long paths and most long paths of such switch pairs always share the same links, which exacerbates the performance degradation.

Based on Eq. (3), the decrease of v leads to the increase of N_{gl}. So the values of v and N_{gl} have a one-to-one correspondence. Therefore, we further analyze the relationship between N_{gl} and performance. We vary the value of N_{gl} and compute the throughput of near-worst case. Fig. 3 shows normalized throughput for 5 different s values. We observe that $N_{gl} = 5$ is a performance threshold across all cases.

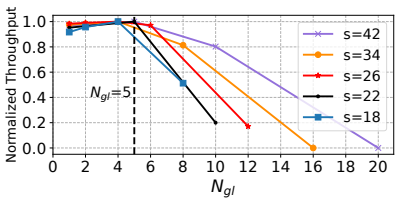


Fig. 3: Relationship between N_{gl} and normalized throughput

Based on the results depicted in Fig. 3, we introduce a constraint on N_{gl} to avoid the performance degradation.

$$N_{gl} <= 5 \tag{4}$$

Eq. (4) ensures that each group G_l contains a maximum of 5 virtual layers. By combining Eq. (3) and Eq. (4), we can derive the following constraint:

$$v \geq 2 + \frac{s-2}{10} \tag{5}$$

Strategy*: To determine the value of v, we select the smallest integer value of v that satisfies Eq. (5) and ensures that N_{gl} remains an integer. It is important to note that Eq. (4) and Eq. (5) are empirical formulas used in this selection process.

V. EVALUATION

A. Throughput analysis

In this section, we conduct numerical evaluation of the throughput and compare the performance of FC+ with KSP and FC. Besides, we compare FC+ (2L-K) with the theoretical upper bound. Furthermore, we compare FC+ with Clos networks that have similar number of switches as FC+. We adopt FC's method (Appendix A.3 in [30]) to build Clos topologies.

We use two sets of networks to assess the performance of FC+. In the 1st set of networks, we generate FC+ topologies with a maximum of 500 ToR switches ($p = 32$). Each ToR switch is connected to 18 other switches ($s = 18$) and 14 servers ($h = 14$). Following the strategies outlined in § IV-B, we choose $v = 4$ and $k = 10$. In the 2nd set of networks, we generate FC+ topologies with a maximum of 2000 ToR switches ($p = 32$). To ensure reasonable performance, each ToR switch is configured with $s = 22$ and $h = 10$. Using the same strategy, we set $v = 4$ and $k = 12$ for these networks.

We utilize multi-commodity flow formulation to calculate the throughput $\theta(T)$ for a given traffic matrix T. In our evaluation, we consider 3 types of traffic patterns: all-to-all, uniform random, and near-worst case. The all-to-all traffic pattern involves each server sending an equal amount of traffic to all other servers. This pattern is commonly used in MPI scenarios. Under the uniform random traffic pattern, each ToR switch randomly sends traffic to other switches. Specifically, we select 12.5% ToR switches for each ToR switch to send traffic to. This pattern is highly representative in data centers like Google's data center [20]. The near-worst case allows us to understand network's performance lower bound.

FC+ vs. FC: Considering that FC utilizes only 1 lossless priority, we specifically compare FC+ (1L-32) with FC to ensure fairness. In Fig. 4, FC+ exhibits superior performance compared to FC across all traffic patterns. Remarkably, even with 1 lossless priority, FC+ achieves a throughput performance improvement of 1.4x to 2x under the near-worst case scenarios. Furthermore, for the all-to-all and uniform random traffic patterns, FC+ demonstrates performance that is closer to KSP routing, while FC still exhibits noticeable performance gaps compared to KSP in all traffic patterns.

In Fig. 6(a), even with a larger number of ports ($s = 22$) connecting to other ToR switches and up to 2000 ToR switches in the networks, FC+ with 1 lossless priority achieves nearly 1.7x throughput compared to FC under the near-worst case.

The Num of Servers	728	1400	2128	2800	3528	4200	5600	7000
FC	7.85	6.46	5.55	5.02	4.56	4.29	4.91	4.55

TABLE VI: Ave. num of paths (1st sets of networks).

The Num of Servers	1000	3000	5000	10000	15000	20000
FC	10.44	7.69	6.50	5.93	8.58	8.11

TABLE VII: Ave. num of paths (2nd sets of networks).

Routing path analysis: We compare average number of paths among all pairs of switches. In both sets of networks, FC+ utilizes 32 paths when $s = 18$ and 40 paths when $s = 22$ for each pair of switches. However, Tab. VI and Tab. VII show that FC's average number of paths is always below 10 due to its virtual topology and constraint for edge disjoint paths. In contrast, FC+ with DF-KSP routing has the flexibility to utilize any number of paths, allowing for improved performance.

FC+ vs. KSP: KSP routing is commonly used in expander topologies such as Jellyfish [23] and Xpander [26] due to its high performance. However, KSP may result in deadlocks. We primarily compare the performance of FC+ with KSP.

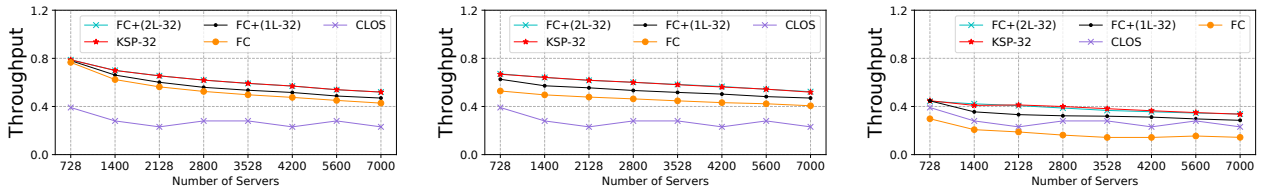
Fig. 4 and Fig. 6(a) demonstrate that FC+ can achieve nearly the same throughput as KSP routing. When utilizing two lossless priorities, the performance curve of FC+ (2L) overlaps with that of KSP. Even with only 1 lossless priority, the performance gap between FC+ (1L) and KSP is smaller than that of FC. Note that in Fig. 6(a), the throughput of KSP degrades when the number of servers reaches 10000. This is because KSP computes the next shortest path based on computed shortest paths. So there is a relatively low probability that most paths of some switch pairs may share the same links, which lead to performance degradation. Additionally, when employing 2 lossless priorities, the average path length of FC+ (2L) is almost the same as that of KSP, indicating that most paths of FC+ (2L) and KSP have the same lengths.

FC+ vs. Upper Bound: To demonstrate FC+ achieves near-optimal performance in typical traffic patterns, we calculate upper bound for each traffic metric using Eq. (6) from [17]:

$$\theta(T) \leq \frac{2E}{\sum_{u \in \mathcal{K}} \sum_{v \in \mathcal{K} \setminus \{u\}} t_{uv} P_{uv}} \tag{6}$$

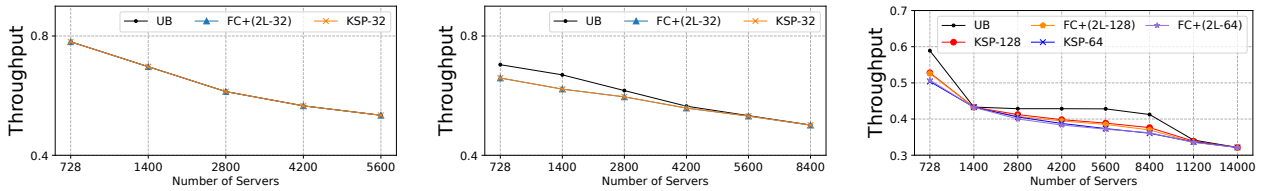
In Eq. (6), $\theta(T)$ represents the throughput for the traffic metric T. E denotes the number of links between switches. $\mathcal{K}$ represents the set of all ToR switches. t_{uv} denotes the traffic from switch u to v, and P_{uv} represents the length of the shortest path from u to v.

In Fig. 5, it can be observed that even in relatively small networks, with only 32 paths, both FC+ (2L) and KSP demonstrate a convergence towards the upper bounds for both all-to-all and uniform random traffic patterns. Furthermore, for the near-worst case, with 128 paths or even 64 paths, FC+ (2L)



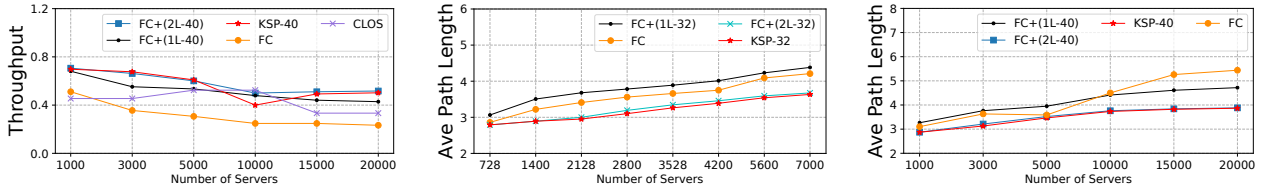
(a) Throughput of the all to all traffic matrix. (b) Throughput of uniform random traffic matrices. (c) Throughput of the near-worst permutation traffic matrix.

Fig. 4: Throughput of 1st set of networks: FC+ vs. FC, Clos and 32-way KSP routing (FC+ (2L-32) overlaps with KSP-32).



(a) Throughput of the all-to-all traffic matrix (UB overlaps with other two lines). (b) Throughput of uniform random traffic matrices. (c) Throughput of the near-worst permutation traffic matrix.

Fig. 5: Throughput: FC+ vs. Upper Bound (UB) and KSP.



(a) Throughput of 2nd sets of networks under the near-worst permutation traffic matrix. (b) Average path length of 1st sets of networks. (c) Average path length of 2nd sets of networks.

Fig. 6: Throughput and average path length.

and KSP also exhibit a convergence towards the upper bounds. These convergence trends align with the results reported in [17], where the throughput approaches the upper bound as the number of servers increases.

In conclusion, FC+ (2L) shows near-optimal performance in various traffic patterns, including near-worst cases and commonly encountered scenarios like uniform random patterns.

FC+ vs. Clos: Clos topologies are extensively used in data centers, and have witnessed the successful deployment of RDMA [9]. While FC demonstrates better performance than Clos in all-to-all and uniform random traffic patterns, it exhibits lower throughput than Clos under near-worst case due to its limited path diversity. Therefore, our focus is primarily on comparing FC+ and Clos in near-worst case. We employ FC's methodology to construct Clos topologies that maximize throughput, using a comparable number of switches [30].

In both Fig. 4(c) and Fig. 6(a), FC+ consistently outperforms Clos under near-worst case. Remarkably, even FC+ (1L) achieves superior performance compared to Clos.

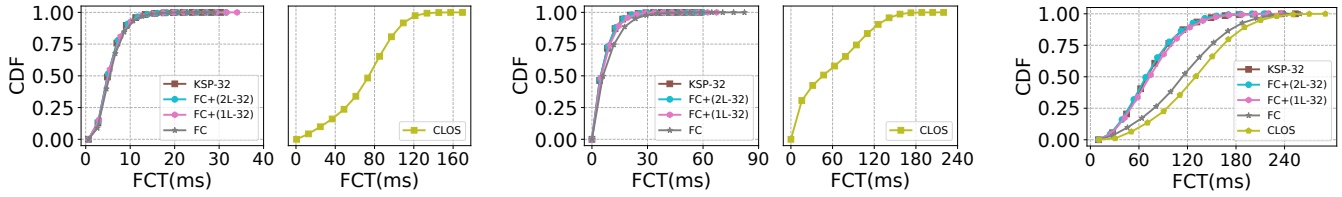
B. Packet-level simulation

We cross-validate throughput analysis using NS3 [11]. We first generate two topologies: FC and FC+ topology. Each topology consists of 120 switches with 32 ports, where 22 ports are connected to other switches and 10 ports are connected to servers. For the FC+ topology, we implement both DF-KSP routing and KSP routing, while for the FC topology, we use FC's edge disjoint routing. Additionally, we generate a 3-layer Clos topology with 124 32-port ToR switches, employing up-down routing. All the three topologies (FC, FC+, and Clos) are configured with a total of 1200 servers. We set the port rate to 25Gb/s and enable DCQCN for congestion control. The ECN marking related parameters are set as follows: Kmin=5KB, Kmax=200KB, and Pmax=0.01, as suggested by the DCQCN paper [31]. To thoroughly evaluate the performance, we adopt a relatively high workload of 70%.

The results of our packet-level simulation are presented in Fig. 7 and Tab. VIII. For the all-to-all traffic pattern, FC+, FC, and KSP-32 exhibit similar performance, while Clos performs poorly due to its lower throughput. Under the uniform random traffic pattern, Clos still exhibits the highest FCT. Additionally,

Network Setup	Num. of hosts	Num. of switches	All-to-All (load= 0.7)			Uniform Random (load= 0.7)			Near-Worst (load= 0.7)		
			Throughput	P50 FCT	P99 FCT	Throughput	P50 FCT	P99 FCT	Throughput	P50 FCT	P99 FCT
FC+(2L-32)	1200	120	1.204	4.935	14.794	1.101	4.736	24.462	0.634	68.713	163.339
FC+(1L-32)	1200	120	1.157	5.064	14.996	1.021	5.024	26.333	0.600	73.024	173.439
KSP-32	1200	120	1.204	4.923	14.655	1.101	4.701	24.233	0.624	70.507	176.324
FC	1200	120	1.102	5.302	15.274	0.882	5.937	34.764	0.413	113.753	225.508
Clos	1200	124	0.524	72.282	129.544	0.524	45.885	160.659	0.524	130.040	240.834

TABLE VIII: FCT results vs. throughput analysis.



(a) All-to-all traffic pattern (workload = 0.7).

(b) Uniform random traffic pattern (workload = 0.7).

(c) Near-worst traffic pattern (workload = 0.7).

Fig. 7: Compare FCTs for FC+, FC, KSP-32 and Clos.

FC shows performance degradation compared to FC+ and KSP-32, consistent with its lower throughput.

In the near-worst case, the limited path diversity of FC leads to a noticeable performance gap compared to FC+ and KSP-32. However, FC+ effectively utilizes available paths to achieve comparable performance to KSP.

C. Consumption of lossless priorities

The introduction of 1 more lossless priority in FC+ allows it to achieve near-optimal performance. Compared to Tagger [12], which requires 3 lossless priorities in the networks with a few hundred of switches, FC+ demonstrates its efficiency by using at most 2 lossless priorities.

As network bandwidth increases, the limited buffer space of switches becomes a challenge, particularly in accommodating 3 or more lossless priorities [8], [9], [28]. This is due to the need to reserve sufficient buffer space to prevent packet loss during the transmission of PFC PAUSE frames. The trend towards using shallow buffers in modern data center networks further exacerbates this challenge. In contrast, FC+ effectively balances performance and resource utilization.

VI. Conclusion

FC+ is a topology-routing co-design aimed at eliminating PFC-induced deadlocks while maintaining near-optimal performance in RoCE deployments over expander networks. By the new design of non-uniform virtual multi-layer topology, the introduction of DF-KSP routing, FC+ outperforms FC and achieve throughput improvement by 1.4x to 2x in near-worst case. Moreover, due to DF-KSP, FC+ overcomes the drawback of FC and has the ability to use more than 1 lossless priorities. Therefore, FC+ can achieve near-optimal performance with only 2 lossless priorities, reducing the number of lossless priorities from 3 to 2 compared to Tagger [12] (the state-of-the-art lossless priority management method to avoid deadlocks without performance loss).

References

[1] W. Bai et al. Empowering azure storage with {RDMA}. In *NSDI*, 2023.
[2] H. Ballani et al. Sirius: A flat datacenter network with nanosecond optical switching. In *SIGCOMM*, 2020.
[3] M. Besta and T. Hoefler. Slim fly: A cost effective low-diameter network topology. In *SC*, 2014.
[4] P. Cao, S. Zhao, M. Y. The, Y. Liu, and X. Wang. Trod: Evolving from electrical data center to optical data center. In *ICNP*, 2021.
[5] P. Cao, S. Zhao, D. Zhang, et al. Threshold-based routing-topology co-design for optical data center. *ToN*, 2023.
[6] W. J. Dally and C. L. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. 1988.
[7] Y. Gao et al. When cloud storage meets rdma. In *NSDI*, 2021.
[8] P. Goyal et al. Backpressure flow control. In *Proceedings of the 2019 Workshop on Buffer Sizing*, 2019.
[9] C. Guo et al. Rdma over commodity ethernet at scale. In *SIGCOMM*, 2016.
[10] M. Handley, C. Raiciu, et al. Re-architecting datacenter networks and stacks for low latency and high performance. In *SIGCOMM*, 2017.
[11] HPCC. https://shorturl.at/hopPV.
[12] S. Hu et al. Tagger: Practical pfc deadlock prevention in data center networks. In *CoNEXT*, 2017.
[13] D. Lee et al. Prevention of deadlocks and livelocks in lossless, backpressured packet networks, 2005. US Patent 6,859,435.
[14] H. Li et al. Performance of the 25 gbps/100 gbps fullmesh roce network using mellanox connetx-4 lx adapter and ruijie s6500 ethernet switch. In *WAINA*, 2019.
[15] P. Lopez, J.-M. Martínez, and J. Duato. A very efficient distributed deadlock detection mechanism for wormhole networks. In *HPCA*, 1998.
[16] R. Mittal et al. Revisiting network support for rdma. In *SIGCOMM*, 2018.
[17] P. Namyar et al. A throughput-centric view of the performance of datacenter topologies. In *SIGCOMM*, 2021.
[18] S. Novakovic et al. Storm: a fast transactional dataplane for remote data structures. In *SYSTOR*, 2019.
[19] NVIDIA. https://shorturl.at/fhFR3.
[20] L. Poutievski et al. Jupiter evolving: Transforming google's datacenter network via optical circuit switches and software-defined networking. In *SIGCOMM*, 2022.
[21] R. Recio, B. Metzler, P. Culley, J. Hilland, and D. Garcia. A remote direct memory access protocol specification. Technical report, 2007.
[22] A. Shpiner et al. Unlocking credit loop deadlocks. In *HotNets*, 2016.
[23] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey. Jellyfish: Networking data centers randomly. In *NSDI*, 2012.
[24] B. Stephens and A. L. Cox. Deadlock-free local fast failover for arbitrary data center networks. In *INFOCOM*, 2016.
[25] B. Stephens et al. Practical dcb for improved data center networks. In *INFOCOM*, 2014.

[26] A. Valadarsky et al. Xpander: Towards optimal-performance datacenters. In *CoNEXT*, 2016.
[27] X. C. Wu and T. E. Ng. Detecting and resolving pfc deadlocks with itsy entirely in the data plane. In *INFOCOM*. IEEE, 2022.
[28] Z. Yu et al. Programmable packet scheduling with a single queue. In *SIGCOMM*, 2021.
[29] S. Zhao, P. Cao, and X. Wang. Understanding the performance guarantee of physical topology design for optical circuit switched data centers. *SIGMETRICS*, 2022.
[30] S. Zhao et al. Flattened clos: Designing high-performance deadlock-free expander data center networks using graph contraction. In *NSDI*, 2023.
[31] Y. Zhu et al. Congestion control for large-scale rdma deployments. *SIGCOMM*, 2015.