Performance Evaluation of Routing Schemes in Data Center Clos Networks

Yi Wang, Chen Tian(✉), Shengjun Wang, and Wenyu Liu

School of Electronic Information and Communications,
Huazhong University of Science and Technology,
Wuhan, Hubei, People's Republic of China
{ywang,tianchen,sjwang,liuwy}@hust.edu.cn
http://ei.hust.edu.cn

Abstract. A number of new Clos data center networking fabrics have been proposed recently by using commodity off-the-shelf hardware. In Clos networks, randomized routing is typically used to achieve load balance. However, with typical communication patterns, pure randomized routing design cannot fully exploit bandwidth provided by Clos topologies. This paper strives to evaluate the performance of both randomized routing and its alternatives. As an incremental improvement, periodic path renewal is discussed first. We then discuss adaptive routing, together with its implementation consideration and costs. To reduce overhead of pure adaptive routing, a mixed routing scheme is also discussed. Through detailed discussions and comprehensive evaluations, we provide designers with useful insight and a range of options for configuring the routing scheme.

Keywords: Clos network · Routing · Data center networks

1 Introduction

Cloud services are driving the creation of gigantic data centers which can concurrently satisfy computing and storage needs for many global businesses. Running distributed file systems (e.g. GFS) and distributed programming models (e.g. MapReduce and Dryad), these gigantic data centers may hold millions of servers in the near future. Unfortunately, existing multi-rooted-tree-like network fabrics cannot economically provide enough bisectional throughput among the servers of large data centers. The nearer to the root of a tree topology, the higher the cost of the switches/routers hardware. As a consequence, the capacity between

Yi Wang—This work is partially supported by "National Natural Science Foundation of China (No. 61202107, No. 61100220, No. 61202303)", by "National High Technology Research and Development Program of China (863 Program No. 2014AA01A702)", by "Natural Science Foundation of Hubei Province (No. 2014CFB1007), and by "National Key Technology Research and Development Program of China (No. 2012BAH46F03)".

S. Zhang et al. (Eds.): ICoC 2014, CCIS 502, pp. 80–94, 2015.
DOI: 10.1007/978-3-662-46826-5_7

different branches of the tree topology is typically oversubscribed by factors from 1:5 to 1:240 [8].

To solve this problem, a number of new data center networking fabrics have been proposed recently by using commodity off-the-shelf hardware. Both VL2 [8] and Fat-Tree [1] organize the switches into Clos networks to replace the current practice of multi-rooted-tree-like fabrics. A Clos network is a multi-stage non-blocking network with an odd number of stages [5], and a folded-Clos network is sometimes also called a fat-tree [11]. DCell and BCube [9] propose server-centric topologies to scale a data center with a few added ports on commodity servers. Because of the non-blocking switching capability, Clos networks are promising in data centers and are the focus of this paper.

Given the abundant paths in Clos networks, a desired data center routing scheme should well exploit the paths to improve the system throughput. Randomized routing, which hashes flows (a flow is a stream of packets that have the same values for a subset of fields of the packet header, such as the five-tuple) randomly across network paths, is adopted to achieve load balance in VL2 [8] and Fat-Tree [1]. However, with typical communication patterns, randomized routing design cannot exploit full bandwidth provided by Clos topologies. First, local uncoordinated randomized decisions do not take into account potential flow collisions and traffic unevenness in the network, thus cause unnecessary congestion and reduce the system throughput. Second, the presence of switch failures, which are common for commodity off-the-shelf hardware, may also cause traffic unevenness when fault-agnostic randomized routing is adopted.

To address this problem, in this paper we study alternative routing schemes in data center Clos networks. Clos networks have been used for a long time in the context of High Performance Computing systems (HPCs); routing schemes in Clos networks also have been extensively studied in system area [7,10]. This paper strives to exploit the evaluate the performance and cost of both randomized routing and its alternatives in the context of new data center Clos fabrics. To our best knowledge, routing scheme comparison in data center Clos networks has never been fully exploited before. Hedera [2] is the most related work; it discuss dynamic flow scheduling in Fat-Tree. While our paper discuss the general principals of routings in all Clos networks, especially focus on VL2. Our analysis (Sect. 2) and evaluation (Sect. 6) shows that, in terms of routing, VL2 is much more promising compared with Fat-Tree.

The main contributions of this paper include:

- As an incremental improvement, periodic path renewal is discussed first, where flows are periodically re-hashed to other paths to prevent persistent congestion.
- We bring forward adaptive routing and discuss the question of how to design adaptive routing to fit new Clos fabrics, together with its implementation consideration and costs.
- To combine the merits of both routing schemes and reduce overhead of pure adaptive routing, we further discuss a mixed routing scheme based on flow differentiation.

– We conduct extensive evaluations to these routing schemes. The simulation results show that adaptive routing can better utilize available link bandwidth compared with randomized routing; under the presence of network faults, adaptive routing can "smooth" the network traffic and provides significant higher throughput. While periodic path renewal can provide throughput improvement for randomized routing, its benefit over adaptive routing is almost negligible. The usage of flow-differentiation can significantly reduce the unnecessary overhead of pure adaptive routing without degrading the system throughput.

The rest of this paper is organized as follows. We provide the background of data center traffics and discuss new data center Clos networks in Sect. 2, together with their existing randomized routing designs. We discuss the integration of periodic path renewal technique and comparison between two representative Clos fabrics in Sect. 3. In Sect. 4, we discuss the design, implementation and costs of adaptive routing. Via flow-differentiation, mixed routing reduces the unnecessary overheads of pure adaptive routing, in Sect. 5. We evaluate these routing schemes by extensive simulations in Sect. 6. Related works are provided in Sect. 7. The paper is concluded in Sect. 8.

2 Background

2.1 Traffic Characteristics in Data Center

A highly utilized data mining cluster is instrumented by Greenberg et al. to understand the nature of the traffic matrix of the state-of-art data centers [8]. They got some important data center network measurement results:

Simpler Flow Size: Almost 99 % of flows are myriad small flows (mice), which is smaller than a threshold (100 MB in their cluster). On the other hand, more than 90 % of bytes are carried in flows larger than the threshold (elephants). Compared with Internet flows, the distribution is simpler and more uniform.

Volatile and Unstable Traffic Patterns: The number of representative traffic matrices in data centers is quite large. Also traffic is hard to be predicted, as only 1 % of the time does the network keep the same matrix for over 800 s [8].

Failure is Common: Most failures are small in size while their downtime can be significant. Even conventional 1+1 redundancy are still insufficient, as it is found that in 0.3 % of failures, all redundant components in a network device group became unavailable [8].

2.2 Data Center Clos Networks

Shown in Fig. 1, there are three levels of switches in VL2: ToR (Top of Rack) level, Aggregation level and Intermediate level. As a folded Clos topology [6], VL2 has extensive path diversity. The default routing design in VL2 is randomized:

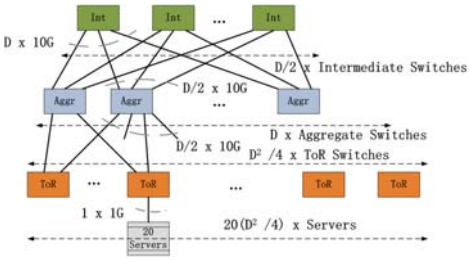

Fig. 1. VL2 Topology [8]

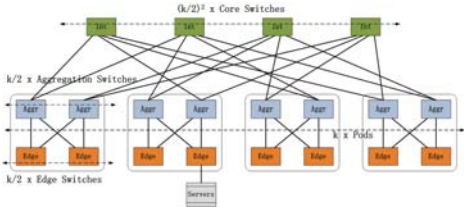

Fig. 2. Fat-tree Topology [1]

a flow from source server takes a random path across source ToR, up to one of two Aggregation switches, via a random chosen Intermediate switch, down to one of two Aggregation switches, across destination ToR, finally reach the destination server. Flow is the basic unit of traffic spreading and thus avoids out-of-order delivery [8].

For this randomized routing, VL2 implements two mechanisms: Valiant Load Balancing (VLB) distributes traffic randomly across intermediate nodes; Equal Cost Multipath (ECMP) makes Intermediate switch failures transparent to servers. VL2 defines anycast addresses for ECMP, each associates with a group of Intermediate switches. The source servers first hash across the anycast addresses; ECMP will then deliver the flows to one of the anycast address's active Intermediate switch.

Shown in Fig. 2, Fat-Tree [1] is also an instance of Clos networks. Edge and Aggregation switches are organized into Pods; Pods are interconnected via Core switches. Fat-tree also presents a randomized routing technique: Edge and Aggregation switches act as traffic diffusers. When faults are observed, a Core switch notifies its neighbors by broadcasting. The neighbors mark the affected link as unavailable and locally choose another link; the Core switch must back propagate the fault information all the way to the endpoints (source) so that subsequent messages exclude routes with the faulty path.

2.3 VL2 v.s. Fat-Tree

Over all, VL2 fabric is more promising from the routing congestion perspective. All links of Fat-Tree are homogeneous, say, 1 GBps. The congestion opportunities

could be homogeneous in every layer. For VL2, the links of aggregation layer have 10 times bandwidth than those of access layer. This aggregation effect can significantly alleviate the congestion opportunity. Our evaluations in Sect. 6 confirm the statement.

3 Randomized Routing in Clos Networks

3.1 Drawbacks of Randomized Routing

Randomized routing can achieve load balance and is simple for implementation, where each source locally selects a random middle-stage switch for a flow. However, it can't maximize the system throughput. Local decisions of randomized routing do not take into account potential flow collisions: randomization has the chance that large flows will be hashed to the same links or Intermediate/Core switches respectively, causing unnecessary congestion [8].

Yet another problem is: the presence of faults may cause unevenness in the network traffic when fault-agnostic randomization routing is adopted. For example, if a VL2 Intermediate switch fails in an anycast address group, all flows to this address are undertaken by remaining active switches in the same group. As a consequence, more flows are allocated to active switches in the same group than that of other switches in different groups, and these switches are more likely to be congested. The same situation also happens to a k–ary Fat-tree when a link from an Aggregation switch to a Core switch fails: all traffic are routed to remaining $k/2-1$ links of the Aggregation switch. Enlarge the group (e.g. upgrade ECMP to support more ways in VL2), if applicable, would only alleviate this problem, but can't get rid of it.

3.2 Periodic Path Renewal

To prevent persistent congestion, periodic path renewal is a promising technique: each running flow in the network periodically re-hashes its path after a fixed time interval T_r has passed since the last path decision. If large flows are hashed to the same path, they will compete for the path only for a short time and finally some of them will re-hash to other paths. In the presence of traffic unevenness, flows hashed to heavy switches/links are more likely to experience congestion; when their renewal timers time out, those flows have a chance to be hashed out to light switches/links; flows previous in the light ones also have an equal chance to be hashed into heavy ones.

Generally elephant flows could survive at least several renewal periods, especially under heavy network traffic. From a statistical point of view, flows would be equally treated if they persist long enough. We expect that, if periodic path renewal is introduced in, randomized routing can be more fair for flow allocation hence the throughput can be improved.

However, there are still two problems.

- Without a global view of network status, blindly re-hashing flows is not optimal: a flow previously in a contended path may be randomized to another contended or even more contended path; even worse, an elephant in an uncontended path may be unnecessarily randomized to a contended path. These situations would still cause unnecessary congestion and lower the whole throughput.
- In the presence of network faults, local uncoordinated flow allocation is still unfair: some switches/links are been allocated more flows than others. This also results in a degradation of the throughput.

These problems are inherent to randomized routing and are unsolvable by periodic path renewal. For comparison purpose, we also evaluate the effects of periodic path renewal for adaptive routing in Sect. 6.

4 Exploring Adaptive Routing

The idea of adaptive routing is to dynamically find a best path for each flow based on the present network traffic status. Efficient adaptive routing can minimize flow collision that occurs when many flows compete for bandwidth. Adaptive routing is also inherent fault-tolerant to already happened faults: all failed nodes/links are automatically detected/avoided by the routing mechanisms; hence adaptive routing can "smooth out" any unevenness in the network to load balance the traffic to the set of available forwarding switches/links.

For each flow, an adaptive path selection process should be performed first: additional start up delay may be added. Fortunately, such delay can be avoided [9]: when a source is performing path selection for a flow, it does not hold packets. The source uses a default path selecting (e.g. a randomized one) for that flow first. After the routing process completes and a better path is selected, the source switches the flow to the new path.

However, these merits do not come at no cost. The design and implementation of adaptive routing mechanism, either centralized path scheduling or distributed path probing, should be provided. The next two parts present our considerations.

4.1 Centralized Path Scheduling

Centralized path scheduling requires a global view of the network status, including links/ports utilization and nodes health. We expect that a dedicated Fabric Master system periodically receives link utilization reports from switches that take part in forwarding. Routing Agents, either on the application servers or edge switches, would implement the adaptive routing mechanism for flows. We also expect a centralized Routing Scheduler: a Routing Agent sends routing requests for flows to Routing Scheduler; Routing Scheduler replies with the best path back to the sender in a timely manner.

It is assumed that Routing Scheduler use a simple scheduling algorithm to sequentially serve incoming routing requests: (1) finds all available paths between two hosts; (2) selects a best suitable path for the flow.

An illustrative implementation framework is shown in Fig. 3. To avoid single-point-of-failure, both Fabric Master and Routing Scheduler should have replications.

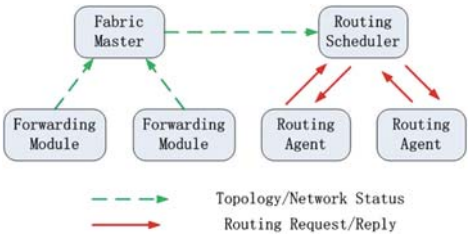

Fig. 3. The Framework of Centralized Path Scheduling

The Fabric Master has a logical matrix with per-link utilization/connectivity information for the entire topology and updates it with the new information. Reports could also serve as heartbeats, which can be used to deduce node/link failure status. Switches can judge its links failure; upon not receiving a report for some configurable period of time, the Fabric Master assumes a switch failure. Fabric Master can be integrated into the existing network control system and should export query interfaces to Routing Scheduler.

Routing Scheduler can be integrated into the existing directory systems in data center Clos network designs [8]: Routing Agent could send lookup requests for each flow which need to perform adaptive routing; after scheduling, the directory system returns the selected routing path information to routing agents as part of the address lookup results. Routing Agents can be integrated into existing Clos networks agents on the application servers (e.g. VL2) or edge switches (e.g. Fat-Tree).

The main cost comes from the mechanism of centralized network status collection. In VL2 topology, Aggregation switches connect both Intermediate switches and ToR switches, hence link utilization reports can be collected solely from Aggregation switches. For a D-port VL2 fabric, there are D Aggregation switches, each with D-port. Each switch port has two directions: *in* and *out*. If we denote utilization of each direction with 1 byte, then an Aggregation switch reports contain $2 * D$ bytes data. That means even a $D = 144$ port switch can report its status in a single packet. If each switch sends reports at a T_r interval, then the number of all report packets is only $1/T_r$ per seconds per switch. Even if each packet is 500 bytes alone, the report only consumes $500{*}8{*}1/T_r = 4/T_r$ kbps of each 10G switch. For the whole $D = 144$ fabric (over 100,000 servers), it consumes $4/T_r * 144 = 576/T_r$ kbps. Even if $T_r = 0.1$ s, the network cost is only several Mbps. Calculation of Fat-Tree also gets similar results.

4.2 Distributed Path Probing

The concept of distributed path probing is similar with other literatures [9]: when a new flow came, the source sends probe packets over multiple paths; the passing

switches process the probe packets to fill the minimum available bandwidth of their incoming and outgoing links; the destination returns probe responses to the source. When the source receives the probe responses, it selects the best path from them. For simplicity, here we use Broadest Fit (BF) algorithm to select one from available paths: flow is assigned to the broadest path that can provide the maximum bandwidth.

One advantage of data center Clos networks is that all available paths can be easily identified. We take VL2 as an example. Observed that each ToR switch has only two outlet connections, for a pair of source and destination servers there are only $2 * 2 = 4$ possible $\langle$*source Aggregation, destination Aggregation*$\rangle$ pairs. With $D/2$ Intermediate switches in a D-port VL2 fabric, there are totally $4 * D/2 = 2 * D$ paths available. Similarly, there are totally $k^2/2$ paths available in a $k-$ary Fat-Tree network. It is apparent that the computation complexity of Broadest Fit routing algorithm is a constant time, which is linear to the network size. That means we can probe all available paths in data center Clos networks, which is almost impossible in hypercube networks [9].

The main cost comes from probing request and response which are proportional to the number of flows. For each request, there are $2 * D$ probing messages for a VL2 network; although the traffic volume is negligible, these path selection mechanism consumes CPU cycle and memory in hosts and network devices. We will evaluate these overheads in Sect. 6. However if mixed routing scheme is adopted (which will be presented later in Sect. 5), this cost can be greatly reduced.

5 Mixed Routing

It is obvious that there is no need to do adaptive routing for *EVERY* flow. As mentioned above in Sect. 2, almost 99 % of flows are less than 100 MB. Compared with elephants, the afforded overhead of traffic engineering to mice may be unnecessary. For example, keep alive operation may has only one or two messages per connection; it is unreasonable to do adaptive routing for such flows.

A mixed routing scheme to control adaptive routing cost can be used by *Flow Differentiation*: mixed routing uses different routing schemes for flows based on its predicted size. To be more specific, we can only perform adaptive routing for elephants and use randomized routing for mice.

Similar load-sensitive routing has been previously proposed in IP routing context [12], while our paper is the first work which evaluates Flow-Differentiation-Based routing in data center networks.

For design consideration, we propose to put the flow differentiation intelligence in the physical servers. Unlike Internet, servers in a data center are controllable.

The next problem of flow differentiation is: how can we identify elephant flows? The answer is to match the flow characteristics with identified patterns definitions. The patterns can be either proactive defined by a system administrator, such as known TCP ports of GFS chunk copy operations. Or, we can reactively change the flow type: all new flows are mice first; after transmitting a certain size, a new flow becomes an elephant flow.

6 Performance Evaluation

6.1 Evaluation Methodology

Topology. We have implemented all algorithms in a flow-level simulator; the TCP AIMD behavior is assumed for flows as that in [2]. The main experiment topology is a full $D = 16$-port VL2 Clos network. The topology has 8 Intermediate switches and 16 Aggregation switches which has sixteen 10 Gigabit ports. There are 64 ToR switches equipped with two 10 Gigabit uplink ports and twenty 1 Gigabit ports to servers. All $N = 1{,}280$ servers have 1 Gigabit port each. There are two anycast groups in the topology: Intermediate switches from 0 to 3 are assigned to group 0; Intermediate switches from 4 to 7 are assigned to group 1. The Fat-Tree topology is set to $k = 16$ and has 1,024 servers; this topology is only used in Part B for comparison.

Traffic Pattern. The flow size distribution exactly follows the measurement results of a data center [8]. For totally N servers in the network, there are N elephants with size over 100 MB and $99 * N$ mice with smaller size.

We use Paired All-to-All data shuffling pattern [7], as its name suggests, N servers are randomly split to $N/2$ pairs such as that no server is in more than a single pair. Each server is exactly the source of one elephant flow and the destination of one elephant flow. All mice flows are evenly distributed among a period with their source and destination randomly chosen. Elephants patterns and mice together emulate the typical ON-OFF burst communication style in data centers.

Note that we target "medium high load" regime: previous measurement shows that averagely only one large flow per server, and the 75th percentile is 2 [3]; our setting has 2 large flows for every server, which is slightly higher in load than normal scenarios. Our preliminary simulation shows that: when the load is too small in scale, naive random routing is already good enough; also, when the load is too high, there are too many elephant flows in the system, and there is seldom the luxury to improve the throughput by any scheduling scheme.

Metrics. The evaluation metrics include:

- *Maximum Bandwidth Utilization:* This metric reflects the normalized concurrent bisectional bandwidth in the network in percentage. The closer the metric to 100 %, the better a routing scheme utilizes the high capacity provided by the Clos networks. The Maximum Bandwidth Utilization reports the highest utilization during the whole experiment.
- *Elephants (Completion) Delay:* This metric reflects the average normalized completion time of elephants. Assume a minimum elephant flow with size 100 MB, its transmission would be completed in $100\,MB * 8/1\,Gbps = 0.8$ s if no congestion occurs; if it is completed in 1 s in an experiment, we denote its normalized completion time as $1/0.8 = 112.5\,\%$. Average completion time is inversely proportional to the achieved throughput of elephants. Generally speaking, the lower the metric, the higher the average elephants throughput.

Table 1. Experiments Scenarios

Scenario	Fail mode	Capacity loss
0	No Fail	0
1	Intermediate 0 Fail	12.5 %
2	Intermediate 0 and 2 Fail	25 %

- *Ratio of Been Selected (Switch):* We use a counter for each Intermediate switch to record how many times it has been selected for flows as the bounce off point. After an experiment, counts of every switch are normalized to their summation. Comparison of these ratios illustrates routing protocols' performance of local uncoordinated flow allocation fairness among switches, especially at the presence of nodes failures.
- *Number of Routing Requests:* This metric reflects the adaptive routing request/response issued throughout the whole data shuffling process. The lower the number, the lower the overhead of adaptive routing schemes.

Failure Scenarios. Throughout this Section, we use three different scenarios, see Table 1. Scenario 0 is ideal where no Intermediate switch fails; scenario 1, to simulate common fail situation, has one Intermediate switch broken; scenario 2 simulates the disastrous situation that broken switches are close in topology (e.g. Intermediate switch 0 and 2 reside in same group in our case). We focus on the Intermediate switches cases because they have direct impact over bisectional throughput.

Protocol Setting. We use *Random* or simply R to denote random routing protocol, *Adaptive* or simply A to denote pure adaptive routing protocol and *Mixed* or simply M for Mixed routing; *Random(NP)* denotes that periodic path renewal is NOT included and so is the others. For simplicity, the renew period is set to 0.1 s for all scenarios.

Centralized Path Scheduling is assumed in the simulation for adaptive routing. In this paper, we limit ourself to evaluate the effect of flow-differentiation performance to mixed routing. We suppose there are a fixed percentage of elephants are misjudged as mice and vice visa; by varying the percentages, we evaluate their effect over routing performance.

6.2 Randomized Routing in VL2 and Fat-Tree

In this part, we compare the performance of randomized routing in VL2 and Fat-Tree. All elephants are injected into the network within 0.4 s: as the minimum elephant needs at least 0.8 s to be transmitted, we can guarantee that all N elephants could compete in the network for at least 0.4 s. All mice are evenly distributed among 10 s. These traffic setting is consistent with the typical ON-OFF type of data center network [8].

Figure 4 presents the comparison of maximum system throughput. Due to the excessive congestion, randomized routing in Fat-Tree can only exploit around

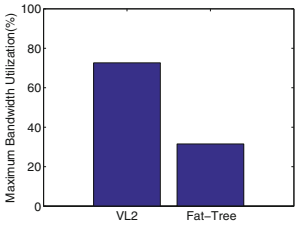

Fig. 4. VL2 v.s. Fat-Tree: Maximum Bandwidth Utilization (%)

30 % network bandwidth. This result is consistent with the Hedera project [2], which is a dedicated work for Fat-Tree traffic engineering. As a comparison, randomized routing can exploit over 70 % available bandwidth provided by Clos networks in VL2.

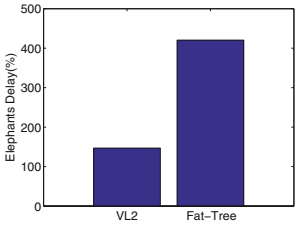

Fig. 5. VL2 v.s. Fat-Tree: Elephants Delay (%)

Figure 5 presents the comparison of average Elephants Completion Delay. Even with periodic path renewal, the *Elephants Delay* of Fat-Tree is around 420 % of standard delay while VL2 is only around 147 % seconds. Obviously the average throughput of VL2 is around 3 times higher than that of Fat-Tree. Due to the superiority of VL2, in later evaluations we will focus on VL2 topology.

6.3 Performance Comparison

In this part, we use scenario 0 (no switch failure) and Paired All-to-All traffic pattern. This experiment can exploit the ability of bandwidth utilization of routing schemes and exemplify the overhead of adaptive routing mechanism. There are 3 routing protocols and the periodic path renewal option, hence together we have 6 schemes to be evaluated.

Figure 6(a) presents the comparison of maximum system throughput. As we have expected, simple randomized routing can only exploit less than 70 % available bandwidth provided by Clos networks; by adding periodic path renewal, the *Random* throughput is only slightly improved to around 72 %. On the contrary, adaptive routing could achieve nearly 90 % maximum bandwidth utilization, i.e. 20 % higher than *Random*. Figure 6(b) also confirms the superiority of adaptive routing: the result of *Adaptive* is close to 110 %; *Random* is around 150 % which implies that average throughput is less than 2/3.

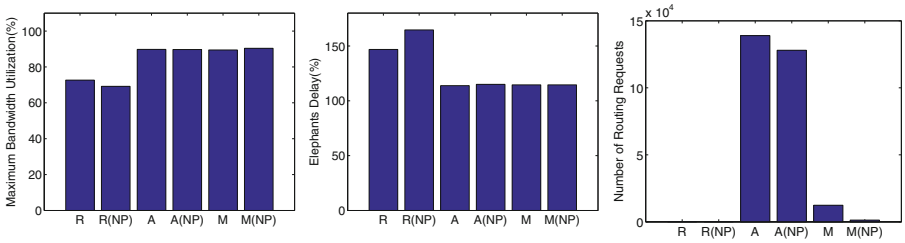

Fig. 6. Paired All-to-All pattern: (a) Maximum Bandwidth Utilization (%) (b) Elephants Delay (%) (c) Number of Routing Requests.

In both Fig. 6(a) and (b), the achieved performance of *Mixed* are almost the same as *Adaptive*. While the Number of Routing Request is significantly reduced by more than 90 %, as shown in Fig. 6(c). We can conclude that the use of flow-differentiation-based routing can significantly reduce the cost of pure adaptive routing while sacrificing minimum performance in ideal traffic pattern.

We can also conclude that, the performance improvement, by adding periodic path renewal, to both *Adaptive* and *Mixed* is almost negligible.

6.4 Resistance to Switch Failure

In this part, we analyze the performance of different schemes under switch failures scenarios.

Figure 7(a) shows simulation results of *Random* randomized routing. When no switch fails, the flows are almost evenly distributed among switches. When Intermediate switch 0 broken down, all flows randomized to anycast group 0 are now shared by other 3 switches. Apparently switches in anycast group 1 are not affected all. Further when Intermediate switch 2 also broken, the numbers of flows allocated to the remaining switches in group 0 (say, 1 and 3) double that allocated to the switches in group 1. It is clear that the presence of faults cause unevenness in the Clos network traffic when fault tolerant randomized routing is adopted.

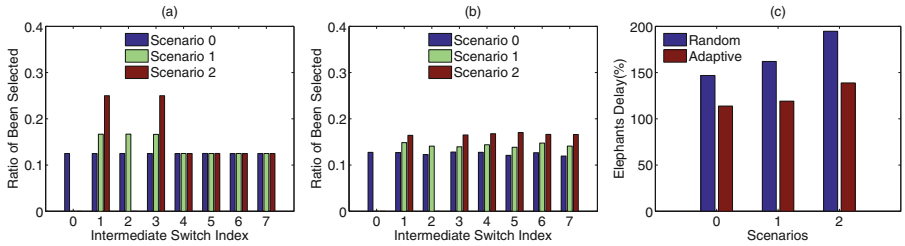

Fig. 7. (a) *Random* Ratio of switch been selected (b) *Adaptive* Ratio of switch been selected (c) Elephant delay comparison under failure.

Figure 7(b) also give the Ratio of Been Selected of *Adaptive* routing. Compared with *Random*, *Adaptive* could fairly distribute flow requests to Intermediate switches under *ALL* failure scenarios: adaptive routing framework can dynamically track the network node status and failed switches are automatically excluded from scheduling; flows are perfectly distributed among all active switches. We can conclude that adaptive routing is particularly useful in local uncoordinated fairness of flow allocation around traffic unevenness caused by the network faults.

Shown in Fig. 7(c), due to capacity loss, Elephants Delay increases for both schemes. Compared with Scenario 0, the performance difference is more apparent in Scenario 1; we even observed over 30 % average throughput improvement of *Adaptive* over *Random* in Scenario 2. It is clear that adaptive routing can "smooth" the traffic unevenness under the presence of network faults. As a result, adaptive routing is free of the risk of unnecessary congestion caused by nonuniformities.

6.5 The Impact of Flow-Differentiation Performance

In this part, we assume that there is a fixed percentage of elephants are misjudged as mice and vice visa; by varying the percentages, we evaluate their effect over routing performance. Two parameters are introduced in:

- *Elephants Error* A fixed percentage of elephants are misjudged as mice.
- *Mice Error* A fixed percentage of mice are misjudged as elephants.

By varying the percentages, we evaluate their effect over routing performance. In this part, *Elephants Error* is increased the from 0 % (perfect judgement) to 30 %, so is the mice.

Figure 8(a) shows that the achieved Maximum Bandwidth Utilizations are almost the same for different scenarios. This is correspondent to the results of Fig. 6(a) and (b): the use of flow-differentiation-based routing sacrifices minimum throughput performance. In Fig. 8(b), Elephants Delay does increase slightly with *Elephants Error*, which suggests that small *Elephants Error* value won't impair the performance of *Mixed* routing.

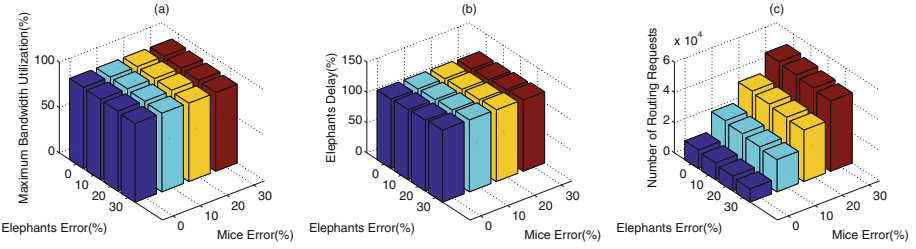

Fig. 8. Impact of Flow-differentiation Performance (a) Maximum Bandwidth Utilization (%) (b)Elephants Delay (%) (c) Number of Routing Requests.

The main difference is shown in Fig. 8(c): the routing overhead increases proportionally with *Mice Error*, which suggests that designers should be careful with their flow differentiation algorithms.

7 Related Work

Research of Clos network is more active in super-computing area: both adaptive routing and randomized oblivious routing have been implemented in a folded-Clos topology. Aydogan et al. showed that an adaptive routing provides better performance than randomized routing on the SP2 network [4]. Kim et al. has a detailed performance analysis and results for adaptive routing in a folded-Clos on chip interconnection, assuming a packet-switched networks [10]. Our research focus on the data center networks which are also packet switching based. The VL2 paper mentioned the possibility of "re-hashing the large flows periodically" to deal with the situations "that large flows will be hashed to the same links and Intermediate switches respectively, causing congestion", also no practical works are presented.

8 Conclusion and Future Work

Through detailed discussions and comprehensive evaluations, we provide data center network designers with useful insights: the usage of flow-differentiation can significantly reduce the unnecessary overhead of pure adaptive routing without degrading the system throughput. Our analysis can be useful to the development of various Cloud service and other networked systems [13–16].

References

1. Al-Fares, M., Loukissas, A., Vahdat, A.: A scalable, commodity data center network architecture. In: ACM SIGCOMM Computer Communication Review, vol. 38, pp. 63–74. ACM (2008)
2. Al-Fares, M., Radhakrishnan, S., Raghavan, B., Huang, N., Vahdat, A.: Hedera: dynamic flow scheduling for data center networks. In: NSDI, vol. 10, pp. 19–19 (2010)
3. Alizadeh, M., Greenberg, A., Maltz, D.A., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., Sridharan, M.: Data center TCP (DCTCP). ACM SIGCOMM Comput. Commun. Rev. **41**(4), 63–74 (2010)
4. Aydogan, Y., Stunkel, C.B., Aykanat, C., Abali, B.: Adaptive source routing in multistage interconnection networks. In: Proceedings of the 10th International Parallel Processing Symposium, 1996, IPPS'96, pp. 258–267. IEEE (1996)
5. Clos, C.: A study of non-blocking switching networks. Bell Syst. Tech. J. **32**(2), 406–424 (1953)
6. Dally, W.J., Towles, B.P.: Principles and Practices of Interconnection Networks. Elsevier, San Mateo (2004)

7. Geoffray, P., Hoefler, T.: Adaptive routing strategies for modern high performance networks. In: 16th IEEE Symposium on High Performance Interconnects, 2008, HOTI'08, pp. 165–172. IEEE (2008)
8. Greenberg, A., Hamilton, J.R., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D.A., Patel, P., Sengupta, S.: VL2: a scalable and flexible data center network. ACM SIGCOMM Comput. Commun. Rev. **39**, 51–62 (2009). ACM
9. Guo, C., Lu, G., Li, D., Wu, H., Zhang, X., Shi, Y., Tian, C., Zhang, Y., Lu, S.: Bcube: a high performance, server-centric network architecture for modular data centers. ACM SIGCOMM Comput. Commun. Rev. **39**(4), 63–74 (2009)
10. Kim, J., Dally, W.J., Dally, J., Abts, D.: Adaptive routing in high-radix clos network. In: SC 2006 Conference, Proceedings of the ACM/IEEE, pp. 7–7. IEEE (2006)
11. Leiserson, C.E.: Fat-trees: universal networks for hardware-efficient supercomputing. IEEE Trans. Comput. **100**(10), 892–901 (1985)
12. Soule, A., Salamatia, K., Taft, N., Emilion, R., Papagiannaki, K.: Flow classification by histograms: or how to go on safari in the internet. ACM SIGMETRICS Perform. Eval. Rev. **32**(1), 49–60 (2004)
13. Tian, C., Alimi, R., Yang, Y.R., Zhang, D.: Shadowstream: performance evaluation as a capability in production internet live streaming networks. In: Proceedings of the ACM SIGCOMM 2012 conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp. 347–358. ACM (2012)
14. Tian, C., Jiang, H., Iyengar, A., Liu, X., Wu, Z., Chen, J., Liu, W., Wang, C.: Improving application placement for cluster-based web applications. IEEE Trans. Netw. Service Manage. **8**(2), 104–115 (2011)
15. Tian, C., Jiang, H., Liu, X., Liu, W.: Revisiting dynamic query protocols in unstructured peer-to-peer networks. IEEE Trans. Parallel Distrib. Syst. **23**(1), 160–167 (2012)
16. Tian, C., Jiang, H., Wang, C., Wu, Z., Chen, J., Liu, W.: Neither shortest path nor dominating set: aggregation scheduling by greedy growing tree in multihop wireless sensor networks. IEEE Trans. Veh. Technol. **60**(7), 3462–3472 (2011)