# 5. RDMA In Production

We added new management and monitoring capabilities to debug the various RDMA and PFC safety issues described in Section 4, and to detect RDMA related bugs and incidents. We now discuss these new capabilities which include the RDMA/PFC configuration monitoring, the PFC pause frame and lossless traffic monitoring, and the active RDMA latency monitoring. We also present the latency and throughput measurements.

我们添加了新的管理和监控功能来调试第 4 节中描述的各种 RDMA 和 PFC 安全问题，以检测 RDMA 相关的错误和事件。我们现在讨论这些新功能，包括 RDMA/PFC 配置监控、PFC 暂停帧和无损流量监控以及主动 RDMA 延迟监控。我们还介绍了延迟和吞吐量测量。

## 5.1 Configuration management and monitoring（配置管理和监控）

> To enable RDMA, we need to configure PFC at the switch side, and RDMA and PFC at the server side. At the switch side, the PFC configuration is part of the QoS configuration. The PFC configuration has a global part which reserves buffer size, classifies packets into different traffic classes based on the DSCP value, maps different traffic classes into different queues, and assigns different bandwidth reservations for different queues. The PFC configuration also has a per port part which enables PFC for every individual physical port. At the server side, there are configurations to enable/disable RoCEv2, PFC configuration, DCQCN configuration, and traffic configuration. In traffic configuration, users specify which type of traffic they would like to put into PFC protection. The specification is based on the destination transport port which is similar to the TCP destination port. We have a configuration monitoring service to check if the running configurations of the switches and the servers are the same as their desired configurations. Our RDMA management and monitoring service handles the complexities introduced by the combinations of multiple switch types, multiple switch and NIC firmware versions, and different configuration requirements for different customers.

为了启用RDMA，我们需要在交换机端配置PFC，在服务器端配置RDMA和PFC。
在交换机侧，PFC配置是QoS配置的一部分。PFC配置有一个全局部分，它保留缓冲区大小，根据DSCP值将数据包分类为不同的流量类别，将不同的流量类别映射到不同的队列，并为不同的队列分配不同的带宽预留。
PFC 配置还具有每个端口的部分，可为每个单独的物理端口启用 PFC。
在服务器端，有启用/禁用RoCEv2、PFC配置、DCQCN配置和流量配置的配置。在流量配置中，用户指定他们想要将哪种类型的流量置于 PFC 保护中。该规范基于与 TCP 目标端口类似的目标传输端口。
我们有配置监控服务来检查交换机和服务器的运行配置是否与其所需的配置相同。我们的 RDMA 管理和监控服务可处理多种交换机类型、多种交换机和 NIC 固件版本以及不同客户的不同配置要求组合带来的复杂性。

### -QoS配置：

QoS（Quality of Service，服务质量）配置是一种网络管理和配置方法，用于优化网络上的数据传输，以确保不同类型的数据流在网络中得到适当的带宽、延迟、丢包率和优先级处理。
QoS配置的主要目的是为了提供更好的性能和可靠性，确保关键数据或应用程序能够在网络上获得更高的服务质量。

QoS配置通常涉及以下方面的设置和控制：

1. **带宽分配**：通过QoS配置，您可以为不同类型的流量（如语音、视频、数据）分配带宽。这可以确保关键应用程序获得足够的带宽，以避免拥塞和性能下降。
2. **流量优先级**：QoS允许您为不同类型的流量设置优先级。例如，实时通信（如VoIP电话或视频会议）可以被赋予更高的优先级，以确保其低延迟和高可用性。
3. **拥塞管理**：QoS配置还可用于管理网络拥塞。当网络拥塞时，QoS可以帮助决定哪些流量将被丢弃或延迟，以减轻拥塞情况。
4. **流量分类**：QoS配置通常包括流量分类，以便识别和处理不同类型的数据。这可以通过协议、端口、IP地址或其他特征进行分类。
5. **队列管理**：QoS允许您配置不同的数据队列，以便根据优先级和策略对数据包进行排队和处理。
6. QoS配置通常在路由器、交换机和网络设备上进行，以确保网络上的流量得到适当的管理。这对于企业网络、数据中心、云服务提供商以及需要满足特定性能需求的各种应用程序都非常重要。通过QoS配置，网络管理员可以更好地控制和优化网络性能，以满足不同应用程序和用户的需求。

### -启用/禁用RoCEv2：

RoCEv2 是 RoCE（RDMA over Converged Ethernet）技术的第二代版本。RoCEv2 通过以太网网络支持 RDMA 数据传输，这是一种高性能、低延迟的数据传输技术，通常用于数据中心、高性能计算和存储环境中。启用或禁用 RoCEv2 意味着您可以在网络设备上配置是否使用 RoCEv2 技术进行数据传输。

以下是一些关于启用和禁用 RoCEv2 的含义：

1. **启用 RoCEv2**：当您启用 RoCEv2 时，您配置网络设备以支持使用 RoCEv2 协议进行 RDMA 数据传输。这通常需要确保网络适配器、交换机和路由器都支持 RoCEv2，并且进行了适当的配置以启用 RDMA 功能。启用 RoCEv2 可能需要对网络进行额外的设置，以确保高性能、低延迟的数据传输。
2. **禁用 RoCEv2**：禁用 RoCEv2 意味着您将网络设备配置为不使用 RoCEv2 技术进行 RDMA 数据传输。数据将通过传统的以太网方式传输，而不是使用 RoCEv2 的优化数据传输。这可以是因为您的网络设备不支持 RoCEv2，或者您有其他特定的要求需要禁用 RoCEv2。

启用或禁用 RoCEv2 通常需要由网络管理员在网络设备上执行，以根据特定的网络需求和性能要求来管理数据传输。配置选择是否使用 RoCEv2 取决于您的网络环境、硬件支持以及特定的应用程序需求。 RoCEv2 可能提供更高的性能，但需要相应的硬件和配置支持。

### -传统的以太网传输方式：

传统的以太网传输方式是指使用标准以太网协议（如IEEE 802.3标准）进行数据传输的方式。这是计算机网络中最常见的数据传输方式，它使用经典的*以太网协议栈，通常以太网帧结构，以太网交换机*和*路由器*来进行通信。

以下是一些传统以太网传输方式的特点：

1. **以太网帧结构**：数据在传统以太网中被封装成以太网帧，这些帧包括*源*和*目的MAC地址*、*数据内容*以及*校验*和*其他控制信息*。帧的大小通常在 64 到 1518 字节之间。
2. **MAC 地址**：每个网络接口都有一个唯一的MAC地址，它用于标识设备。**在传统以太网中，数据帧根据目的MAC地址路由到正确的目标设备**。
3. **CSMA/CD**：传统以太网使用载波侦听多路访问/碰撞检测（CSMA/CD）协议来管理冲突和竞争。当多个设备尝试同时传输数据时，CSMA/CD协议用于避免冲突。
4. **速率**：传统以太网可以以不同的速率运行，包括10 Mbps（以太网）、100 Mbps（快速以太网）和1 Gbps（千兆以太网），以及更高速的变种。
5. **无差错性**：传统以太网通常依赖底层协议来检测和纠正错误。传输过程中可能会出现丢包或冲突。

这种传统的以太网传输方式通常适用于常规的数据通信，但在对于要求更高性能、低延迟和可靠性的应用，如**高性能计算、存储区域网络和远程直接内存访问（RDMA）等**情况下，其他网络技术（如RoCEv2）可能更为适用，因为它们提供了更高的性能和更低的延迟。

## 5.2 PFC pause frame and traffic monitoring

> Besides configuration monitoring, we have also built monitoring for the PFC pause frames and the two RDMA traffic classes. For pause frame, we monitor the number of pause frames been sent and received by the switches and servers. We further monitor the pause intervals at the server side. Compared with the number of pause frames, pause intervals can reveal the severity of the congestion in the network more accurately. Pause intervals, unfortunately, are not available for the switches we currently use. We have raised the PFC pause interval monitoring requirement to the switching ASIC providers for their future ASICs. For RDMA traffic monitoring, we collect packets and bytes been sent and received per port per priority, packet drops at the ingress ports, and packet drops at the egress queues. The traffic counters can help us understand the RDMA traffic pattern and trend. The drop counters help us detect if there is anything wrong for the RDMA traffic: normally no RDMA packets should be dropped.

除了配置监控之外，我们还构建了对 PFC 暂停帧和两个 RDMA 流量类别的监控。
对于暂停帧，我们监控交换机和服务器发送和接收的暂停帧的数量。我们还可以进一步监控服务器端的暂停间隔。
与暂停帧的数量相比，暂停间隔可以更准确地揭示网络拥塞的严重程度。
不幸的是，暂停间隔不适用于我们当前使用的开关！
我们向交换 ASIC 提供商未来的 ASIC 提出了 PFC 暂停间隔监控要求。
对于 RDMA 流量监控，我们收集每个端口每个优先级发送和接收的数据包和字节、入口端口的数据包丢失以及出口队列的数据包丢失。流量计数器可以帮助我们了解 RDMA 流量模式和趋势。丢弃计数器帮助我们检测 RDMA 流量是否存在任何问题：通常不应丢弃任何 RDMA 数据包。【丢包是传输出现问题的signal】

### -问题解释：为什么“暂停间隔不适用于我们当前使用的开关”?

"Pause intervals"（暂停间隔）通常是与“暂停帧”（pause frames）相关的一个概念，它用于管理网络设备之间的流量控制。在以太网网络中，暂停帧用于通知接收设备降低或停止数据传输，以防止拥塞。
然而，不是所有以太网交换机和网络设备都支持或提供关于暂停间隔的信息，下述可以解释为什么暂停间隔不适用于某些网络设备：

1. **设备不支持暂停间隔报告**：某些较旧或低成本的以太网交换机和网络设备可能没有支持报告或记录暂停间隔的功能。这意味着它们不能提供有关暂停间隔的信息。
2. **设备配置或监视的限制**：即使某些设备理论上支持暂停间隔，但它们可能被配置为不生成或记录这些信息，或者管理员可能没有启用相关的监视和报告功能。
3. **协议或固件版本**：有些设备可能需要特定的协议或固件版本才能支持暂停间隔的报告。使用较旧的设备或未经升级的设备可能无法提供这些信息。
4. **限制性因素**：特定的网络拓扑、配置或限制条件可能导致某些设备无法提供暂停间隔的信息。例如，某些网络设备在虚拟化环境中可能受到限制。

因此，如果某个网络中的交换机或网络设备不提供有关暂停间隔的信息，那么网络管理员可能需要依赖其他指标和工具来监视和评估网络性能，以确保网络正常运行。这可能包括查看暂停帧数量、带宽利用率、丢包率等其他可用的性能指标。

### -说人话解析：

1. 这段话描述了对 RDMA（远程直接内存访问）流量进行监测和分析的方法。
2. 在这种监测中，收集了每个端口、每个优先级（或重要性级别）的数据包和字节数的发送和接收情况，以及在流入端口和出口队列中发生的数据包丢弃情况。这些流量计数器可以帮助我们了解 RDMA 流量的模式和趋势。
3. 同时，丢包计数器帮助我们检测是否存在与 RDMA 流量有关的问题，正常情况下不应该出现 RDMA 数据包的丢失。丢包通常表示数据传输的中断或问题，因此检测丢包可以帮助管理员识别和解决与 RDMA流量相关的异常情况。

总之，这段话强调了对 RDMA 流量进行详细监测和计数的重要性，以便了解其性能和运行状况，并能够迅速发现并解决任何与 RDMA 数据传输相关的问题。

## 5.3 RDMA Pingmesh

> We have developed an active latency measurement service for RDMA similar to the TCP Pingmesh service [21]. We let the servers ping each other using RDMA and call the measurement system RDMA Pingmesh. RDMA Pingmesh launches RDMA probes, with payload size 512 bytes, to the servers at different locations (ToR, Podset, Data center) and logs the measured RTT (if probes succeed) or error code (if probes fail). From the measured RTT of RDMA Pingmesh, we can infer if RDMA is working well or not. Our RDMA management and monitoring took a pragmatic approach by focusing on configurations, counters, and end-to-end latency. We expect this approach works well for the future 100G or higher speed networks. RDMA poses challenges for packet-level monitoring due to the high network speed and NIC offloading, which we plan to address in our next step.

我们已经为 RDMA 开发了一种主动延迟测量服务，类似于 TCP Pingmesh 服务（主动探测活动的等待时间） [21]。
我们让服务器使用 RDMA 相互 ping（ping通），并将测量系统称为 RDMA Pingmesh。
RDMA Pingmesh 向不同位置（ToR、Podset、数据中心）的服务器启动负载大小为 512 字节的 RDMA 探针，并**记录测量的 RTT（如果探测成功）或错误代码（如果探测失败）**。
从 RDMA Pingmesh 测量的 RTT，我们可以推断 RDMA 是否工作良好。
我们的 RDMA 管理和监控采取了务实的方法，重点关注配置、计数器和端到端延迟。我们预计这种方法适用于未来 100G 或更高速度的网络。由于高网络速度和 NIC 卸载，RDMA 给数据包级监控带来了挑战，我们计划在下一步中解决这些问题。

### -TCP Pingmesh 服务：

TCP Pingmesh 服务是一种用于监测和评估计算机网络性能的服务。它的主要目的是**测试和记录网络中各个节点之间的 TCP 连接的性能和可用性**。下面是对 TCP Pingmesh 服务的详细解释：

1. **TCP 连接测试**：TCP Pingmesh 服务使用 TCP 协议来建立连接并发送数据包，以测量不同节点之间的网络连接性能。它通常会测试网络的**延迟、丢包率、吞吐量和带宽**等指标。
2. **多节点监测**：Pingmesh 通常涉及多个网络节点，这些节点可以是数据中心、云服务、边缘设备等。它通过在这些**节点之间建立多个 TCP 连接来监测它们之间的通信质量**。
3. **周期性测试**：Pingmesh 服务通常*以周期性的方式执行测试，以确保网络性能的持续监测*。这有助于及早发现网络问题和性能下降。
4. **性能指标记录**：Pingmesh 服务会记录各个节点之间的性能指标，以便网络管理员能够了解网络的健康状况。这些指标可以用于故障排除、网络优化和容量规划等目的。
5. **网络分析和故障排查**：Pingmesh 服务提供了对网络连接性能的详细数据，这对于分析网络问题和进行故障排查非常有用。当网络连接出现问题时，管理员可以查看 Pingmesh 数据以找出根本原因。

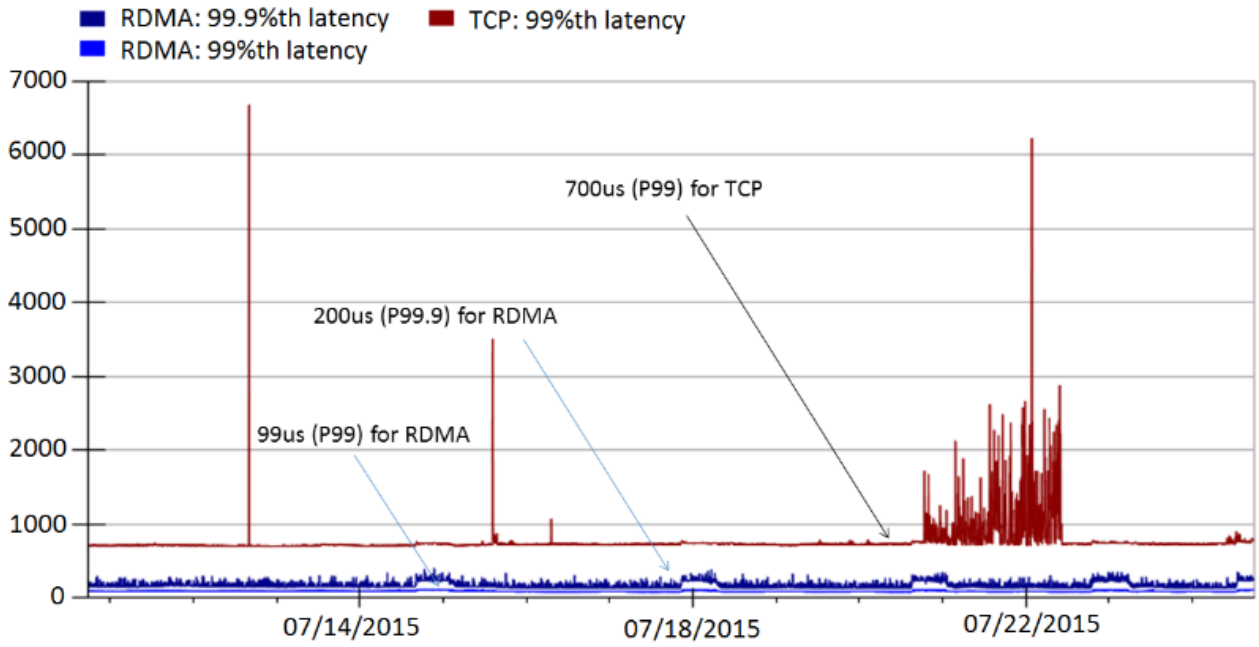
总之，TCP Pingmesh 服务是一种用于监测和评估网络性能的工具，**它通过定期测试不同节点之间的 TCP 连接，提供了有关网络性能和可用性的有用信息**。这有助于确保网络在高效运行、问题发生时及早发现和解决，以提供更好的用户体验和网络服务。

## 5.4 RDMA Performance

In what follows, we present the RDMA performance results in both testbed and production networks.

### Latency reduction:

> Figure 6 shows the end-to-end latency comparison of TCP and RDMA for a highlyreliable, latency-sensitive online service. This service has multiple instances in Microsoft global data centers and it has 20K servers in each data center. The measurements are from one of the data centers. At the time of measurement, half of the traffic was TCP and half of the traffic was RDMA. The RDMA and TCP latencies were all measured by Pingmesh. The latency measurements for both TCP and RDMA were for intraDC communications. Since the online service is latency sensitive, the peak traffic volume per sever was around 350Mb/s, and the aggregate server CPU load of the service was around 20% - 30% during the measurement. The network capacity between any two servers in this data center is several Gb/s. The network was not the bottleneck, but the traffic was bursty with the typical many-to-one incast traffic pattern. As we can see, the 99th percentile latencies for RDMA and TCP were 90us and 700us, respectively. The 99th percentile latency for TCP had spikes as high as several milliseconds. In fact, even the 99.9th latency of RDMA was only around 200us, and much smaller than TCP's 99th percentile latency. Although the network was not the bottleneck, TCP's latency was high at the 99th percentile. This is caused by the kernel stack overhead and occasional incast packet drops in the network. Although RDMA did not change the incast traffic pattern, it eliminated packet drops and kernel stack overhead. Hence it achieved much smaller and smoother high percentile latency than TCP.



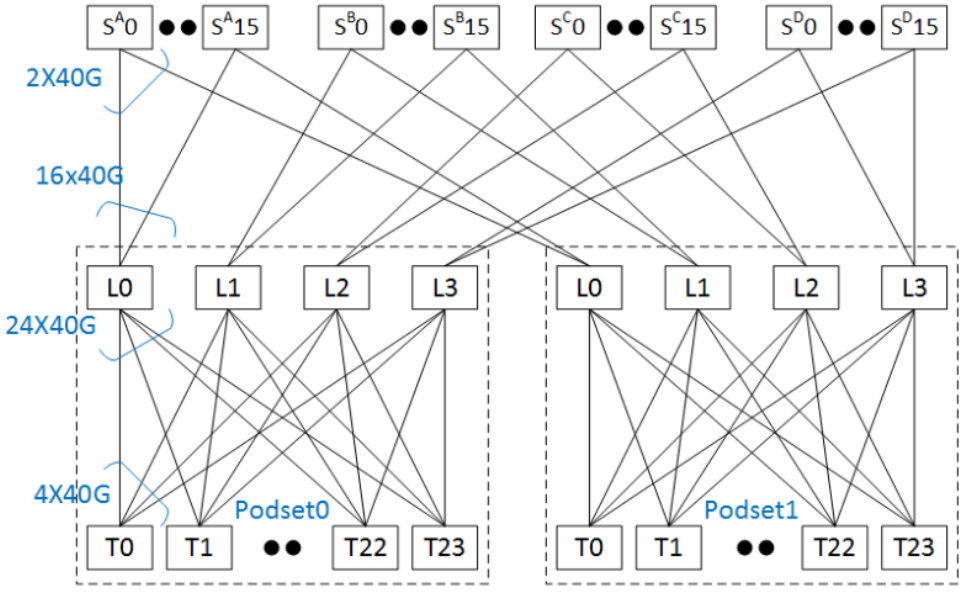

延迟减少：
图 6 显示了 TCP 和 RDMA 对于高度可靠、延迟敏感的在线服务的端到端延迟比较。该服务在微软全球数据中心有多个实例，每个数据中心有2万台服务器。测量结果来自其中一个数据中心。测量时，一半流量是 TCP，一半流量是 RDMA。 RDMA 和 TCP 延迟均由 Pingmesh 测量。
TCP 和 RDMA 的延迟测量均针对 DC 内通信。由于在线服务对延迟敏感，测量期间每台服务器的峰值流量约为 350Mb/s，该服务的服务器 CPU 总负载约为 20% - 30%。该数据中心内任意两台服务器之间的网络容量为数Gb/s。
网络不是瓶颈，但流量突发，是由于典型的多对一 incast 流量模式。
我们可以看到，RDMA 和 TCP 的 99% 延迟分别为 90us 和 700us。 TCP 第 99 个百分位数的延迟峰值高达几毫秒。事实上，即使是 RDMA 的 99.9% 延迟也只有 200us 左右，远小于 TCP 的 99% 延迟。尽管网络不是瓶颈，但 TCP 的延迟在第 99 个百分点处很高。这是由内核堆栈开销和网络中偶尔发生的 incast 数据包丢失引起的。
尽管 RDMA 没有改变 incast 流量模式，但它消除了数据包丢失和内核堆栈开销。因此，它实现了比 TCP 更小、更平滑的高百分位数延迟。

### 为什么RDMA不能改变incast流量模式?

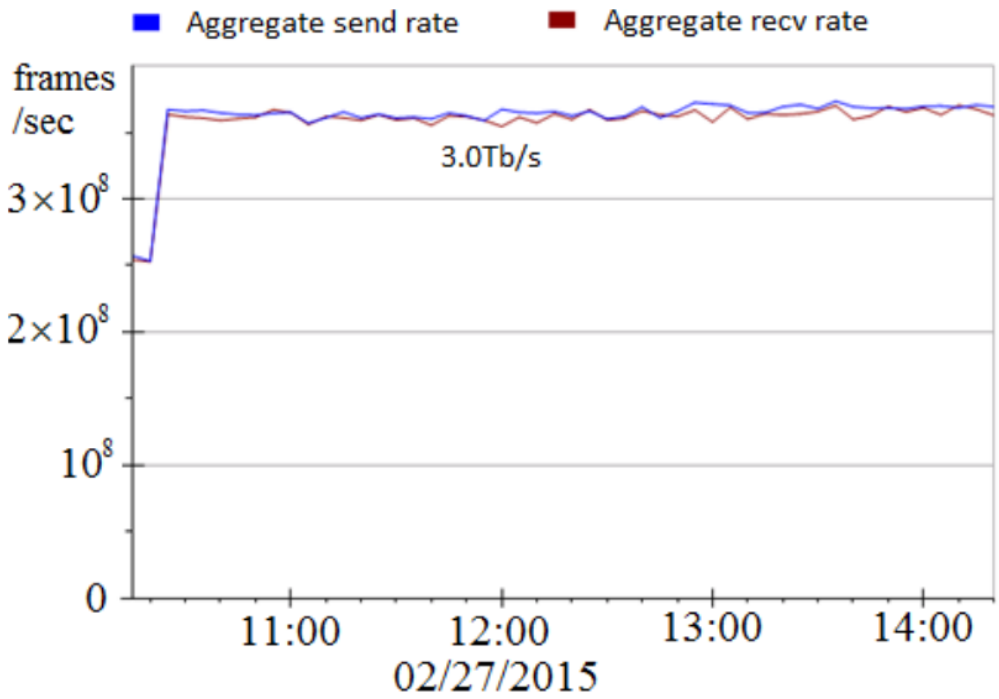
### Throughput：

> The following experiment shows the RDMA performance with hundreds of servers in a three-tier Clos network. We ran this experiment using two podsets after a data center was online but before it was handed to the customer – i.e. there is no customer traffic during the experiment. The network topology is shown in Figure 7(a). All the ports in the network are 40GbE. A podset is composed of 4 Leaf switches, 24 ToR switches, and 576 servers. Each ToR switch connects 24 servers. The 4 Leaf switches connect to a total of 64 Spine switches. The oversubscription ratios at the ToR and the Leaf are 6:1 and 3:2, respectively. The aggregate bandwidth between a podset and the Spine switch layer is 64x40Gb/s = 2.56Tb/s. We used a ToR-to-ToR traffic pattern. We paired the ToRs in the two podsets one by one. ToR i in podset 0 was paired with ToR i in podset 1. In each ToR, we selected 8 servers, and let each server estab- lish 8 RDMA connections to the corresponding server in the other ToR. All these RDMA connections needed to traverse the Leaf-Spine links. All the RDMA connections sent data as fast as possible. In total we had 3074 connections distributed among the 128 Leaf-Spine links, which were the bottlenecks in this experiment. Figure 7(b) shows the aggregate throughput measured from the servers. The unit of the y-axis is frames per second. The RDMA frame size is 1086 bytes with 1024 bytes as payload. The aggregate throughput is 3.0Tb/s. This is 60% network utilization of the total 5.12Tb/s network capacity. During the whole experiment, not a single packet was dropped. Every server was sending and receiving at 8Gb/s with the CPU utilization close to 0%. Since we use ECMP for multi-path routing in our network, 60% utilization is what we can achieve for this experiment. This 60% limitation is caused by ECMP hash collision, not PFC or HOL blocking. Both our simulation and the results in [2], in which no PFC was used, showed similar utilization numbers for ECMP routing in three-tier Clos networks. lish 8 RDMA connections to the corresponding server in the other ToR. All these RDMA connections needed to traverse the Leaf-Spine links. All the RDMA connections sent data as fast as possible. In total we had 3074 connections distributed among the 128 Leaf-Spine links, which were the bottlenecks in this experiment. Figure 7(b) shows the aggregate throughput measured from the servers. The unit of the y-axis is frames per second. The RDMA frame size is 1086 bytes with 1024 bytes as payload. The aggregate throughput is 3.0Tb/s. This is 60% network utilization of the total 5.12Tb/s network capacity. During the whole experiment, not a single packet was dropped. Every server was sending and receiving at 8Gb/s with the CPU utilization close to 0%. Since we use ECMP for multi-path routing in our network, 60% utilization is what we can achieve for this experiment. This 60% limitation is caused by ECMP hash collision, not PFC or HOL blocking. Both our simulation and the results in [2], in which no PFC was used, showed similar utilization numbers for ECMP routing in three-tier Clos networks. lish 8 RDMA connections to the corresponding server in the other ToR. All these RDMA connections needed to traverse the Leaf-Spine links. All the RDMA connections sent data as fast as possible. In total we had 3074 connections distributed among the 128 Leaf-Spine links, which were the bottlenecks in this experiment. Figure 7(b) shows the aggregate throughput measured from the servers. The unit of the y-axis is frames per second. The RDMA frame size is 1086 bytes with 1024 bytes as payload. The aggregate throughput is 3.0Tb/s. This is 60% network utilization of the total

5.12Tb/s network capacity. During the whole experiment, not a single packet was dropped. Every server was sending and receiving at 8Gb/s with the CPU utilization close to 0%. Since we use ECMP for multi-path routing in our network, 60% utilization is what we can achieve for this experiment. This 60% limitation is caused by ECMP hash collision, not PFC or HOL blocking. Both our simulation and the results in [2], in which no PFC was used, showed similar utilization numbers for ECMP routing in three-tier Clos networks. We unfortunately did not record the end-to-end RDMA latency in the above throughput experiment. To further investigate the relationship between network latency and throughput, we conducted the following experiment in our testbed with a two-tier network. We had two ToR switches in this testbed. Each ToR switch had 24 servers, and each ToR used 4 uplinks to connect to four Leaf switches. All the links were 40GbE. The oversubscription ratio was 6:1. We mimicked The traffic pattern in Figure 7. We chose 20 servers in every ToR and paired every server in one ToR with one server in another ToR and let every server-pair establish 8 RDMA connections. Every server achieved 7Gb/s sending/receiving throughput. We show the RDMA latency measured in Pingmesh in Figure 8. Once the experiment started, the end-to-end RDMA latencies increased from 50us at the 99th percentile and 80us at the 99.9th percentile to 400us and 800us, respectively. Note that the 99th percentile latency of TCP did not change during the experiment in Figure 8. This is because we put RDMA and TCP packets into two different queues in the switches. Hence RDMA and TCP did not interfere with each other. We note that the 99th percentile latency of TCP was 500us in Figure 8, whereas it was 700us in Figure 6. The difference was caused by the fact that the servers in Figure 6 were servicing realworld workload whereas the servers in Figure 8 were almost idle (except running the RDMA traffic generator). Figure 8 also demonstrated that the RDMA latency increase was due to the network congestion created by the RDMA traffic. The above measurement results show that, compared to TCP, RDMA achieves low latency and high throughput by bypassing the OS kernel and by eliminating packet drops. But RDMA is not a panacea for achieving both low latency and high throughput. The RDMA latency can still increase as the network becomes congested and queues build up.




(a) The network topology.



(b) The aggregate RDMA throughput.

1. 控制无关变量，保证效果：
   以下实验显示了三层 Clos 网络中数百台服务器的 RDMA 性能。我们在数据中心上线后但在将其交给客户之前使用两个 Podset 运行此实验 - 即实验期间没有客户流量。
2. 网络抽象结构：
   网络拓扑如图7(a)所示。网络中的所有端口均为 40GbE。
   一个 Podset 由 4 台 Leaf 交换机、24 台 ToR 交换机和 576 台服务器组成：【1】每个 ToR 交换机连接 24 台服务器；【2】4 个 Leaf 交换机总共连接 64 个 Spine 交换机。【3】ToR 和 Leaf 的超额认购比例分别为 6:1 和 3:2。 Podset 和 Spine 交换机层之间的聚合带宽为 64x40Gb/s = 2.56Tb/s。
3. 实验方法：
   我们使用了 ToR 到 ToR 流量模式。我们将两个 Podset 中的 ToR 一一配对。 podset 0 中的 ToR i 与 podset 1 中的 ToR i 配对。在每个 ToR 中，我们选择 8 个服务器，并让每个服务器与另一个 ToR 中的相应服务器建立 8 个 RDMA 连接。所有这些 RDMA 连接都需要遍历 Leaf-Spine 链路。
4. 实验预估瓶颈：
   所有 RDMA 连接都尽可能快地发送数据。我们总共有 3074 个连接分布在 128 个 Leaf-Spine 链路中，这是本实验的瓶颈。
5. 实验结果：
   图 7(b) 显示了从服务器测量的总吞吐量。 y 轴的单位是每秒帧数。 RDMA 帧大小为 1086 字节，其中 1024 字节作为有效负载。总吞吐量为 3.0Tb/s。这是总 5.12Tb/s 网络容量的 60% 网络利用率。
   在整个实验过程中，没有一个数据包被丢弃。每台服务器的发送和接收速度均为 8Gb/s，CPU 利用率接近 0%。由于我们在网络中使用 ECMP 进行多路径路由，因此本实验可以实现 60% 的利用率。这个 60% 的限制是由 ECMP 哈希冲突引起的，而不是 PFC 或 HOL 阻塞引起的。我们的模拟和 [2] 中的结果（其中未使用 PFC）都显示三层 Clos 网络中 ECMP 路由的利用率相似。
6. 实验反刍与另设实验：

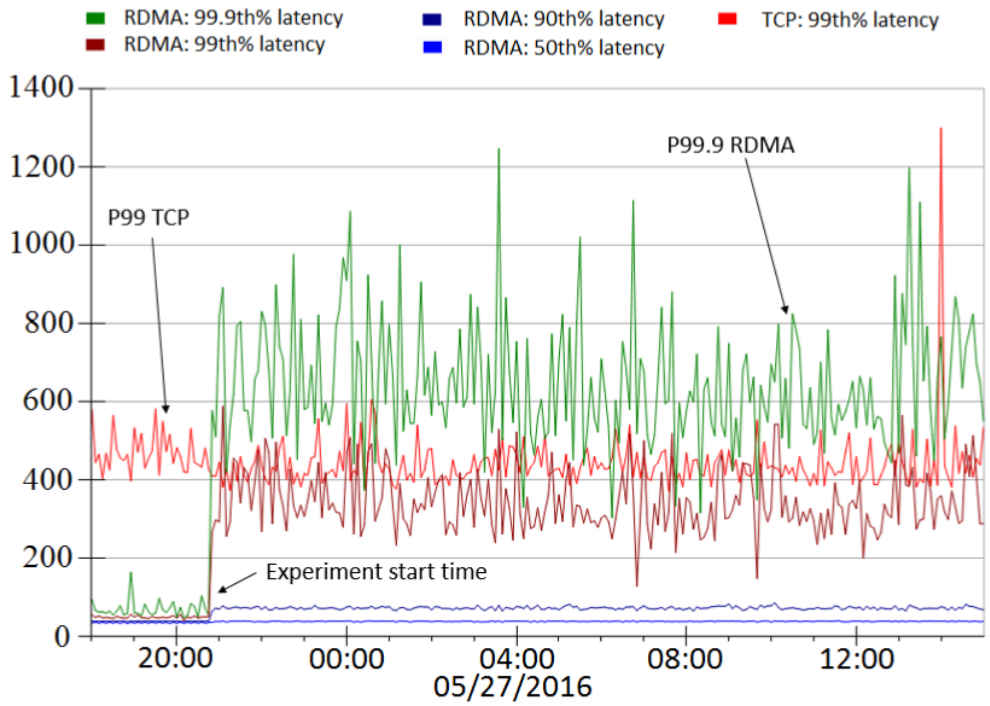



Figure 8: The end-to-end RDMA latency jumped up as the experiment started and network throughput increased.

遗憾的是，我们在上述吞吐量实验中没有记录端到端 RDMA 延迟。
为了进一步研究网络延迟和吞吐量之间的关系，我们在测试台上使用两层网络进行了以下实验。我们在这个测试台中有两个 ToR 交换机。每个 ToR 交换机有 24 台服务器，每个 ToR 使用 4 个上行链路连接到 4 个 Leaf 交换机。所有链路均为 40GbE。超额认购比例为6:1。
我们模仿图 7 中的流量模式。我们在每个 ToR 中选择 20 台服务器，并将一个 ToR 中的每台服务器与另一个 ToR 中的一台服务器配对，并让每个服务器对建立 8 个 RDMA 连接。每台服务器都实现了 7Gb/s 的发送/接收吞吐量。
我们在图 8 中显示了在 Pingmesh 中测得的 RDMA 延迟。实验开始后，端到端 RDMA 延迟分别从第 99 个百分位数的 50us 和第 99.9 个百分位数的 80us 增加到 400us 和 800us。
请注意，在图 8 的实验过程中，TCP 的第 99 个百分位数延迟没有变化。这是因为我们将 RDMA 和 TCP 数据包放入交换机中的两个不同队列中。因此，RDMA 和 TCP 不会互相干扰。
我们注意到，图 8 中 TCP 的第 99 个百分位延迟为 500us，而图 6 中为 700us。造成差异的原因是图 6 中的服务器正在为现实世界的工作负载提供服务，而图 8 中的服务器几乎处于空闲状态（除了运行 RDMA 流量生成器）。
图 8 还表明，RDMA 延迟增加是由于 RDMA 流量造成的网络拥塞造成的。
上述测量结果表明，与TCP相比，RDMA通过绕过操作系统内核并消除丢包，实现了低延迟和高吞吐量。**但 RDMA 并不是同时实现低延迟和高吞吐量的灵丹妙药。随着网络变得拥塞和队列增多，RDMA 延迟仍然会增加！**

### -ECMP技术：

ECMP（Equal-Cost Multi-Path）是一种网络路由技术，用于**在网络中选择多个等价路径中的一个来传送数据流量**。它的目标是实现负载均衡，提高网络性能和可靠性。

多路径路由是指在网络中存在多个路径可以到达相同的目标，而这些路径的成本（通常是跳数、带宽或延迟）是相等的。ECMP通过在多个等价路径之间均匀分配数据流量，以确保网络的各个路径得到合理利用。

在ECMP中，路由器或交换机会将传入的数据流量分为多个数据包，并使用某种散列算法来计算每个数据包应该选择哪个等价路径进行传输。**这个散列算法通常基于数据包的源IP地址、目标IP地址、源端口、目标端口等字段**，以确保相同的数据流量在同一路径上传输。

### -ECMP对应的哈希冲突：

def(哈希冲突): 是指*多个数据包经过散列算法计算后，得到相同的哈希值*。

当哈希冲突发生时，多个数据包将被分配到相同的路径，这可能导致某些路径负载过重，而其他路径负载较轻。这会降低ECMP的均衡性，因为它无法充分利用所有可用的路径。

为了减少哈希冲突，一些ECMP实现采用了更复杂的哈希算法，以使相同数据流量的数据包在尽可能多的情况下选择不同的路径。此外，有些ECMP实现还提供了配置选项，允许管理员自定义散列算法，以适应特定网络的需求。

总之，ECMP是一种用于实现多路径路由和负载均衡的网络技术，它通过散列算法将数据包分配到不同的路径。哈希冲突是指多个数据包被分配到相同路径的情况，可能导致不均衡的负载分布。网络管理员可以采取措施来减少哈希冲突，以提高ECMP的性能。