Fast Deadlock-free Routing Reconfiguration for Arbitrary Datacenter Networks

July 10, 2016 7:36 AM

ABSTRACT

RDMA is being deployed in DCNs for the benefit of ultra-low latency, high throughput and low CPU overhead in recent years. Current practice of RDMA deployment introduces the deadlock problem as it requires PFC to provide a lossless L2 network. While deadlock can be avoided by using a routing function that includes no cyclic buffer dependency, in this paper we demonstrate that for both tree based and non-tree based DCNs, reconfiguration-induced deadlock could still occur during the routing reconfiguration process even if the routing functions are deadlock-free.

Deadlock-free routing reconfiguration can be ensured by simply diving the reconfiguration process into multiple static stages. However, it could lead to a very slow routing reconfiguration as many unnecessary constraints on the ordering of update actions are introduced. Motivated by this, in this paper, we develop an approach for achieving fast deadlock-free routing reconfiguration which introduces much less constraints on the ordering and can significantly speed up the routing reconfiguration process.

1. INTRODUCTION

The growing demand for online services and cloud computing has driven today's datacenter networks (DCNs) to a large scale with hundreds of thousands of servers and thousands of switches. With this enormous number of network devices, network failure and device upgrade become the norm rather than the exception.

Network reconfiguration will be needed when there is failure or upgrade of links/nodes, new switch onboarding, load balancer reconfiguration, etc. To support this, the network's routing function, which includes all the paths packets can take in the network, are often needed to be reconfigured for the purpose of either maintaining the connectivity of the network or better serving the current network traffic.

On the other hand, as DCNs enter the 40/100Gbps era, RDMA is currently being deployed for achieving ultra-low latency, high throughput and low CPU overhead. To enable efficient operation, RDMA usually runs over a lossless L2 network. The using of a lossless L2 network introduces the deadlock problem into the DCNs, which refers to a standstill situation where a set of switch buffers form a permanent cyclic waiting dependency and no packet can get drained at any of these buffers. Once deadlock occurs, no packet can be delivered through a part of or even the whole DCN.

Under static circumstances (i.e., when both of the network topology and the routing function are fixed), deadlock can be avoided by using a routing function that contains no cycle in the corresponding buffer dependency graph.

Under dynamic circumstances, however, deadlock may occur during reconfiguration process when transitioning from an old deadlock-free routing function R_s to a new deadlock-free routing function R_t. This is because during the routing reconfiguration process, due to the asynchronous updates of switch rules, any paths included in $R_s \cup R_t$ may take effect at the same time. When $R_s \cup R_t$ contains a cycle in the corresponding buffer dependency graph, deadlock may occur if the routing reconfiguration process is not well planed. We refer to this kind of deadlock as *reconfiguration-induced deadlock*.

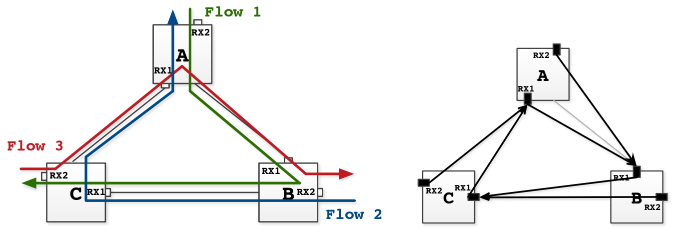
Reconfiguration-induced deadlock can be avoided by imposing some constraints on the ordering of configuration actions during the reconfiguration process. For example, deadlock-free can be guaranteed by removing all the paths included in R_s first before adding any new path included in R_t. Alternatively, we can remove some paths in R_s to reduce the routing function into $R_s \cap R_t$ at first, and then add the new paths included in R_t to finish the reconfiguration procss.

The speed of routing reconfiguration is important as it determines the response time to a network failure. Although both of the above approaches can ensure deadlock-free, they will lead to a slow routing reconfiguration process as multiple staitc intermediate stages are needed.

In this paper, we develop an approach for achieving fast deadlock-free routing reconfiguration. It is based on two observations: 1) there exist multiple valid orderings that is deadlock-free; and 2) choosing an ordering with minimum order dependencies among configuration actions can lead to fast reconfiguration. Our approach is general and can be applied to arbitrary DCNs, including Fat-tree, VL2, HyperX, Jellyfish, etc.

2. BACKGROUND AND MOTIVATION

2.1 PFC deadlock problem



(a) Topology and flows. (b) Buffer dependency graph.

Figure 1: Flows 1, 2 and 3 forms a cycle in the buffer dependency graph that can create a routing deadlock. In both figures, RX represents an ingress switch queue.

Priority-based Flow Control (PFC): The deployment of RDMA over Ethernet requires PFC to provide a lossless L2 network. PFC is a mechanism for ensuring zero packet loss under congestion in data center bridging (DCB) networks. PFC allows an overwhelmed network device to send a PAUSE frame to its immediate upstream device, which halts the transmission of the sender for a specified period of time.

PFC works in a per ingress queue fashion. When PFC is enabled, the switch will maintain a counter to track the virtual queue length of each ingress queue. Once the queue length exceeds a pre-configured PFC threshold, a PAUSE frame will be generated.

PFC deadlock problem: The using of PFC can cause deadlock problem. In Fig. 1, we use a simple example to show how deadlock can be created when there is a cyclic buffer dependency among a set of switch buffers.

As shown in Fig. 1(a), three flows are runing over three switches A, B and C. Flow 1 starts at a host (not shown) attached to A, passes through B, and ends at a host attached to C. Flow 2 and flow 3 are two symmetric flows of flow 1. Buffer dependencies among active ingress queues are drawn in Fig. 1(b). The path flow 1 takes introduces two dependency links, one from RX2 of A to RX1 of B, the other from RX1 of B to RX1 of C. Similarly, paths taken by flow 2 and flow 3 introduce the other four dependency links in Fig. 1(b).

As we can find in Fig. 1(b), the paths taken by the 3 flows introduce a cyclic buffer dependency among switches A, B and C. When network congestion occurs, it is possible that all the three RX1 queues become full of the packets destined for the next-hop switch and trigger PFC PAUSE simultaneously. Then a PFC deadlock is created as links A-B, B-C and C-A will be permanently paused and no packet in the three RX1 queues can ever get drained.

PFC deadlock problem can be avoided by leveraging a routing function that introduces no cycle in the buffer dependency graph. However, this approach cannot eliminate the cyclic buffer dependency during routing reconfiguration, as we are going to show in the next.

2.2 Reconfiguration-induced Deadlock

In this part, we use examples to show 1) cyclic buffer dependency can be generated for both tree based and non-tree based DCNs when the routing reconfiguration is not well planed; 2) a bad deadlock-free reconfiguration plan will lead to a slow reconfiguration process.

2.2.1 *Deadlock Under Tree Based DCNs*

Fig. 2(a) shows a small Leaf-Spine DCNs. The capacity of all the links are 40Gbps. Due to maintenance issue, the network operator now wants to replace two links L2-T2 and L3-T3 in the topology. To avoid long-term packet loss during link replacement, the network traffic passing through these two links needs to be migrated to some other paths.

In this example, we consider the traffic from switch T1 to switch T4 following the path T1-L2-S1-L3-T4, and the traffic from switch T3 to switch T2 following the path T3-L3-S2-L2-T2. There are three alternative paths that the traffic load from T1 to T4 can migrate to: 1) path-1 along T1-L1-S1-L4-T4; 2) path-2 along T1-L2-S1-L4-T4; 3) non-shortest path-3 along T1-L2-S1-L3-S2-L4-T4.

As we can find in Fig. 2(a), the traffic load from T1 to T4 is 15Gbps. Path-1 and path-2 thus are not congestion-free choices as the load on link S1-L4 are 30Gbps, larger than 25Gbps. Path-3 is a good choice the load on links L3-S2, S2-L4 and L4-T4 are all smaller than 25Gbps.

While not drawn in the Fig. 2(a), let's say non-shortest path T3-L3-S2-L2-S1-L1-T2 is the only congestion-free choice that the traffic from T3 to T2 can migrate to. Fig. 2(b) shows the routing state after performing congestion-free routing reconfiguration.

In Fig. 2(c), we focus on four switches L2, L3, S1 and S2. We reposit the locations of these four switches and draw both ingress queues (RX) and egress queues (TX) for the purpose of better explanation. As shwon In Fig. 2(d), there is a cyclic buffer dependency among the ingress queues. This indicates that the network is now exposed to the danger of PFC deadlock.

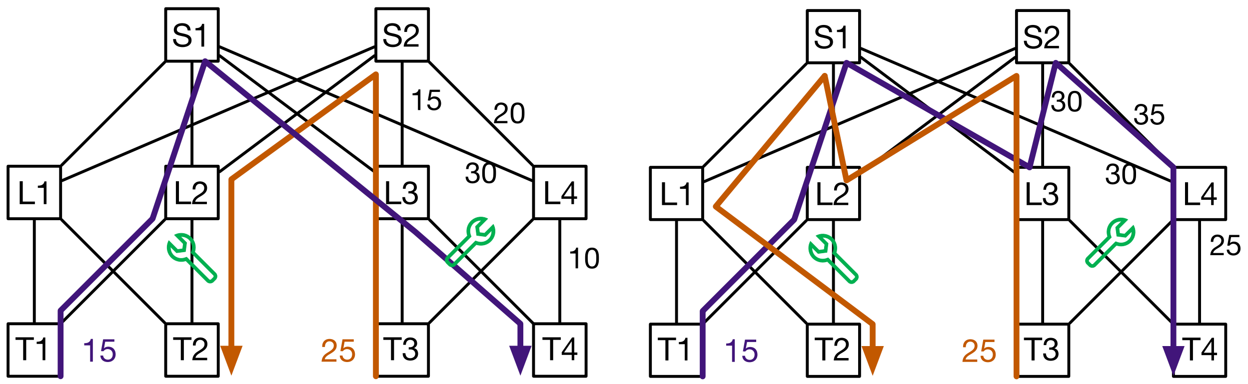
To avoid possible PFC deadlock problem, one safe congestion-free reconfiguration plan is to replace the two links one by one. Only using one non-shortest path will not introduce cyclic buffer dependency into the network.

2.2.2 *Deadlock Under Non-tree Based DCNs*

As shown in Fig. 3(a), in this example we consider a 4-node network **N**. Fig. 3(b)-(c),(e)-(f) are four spanning trees **T1-T4** which specify the routing paths that can be used in **N**. For example, path p1 is a legal routing path specified in T1.

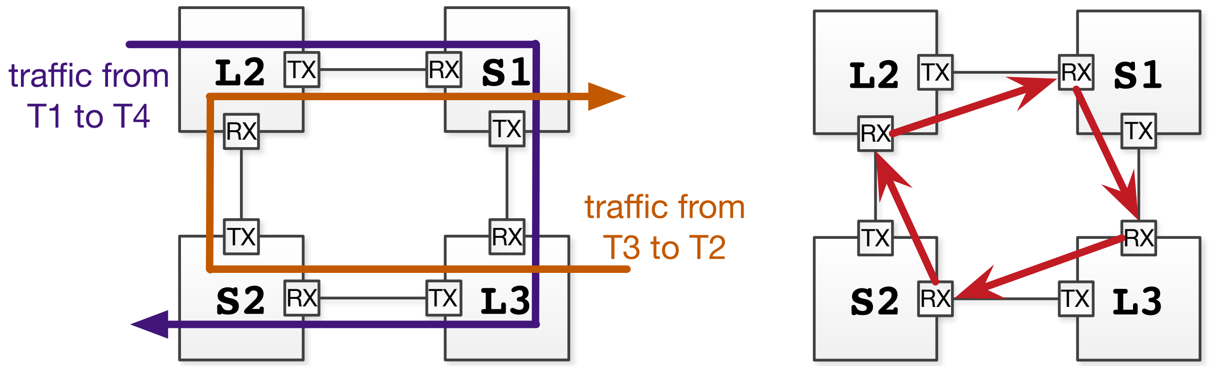
Let $\mathbf{R}_i$ be the set of paths specified in tree **Ti**. Let $\mathbf{R}_s = \mathbf{R}_1 \cup \mathbf{R}_2$, and $\mathbf{R}_t = \mathbf{R}_3 \cup \mathbf{R}_4$. It is easy to check both $\mathbf{R}_s$ and $\mathbf{R}_t$ are deadlock-free routing functions. Initially, $\mathbf{R}_s$ are used as the routing function of **N**. Due to the failure of link S2-S3, switch S3 becomes unreachable. To maintain the connectivity of **N**, we can perform a routing reconfiguration to transition from $\mathbf{R}_s$ to $\mathbf{R}_t$.

During the reconfiguration process, if path p2 in **T3** and



(a) Routing state before reconfiguration.

(b) Routing state after reconfiguration.



(c) Traffic among switches L2, L3, S1 and S2.

(d) Buffer dependencies among ingress queues.

Figure 2: Reconfiguration-induced deadlock case.

path p3 in **T4** are added to the routing function before path p1 in **T1** is removed, a cyclic buffer dependency will be generated. This may cause a PFC deadlock as we explained in Sec. 2.1.

In Fig. 3(d), we present three possible deadlock-free reconfiguration schmes. The first scheme is to remove all the paths in **T1** and **T2** before adding any new paths in **T3** and **T4**. This scheme will lead to a slow reconfiguration process as all the operations of adding new paths are delayed by the operations of removing old paths.

The second scheme only requires path p1 is removed before paths p2 and p3 are added. All the other paths not mentioned can be updated freely without any order constraint. Hence the speed of routing reconfiguration can be improved. The third scheme is an optimized reconfiguration scheme in terms of imposing minimum order constraints on the update actions. The intuition here is that as long as paths p1, p2 and p3 do not take effect at the same, deadlock-free can be well guaranteed.

While for this example it may seem easy to find a deadlock-free reconfiguration scheme that requires minimum order constraints, in general it is difficult as there are combinatorial such schemes to be checked.

2.3 Measurement of Rule Update Time

In this part, we demonstrate that adding order constraints to the update of switch rules will significantly prolong the reconfiguration process.

3. SOLUTION

In this part, we present our preliminary solution for achieving fast deadlock-free network update.

Enumerating all the possible orders of path updates would be computationally impossible as there are combinatorial such options. Hence we seek to find an efficient heuristic solution for minimizing the number of order constraints on update actions.

The intuition behind our solution is that *For each possible cycle in the buffer dependency graph, as long as we ensure that one old dependency link is removed before one new dependency link is added, the reconfiguration process is deadlock-free.*

The idea of our heuristic solution is as follows: First, for

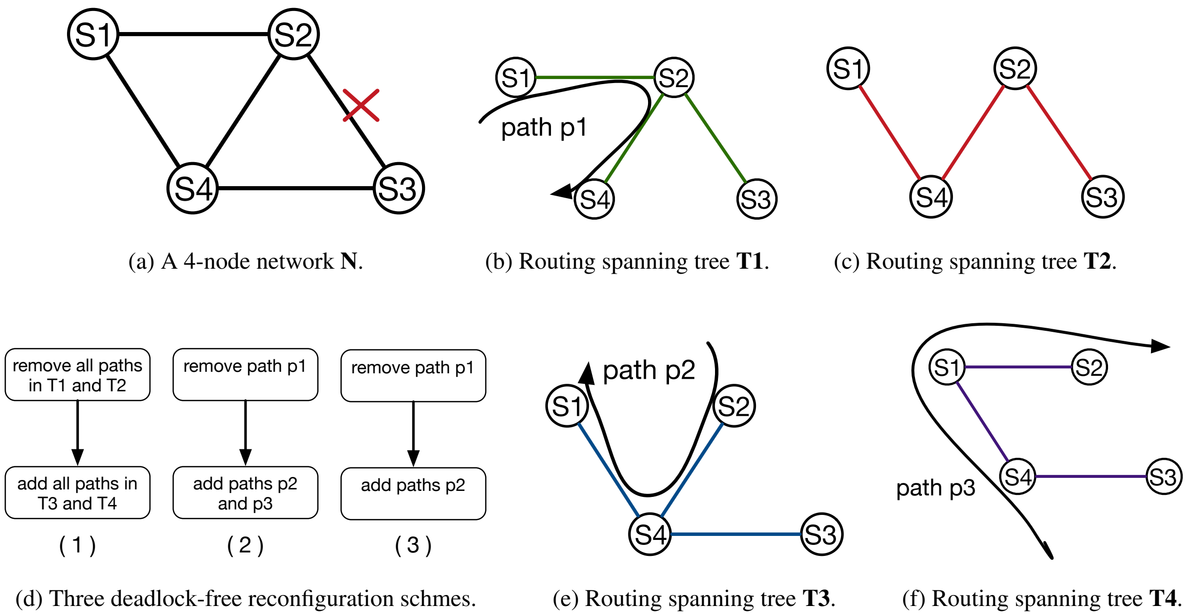


Figure 3: Reconfiguration-induced deadlock case.

each cycle in the buffer dependency graph, we enumerate all the sets of update actions that can exactly add or remove one dependency link from the graph. Then we pick up two minimum action sets A and B, where A can remove one dependency link for a given cycle while B can add an dependency link for the same cycle. The deadlock-free reconfiguration scheme our solution produces will ensure that A is finished before B starts to be configured.

4. EVALUATION

to be added. In this part, we evaluate the performance of our solution via simulations.

Topology: 4-level Fat-tree, HyperX, Jellyfish, etc.

Model of switch rule update: parallel update, sequential update, etc. We also need to model the delay of control messages in our simulator.

5. RELATED WORKS

to be added.

6. REFERENCES

[1] Y. Zhu, H. Eran, D. Firestone, C. Guo, M. Lipshteyn, Y. Liron, J. Padhye, S. Raindel, M. H. Yahia, and M. Zhang, "Congestion control for large-scale rdma deployments," ser. SIGCOMM '15.

[2] R. Mittal, V. T. Lam, N. Dukkipati, E. Blem, H. Wassel, M. Ghobadi, A. Vahdat, Y. Wang, D. Wetherall, and D. Zats, "Timely: Rtt-based congestion control for the datacenter," ser. SIGCOMM '15.

[3] "Rdma over converged ethernet (roce)," http://www.mellanox.com/page/products_dyn?product_family=79.

[4] "Ieee. 802.11qbb," ser. Priority based flow control, 2011.

[5] R. Mittal, V. T. Lam, N. Dukkipati, E. Blem, H. Wassel, M. Ghobadi, A. Vahdat, Y. Wang, D. Wetherall, and D. Zats, "Brent, stephens and alan, l. cox and ankit, singla and john, carter and colin, dixon and wesley, felter," ser. INFOCOM '14.

[6] J. Flich, T. Skeie, A. Mejia, O. Lysne, P. Lopez, A. Robles, J. Duato, M. Koibuchi, T. Rokicki, and J. C. Sancho, "A survey and evaluation of topology-agnostic deterministic routing algorithms," *IEEE Transactions on Parallel and Distributed Systems*, 2012.

[7] M. Gerla and L. Kleinrock, "Flow control: A comparative survey," *IEEE Transactions on Communications*, 1980.

[8] M. Karol, S. J. Golestani, and D. Lee, "Prevention of deadlocks and livelocks in lossless backpressured packet networks," *IEEE/ACM Transactions on Networking*, 2003.