6. EXPERIENCES

6.1 RDMA Deployment

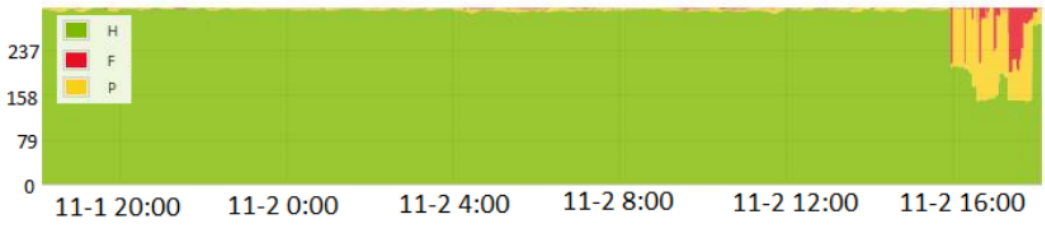
> RoCEv2 was a new technology to us when we began this work three years ago. We were unaware of any large-scale RoCEv2 deployment at that time. Though the benefits (zero packet drops, low latency, and low CPU overhead) were attractive, we were concerned about the maturity of RoCEv2. We devised a step-by-step procedure to onboard RDMA. For the first step, we built a small lab network with tens of servers. This step helped us eliminate most of the bugs at early stage. In the second step, we used test clusters to improve the maturity of RoCEv2. The test clusters' setup and management were the same as their production counterparts. In the third step, we enabled RDMA in production networks at ToR level only. In the fourth step, we enabled PFC at the Podset level, i.e., we enabled PFC in the ToR and Leaf switches within the Podsets. In the last step, we enabled PFC up to the Spine switches. In every step when we carried out deployment in production, we followed our safe deployment procedure to enable RDMA through several phases in our global data centers. This step-by-step procedure turned out to be effective in improving the maturity of RoCEv2. The RDMA transport livelock and most of the bugs were detected in lab tests. The PFC deadlock and slow-receiver symptom were detected in the test clusters. Only the NIC PFC pause frame storm and a few other bugs hit our production networks. Using the same management and monitoring for both the test clusters and the production networks turned out to be invaluable. It made our life easier as the test clusters were always well managed. At the same time, it let us thoroughly test RoCEv2 as well as the management and monitoring systems before RoCEv2 went into production.

当我们三年前开始这项工作时，RoCEv2 对我们来说是一项新技术。当时我们并不知道有任何大规模的 RoCEv2 部署。尽管其优势（零丢包、低延迟和低 CPU 开销）很有吸引力，但我们对 RoCEv2 的成熟度感到担忧。
我们设计了一个加载 RDMA 的分步程序。第一步，我们建立了一个包含数十台服务器的小型实验室网络。这一步帮助我们在早期阶段消除了大部分错误。第二步，我们使用测试集群来提高RoCEv2的成熟度。测试集群的设置和管理与生产集群相同。第三步，我们仅在 ToR 级别的生产网络中启用 RDMA。第四步，我们在 Podset 级别启用 PFC，即在 Podset 内的 ToR 和 Leaf 交换机中启用 PFC。在最后一步中，我们启用了 Spine 交换机的 PFC。
在我们进行生产部署的每一步中，我们都遵循安全部署程序，以便在我们的全球数据中心的多个阶段中启用 RDMA。事实证明，这一分步过程有效提高了 RoCEv2 的成熟度。
RDMA 传输活锁和大多数错误都是在实验室测试中检测到的。在测试集群中检测到PFC死锁和慢接收器症状。只有 NIC PFC 暂停帧风暴和其他一些错误影响了我们的生产网络。事实证明，对测试集群和生产网络使用相同的管理和监控是非常有价值的。由于测试集群始终得到良好的管理，这让我们的生活变得更加轻松。同时，它也让我们在RoCEv2投入生产之前对RoCEv2以及管理和监控系统进行了全面的测试。

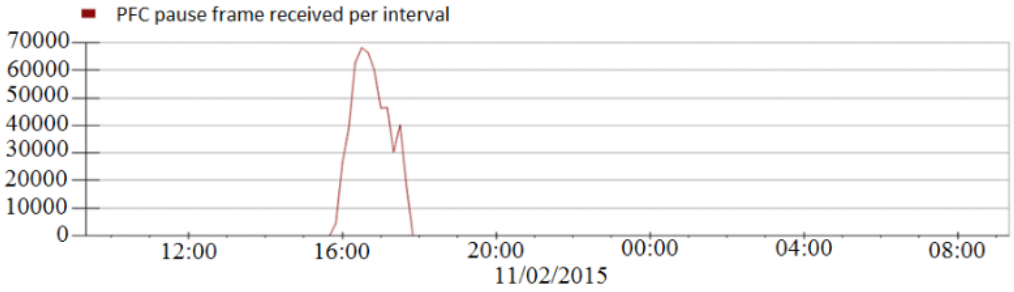
6.2 Incidents

1. NIC PFC storm (网卡PFC风暴)

> The following is one of the few NIC PFC storm incidents we have encountered. In this incident, one of our customers experienced a service availability issue. Many of their servers became unavailable as shown in Figure 9(a). At the same time, we observed that many of the servers were continuously receiving large number of PFC pause frames as shown by our monitoring system in Figure 9(b). The y-axis shows the number of PFC pause frames sent/received in every five minutes. We were able to trace down the origin of the PFC pause frames to a single server. That server was unresponsive and was in Failing (F) state as detected by our data center management system. But from the connected ToR switch, we could observe the number of pause frames from the server was always increasing, at more than two thousands pause frames per second.
> We also observed that the server was not sending or receiving any data packets. After we power-cycled that server, the server came back up and the pause frames were gone. NIC PFC storms happened very infrequently. With hundreds of thousands of servers in production, the number of the NIC PFC storm events we have experienced is still single digit. Nonetheless, once NIC PFC storm happens, the damage is huge due to the PFC pause frame propagation. As we can see from this incident, half of our customers servers were affected and put into non healthy state. After we put the NIC PFC storm prevention watchdogs at both the servers and the ToR switches, we did not experience NIC PFC storms anymore.



(a) Server availability reduction. H (healthy), F (failing), and P (probation) are server states.



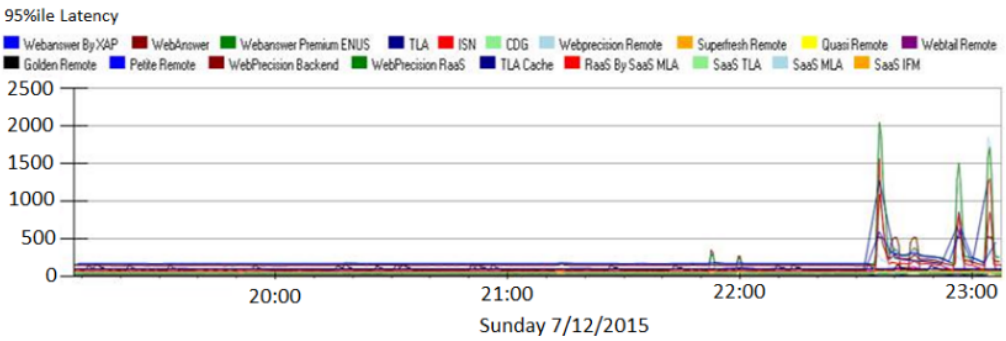
(b) The PFC pause frames received by the servers.

以下是我们遇到的为数不多的NIC PFC风暴事件之一。在此事件中，我们的一位客户遇到了服务可用性问题。他们的许多服务器变得不可用，如图 9(a) 所示。同时，我们观察到许多服务器连续接收大量 PFC 暂停帧，如图 9(b) 中我们的监控系统所示。 Y 轴显示每五分钟发送/接收的 PFC 暂停帧的数量。我们能够将 PFC 暂停帧的来源追溯到单个服务器。我们的数据中心管理系统检测到**该服务器没有响应并且处于故障 (F) 状态**。但从连接的 ToR 交换机中，我们可以观察到来自服务器的暂停帧数量始终在增加，每秒超过两千个暂停帧。

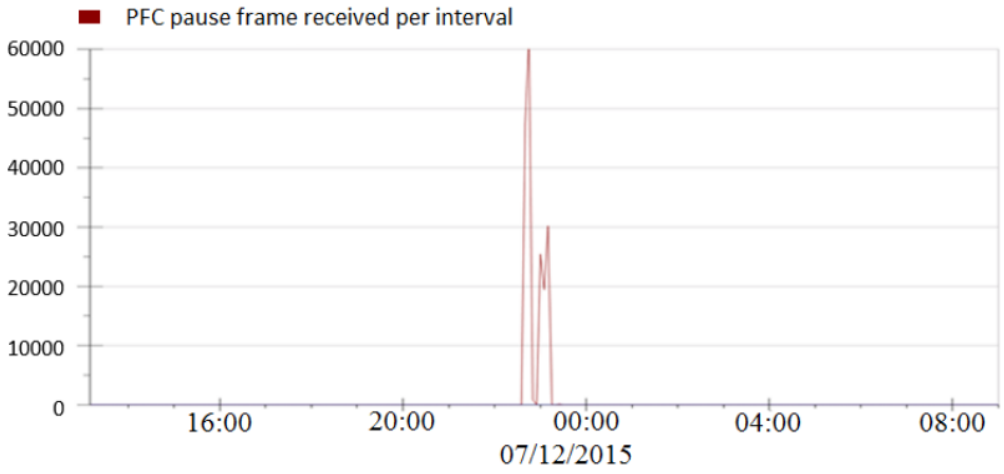
我们还观察到服务器没有发送或接收任何数据包。在我们重新启动该服务器后，服务器恢复正常并且暂停帧消失了。
NIC PFC 风暴很少发生。在数十万台服务器投入生产的情况下，我们经历的NIC PFC风暴事件数量仍然是个位数。
然而，一旦网卡PFC风暴发生，由于PFC暂停帧的传播，造成的损害是巨大的。从这次事件中我们可以看到，我们一半的客户服务器受到影响并进入非健康状态。当我们在服务器和 ToR 交换机上安装 NIC PFC 风暴防护看门狗后，我们就不再遇到 NIC PFC 风暴了。

2. Switch buffer misconfiguration(交换机缓冲区配置错误)

> The ToR and Leaf switches we use have a small and limited buffer size of 9MB or 12MB. To better utilize the scarce buffer space, we need to enable dynamic buffer sharing. In dynamic buffer sharing, the ports allocate memory from a shared buffer pool. The shared buffer allocation per port per traffic class is controlled by a parameter called α. As long as α × U B > Bp,i, where U B is the unallocated shared buffer size and Bp,i is the allocated buffer size for traffic class i of ingress port p, we can allocate memory from the shared buffer for traffic class i from ingress port p. Hence a large α can help reduce the chance of PFC pause frames from been generated. But a large α may cause imbalanced and unfair buffer allocation. We have found that the default α value (α = 1 16) for a type of ToR switch worked well in our production network. When we onboarded a new type of switch from the same switch provider, we took it for granted that it would use the same default settings as before. Then in the midnight of 07/12/2015, we ran into an incident. As shown in Figure 10(a), the latencies of many latency-sensitive services increased dramatically. Also we have observed that many servers were receiving a large number of PFC pause frames, up to 60000 pause frames in 5 minutes (Figure 10(b)). Further analysis revealed the origins of the pause frames. The pause frames were generated by two ToR switches, then propagated into the rest of the network, and affected thousands of servers. Why there were so many pause frames been generated? There were two reasons. The first was the incast traffic pattern. These two ToR switches hosted many chatty servers, which sent queries to more than one thousand servers simultaneously. Once the responses came back to the chatty servers, incast happened, which created network congestion condition for PFC pause frame generation. The second reason was that we found the α value of the new type of ToR switch was changed to 1 64 , though these two types of switches were from the same provider. A much smaller α made the dynamic buffer allocated to the congested ingress ports much smaller. Hence the PFC pause frames could be triggered much more easily. We could not change the traffic pattern, so we tuned the α value back to 1 16 for these switches. The lesson we learned from this incident is that PFC pause frames did propagate and cause collateral damage in our production network. To reduce the damage, we need to reduce PFC pause frames from being generated. Our work on the NIC PFC storm and the slow-receiver symptom prevent servers from been generating pauses. Moreover, parameter tuning of the dynamic buffer sharing and the per-flow based DCQCN [42] congestion control reduce the pauses generated by the switches.



(a) Services latency increase caused by the PFC pause frame propagation. Every color here represents an impacted service.



(b) The PFC pause frames received by the servers.

我们使用的 ToR 和 Leaf 交换机的缓冲区大小较小且有限，为 9MB 或 12MB。为了更好地利用稀缺的缓冲区空间，我们需要启用动态缓冲区共享。在动态缓冲区共享中，端口从共享缓冲池分配内存。每个流量类别每个端口的共享缓冲区分配由称为 α 的参数控制。只要 α × U B > Bp,i，其中 U B 是未分配的共享缓冲区大小，Bp,i 是入口端口 p 的流量类别 i 的已分配缓冲区大小，我们就可以从共享缓冲区中为流量类别 i 分配内存：入口端口 p.
因此，较大的 α 有助于减少生成 PFC 暂停帧的机会。但较大的α可能会导致缓冲区分配不平衡和不公平。
我们发现某种 ToR 交换机的默认 α 值 (α = 1/16) 在我们的生产网络中运行良好。
当我们从同一交换机提供商处安装新型交换机时，我们理所当然地认为它将使用与以前相同的默认设置。然后在 2015 年 7 月 12 日午夜，我们遇到了一起事件。如图 10(a) 所示，许多延迟敏感服务的延迟急剧增加。我们还观察到许多服务器收到大量 PFC 暂停帧，5 分钟内多达 60000 个暂停帧（图 10(b)）。
进一步的分析揭示了暂停帧的起源。暂停帧由两个 ToR 交换机生成，然后传播到网络的其余部分，并影响了数千台服务器。
为什么会产生这么多暂停帧？有两个原因: 第一个是 incast 流量模式。这两个 ToR 交换机托管了许多聊天服务器，这些服务器同时向一千多个服务器发送查询。一旦响应返回到聊天服务器，就会发生 incast，这为 PFC 暂停帧的生成创造了网络拥塞条件。第二个原因是我们发现新型ToR交换机的α值更改为 1/64，尽管这两种类型的交换机来自同一提供商。小得多的 α 使得分配给拥塞入口端口的动态缓冲区小得多。
因此，可以更容易地触发 PFC 暂停帧。我们无法更改流量模式，因此我们将这些交换机的 α 值调整回 1/16。
我们从这次事件中吸取的教训是，PFC 暂停帧确实会传播并在我们的生产网络中造成附带损害。为了减少损害，我们需要减少 PFC 暂停帧的生成。我们针对 NIC PFC 风暴和缓慢接收器症状所做的工作可防止服务器产生暂停。
此外，动态缓冲区共享的参数调整和基于每个流的 DCQCN [42] 拥塞控制减少了交换机产生的暂停。

6.3 Lessons learned and discussion

> During the three years period of designing, building, and deploying RoCEv2, we have learned several lessons which we share as follows. Deadlock, livelock, and PFC pause frames propagation did happen. The PFC deadlock we met was a surprise to us, as we once believed that our Clos-based network topology was free of cyclic buffer dependency hence free of deadlock. We did not expect the slowserver symptom, though we were fully aware that PFC backpressure can cause PFC pause frame propagation in the network. We did not foresee the RDMA transport livelock either. The lesson we learned is that a design works in theory is not enough, as there may be many hidden details which invalidate the design. We have to use well designed experiments, test clusters, and staged production deployments, to verify the designs and to unveil the unexpected facets methodologically. NICs are the key to make RDMA/RoCEv2 work. Most of the RDMA/RoCEv2 bugs we ran into were caused by the NICs instead of the switches. We spent much more time on the NICs than on the switches. In hindsight, this happened for two reasons. The first reason is because the NIC implements the most complicated parts of the RDMA functionalities, including the RDMA verbs and the RDMA transport protocol. As a comparison, the switch side functionalities are relatively simple (e.g., PFC implementation) or well tested (e.g., ECN implementation). The second reason is that the NICs we use are resource constrained. The NIC leverages the server's DRAM to store its data structures and uses its own local memory as the cache. Cache management then becomes a big part of the NIC and introduces bugs as well as performance bottlenecks, e.g., the slowreceiver symptom. Be prepared for the unexpected. Our experiences of running one of the largest data center networks in the world taught us that network incidents happen. From day one when we began to work on RoCEv2, we put RDMA/RoCEv2 management and monitoring as an indispensable part of the project. We upgraded our management and monitoring system for RDMA status monitoring and incidents handling at the same time when we worked on the DSCP-based PFC design and the safety and performance bugs. As a result, when our customers began to use RDMA, the RDMA management and monitoring capabilities were already in production. This RDMA management and monitoring system is essential for RDMA health tracking and incident troubleshooting. It helped us detect, localize, and rootcause the RoCEv2 bugs and incidents as we have shown in Sections 6.2 and 4. Is lossless needed? RoCEv2 depends on a lossless network to function well. In this work, we have demonstrated that we indeed can build a lossless network using PFC, and all the real-world scalability and safety challenges can be addressed. Looking forward into the future, the question we would like to ask is: do we really need a lossless network to get the benefits of RoCEv2? Given the progress on programmable hardware, e.g., FPGA and FPGA integrated CPU [8], it may become feasible and economical to build much faster and more advanced transport protocols and forward error correction algorithms directly in commodity hardware, hence relieving RoCEv2 from been depending on lossless network.

在设计、构建和部署 RoCEv2 的三年期间，我们吸取了一些经验教训，现分享如下。

1. 死锁、活锁和 PFC 暂停帧传播确实发生了。
   我们遇到的 PFC 死锁让我们感到惊讶，因为我们曾经相信基于 Clos 的网络拓扑不存在循环缓冲区依赖，因此不会出现死锁。尽管我们充分意识到 PFC 背压可能会导致 PFC 暂停帧在网络中传播，但我们并没有预料到会出现服务器缓慢的症状。我们也没有预见到 RDMA 传输活锁。我们学到的教训是，光有理论上的设计是不够的，因为可能存在许多隐藏的细节，导致设计无效。我们必须使用精心设计的实验、测试集群和分阶段生产部署来验证设计并从方法上揭示意想不到的方面。
2. NIC 是 RDMA/RoCEv2 发挥作用的关键。
   我们遇到的大多数 RDMA/RoCEv2 错误都是由 NIC 而不是交换机引起的。我们在网卡上花费的时间比在交换机上花费的时间多得多。事后看来，发生这种情况有两个原因。第一个原因是因为 NIC 实现了 RDMA 功能中最复杂的部分，包括 RDMA 动词和 RDMA 传输协议。相比之下，交换机端功能相对简单（例如，PFC 实现）或经过良好测试（例如，ECN 实现）。第二个原因是我们使用的网卡资源有限。 NIC 利用服务器的 DRAM 来存储其数据结构，并使用自己的本地内存作为缓存。然后，缓存管理成为 NIC 的重要组成部分，并引入错误和性能瓶颈，例如接收器速度慢的症状。
3. 为意外情况做好准备。
   我们运行世界上最大的数据中心网络之一的经验告诉我们，网络事件时有发生。从我们开始研究RoCEv2的第一天起，我们就把RDMA/RoCEv2管理和监控作为项目中不可或缺的一部分。在处理基于 DSCP 的 PFC 设计以及安全和性能错误时，我们同时升级了 RDMA 状态监控和事件处理的管理和监控系统。因此，当我们的客户开始使用RDMA时，RDMA管理和监控功能已经投入生产。该 RDMA 管理和监控系统对于 RDMA 运行状况跟踪和事件故障排除至关重要。它帮助我们检测、定位 RoCEv2 错误和事件并找出根本原因，如第 6.2 节和第 4 节所示。
4. 是否真的需要无损?
   RoCEv2 依赖于无损网络才能正常运行。在这项工作中，我们证明了我们确实可以使用 PFC 构建无损网络，并且可以解决所有现实世界的可扩展性和安全性挑战。展望未来，我们想问的问题是：
   我们真的需要无损网络才能获得RoCEv2的好处吗？鉴于可编程硬件（例如 FPGA 和 FPGA 集成 CPU [8]）的进步，直接在商用硬件中构建更快、更先进的传输协议和前向纠错算法可能变得可行且经济， 从而使 RoCEv2 不再依赖于无损网络。

- 基于 Clos 的网络拓扑

Clos 网络拓扑，也称为 Clos 网络结构或 Clos 网络，是一种网络拓扑结构，最初由美国工程师和数学家 Charles Clos 在 1953 年提出。这种拓扑结构主要用于构建大规模的交换机和路由器网络，以支持高容量和可伸缩性需求。

Clos 网络拓扑由三个关键要素构成：输入层、中间层和输出层。它通常用于构建数据中心网络，以满足不同网络流量的需求。以下是 Clos 网络拓扑的基本原理：

1. 输入层（Ingress Layer）： 输入层包含一组交换机或路由器，通常用于接受来自网络中其他设备的数据包。这些交换机将输入流量分发到中间层的交换机。
2. 中间层（Intermediate Layer）： 中间层包含多个交换机，这些交换机通常连接到输入层和输出层的设备。它们用于路由和转发数据包，以支持网络中的通信。中间层的交换机通常是多个等级的，允许构建大规模网络。
3. 输出层（Egress Layer）： 输出层包含一组交换机或路由器，用于将数据包发送到目标设备或网络。输出层的设备通常接受来自中间层的数据包，并将它们传送到最终目的地。

Clos 网络的优势包括高容量、可伸缩性和容错性。由于其分层结构，Clos 网络可以轻松扩展以支持更多的设备和流量，而且即使其中的某些交换机或链路发生故障，仍能够保持连通性。这使得它特别适合用于大规模数据中心和云计算环境，其中高性能和可靠性至关重要。
总之，Clos 网络拓扑是一种用于构建大规模高性能网络的有效结构，它通过层次化的交换机布局和分散的数据流量管理来满足不同网络需求。

-为什么Clos 网络拓扑不存在循环缓冲区依赖?

Clos 网络拓扑通常被设计为不存在循环缓冲区依赖，这是因为 Clos 网络的结构和路由算法能够有效地避免这种情况，从而确保数据包不会在网络中形成无限循环。
这种无循环缓冲区依赖的特性可以追溯到 Clos 网络的基本原则和路由算法：

1. 非阻塞结构：Clos 网络的基本原则之一是非阻塞性，也就是说，它被设计为能够同时支持所有输入端口与输出端口之间的通信，而不会出现阻塞情况。这一特性要求在设计 Clos 网络时，确保中间层的交换机能够提供足够的路径和带宽，以避免数据包在网络中发生冲突和竞争，从而避免循环缓冲区依赖。
2. 路由算法：Clos 网络通常使用适当的路由算法，以确保数据包被正确引导到其目标。这些路由算法通常会考虑网络的结构，避免数据包在中间层的交换机之间循环。通过正确设计和配置路由算法，可以确保数据包按照正确的路径到达目的地，而不会出现循环路径。
3. 网络设计：Clos 网络的设计需要综合考虑输入层、中间层和输出层之间的连接和交换机配置。通过仔细的设计和规划，可以避免交换机之间的冲突和竞争，从而避免循环缓冲区依赖。
   总之，Clos 网络的非阻塞性和正确设计的路由算法是确保不存在循环缓冲区依赖的关键因素。这些特性使得 Clos 网络成为一种有效的网络拓扑结构，适用于需要高性能、可伸缩性和低延迟的应用，如大规模数据中心和云计算环境。

-循环缓冲区依赖是什么?

对于一些网络拓扑和路由算法，循环缓冲区依赖是一个可能出现的问题。
def(循环缓冲区依赖): 指的是数据包在网络中形成循环路径，而这些路径可能导致数据包需要多次经过相同的网络元素（例如交换机或路由器），从而产生缓冲区依赖和死锁。

Clos 网络通常能够有效地避免循环缓冲区依赖的原因是，它的结构和路由算法被设计为确保数据包在网络中不会无限循环。具体来说，Clos 网络的中间层结构通常是非阻塞的，这意味着可以同时支持所有输入和输出端口之间的通信，而不会出现阻塞情况。此外，Clos 网络的路由算法通常会考虑网络拓扑，确保数据包按照正确的路径前进，而不会形成循环路径。

虽然 Clos 网络的设计通常能够有效避免循环缓冲区依赖，但在实际部署中，仍然需要进行仔细的规划和配置，以确保网络操作正常，特别是在高负载和故障情况下。循环缓冲区依赖和死锁问题可能会在一些其他网络拓扑和路由算法中出现，因此网络设计者需要谨慎考虑这些问题，以确保网络的可靠性和性能。