RDMA over Commodity Ethernet at Scale

1. 作者介绍

Chuanxiong Guo, Haitao Wu, Zhong Deng, Gaurav Soni, Jianxi Ye, Jitendra Padhye, Marina Lipshteyn

Systems Research Group—Redmond

2. 摘要

在过去一年半的时间，我们已经使用 RoCEv2 来支持一些微软高可靠性、延迟敏感的服务。本篇论文讲述了在此过程中遇到的挑战以及解决方案。为了把 RoCEv2 扩展到 VLAN 之外，我们设计了一个基于 DSCP 的优先级流量控制机制（PFC）来确保大规模部署。我们已经解决了很多安全挑战，比如 PFC-induced deadlock、RDMA transport live lock 以及 NIC PFC pause frame storm 问题。我们也建立了监控和管理系统来确保 RDMA 按照预期的方式工作运行。我们的经验表明，大规模运行 RoCEv2 的安全和可扩展问题都可以被解决，RDMA 可以代替 TCP 进行内部数据中心通信，并且可以实现低延迟、低 CPU 开销、高吞吐量。

Keywords: RDMA, RoCEv2, PFC, PFC propagation, Deadlock

3. 介绍

随着线上服务和云计算的快速发展，大规模数据中心（DC）被建立在世界各处。连接 DC 中的服务器需要高速、可扩展的数据中心网络(DCN)。DCN 的建立需要商用交换机和网卡。最先进的 DCN 必须支持 DC 中任何两个服务器之间 Gb/s 或更高的吞吐量。

在当今数据中心网络中，TCP/IP 仍然是主导的传输/网络协议栈，但是传统 TCP/IP 协议栈并不能满足新一代 DC 工作负载的需求，有两个原因：

1） OS 内核中的数据包操作带来的 CPU 开销仍然很高，尽管做了很多硬件和软件上的优化，比如卸载校验和 checksum offloading、large segment offload(LSO)、receive side scaling(RSS)和适度中断(interrupt moderation)。在我们数据中心中的测量结果显示，用 8 个 TCP 连接以 40Gb/s 发送数据会占用 6% 的总的 CPU 时间，硬件环境是 32 核 Intel Xeon E5-2690 的 Windows 2012R2 服务器。使用 8 个 TCP 连接以 40Gb/s 接收数据需要占用 12%的总的 CPU 时间，现代的数据中心是无法忍受这种高 CPU 开销的。

2） 许多当代 DC 应用，比如 Search，对网络延迟都是高度灵敏的。TCP 不能提供需要的低延迟，即使平均流量负载是适当的，有两个原因。第一，内核软件产生的延迟

有几十毫秒【21】；第二，由于网络拥塞，会有数据丢包现象出现，尽管很少，但还是存在，这时因为数据中心流量固有的突发性。TCP 通过超时或者快速重传来恢复正常，而这两种情况中，都有很大的延迟。

本篇论文总结了部署 RcCEv2 以在微软数据中心上解决以上提出的问题的经验，RDMA 是一种在不中断 CPU 操作的前提下可以访问远程系统的内存的方法，RDMA 广泛应用于以 Infiniband 为架构的高性能计算中， RoCEv2 支持 RDMA 在以太网上的实现，而不是 Infiniband。

和 TCP 不同，RDMA 需要一个无损网络。比如说，交换机中的缓冲区溢出(buffer overflow)不能引发数据包丢失，RoCEv2 使用基于优先级的流量控制 PFC 来实现无损网络。当缓冲区占用率超过指定阈值时，PFC 通过暂停上游发送实体来防止缓冲区溢出。 尽管 PFC 存在一些众所周知的问题，如 head-of-the line blocking 和 deadlock 的可能性，但我们的部署过程中发现了几个尚未被提出的新问题，例如 RDMA transport live lock，NIC PFC pause frame storm, the slow receiver symptom。 即使是我们遇到的死锁问题的根本原因也是和之前研究文献中经常讨论的简单例子完全不同。

VLAN 标签典型地被用来在混合 RDMA/TCP 部署中鉴别支持 PFC 的流量。这种解决方案不会出现在我们的环境中。因此。我们提出了基于 PFC 的 DSCP（Differentiated Services Code Point）概念，把 RDMA 从二层的 VLAN 扩展到三层的 IP。

文章对应的 RDMA 部署已经顺利运行了一年半，支持了微软的一些高可靠、延迟敏感的线上服务。经验表明，通过改进 RoCEv2 的设计、解决不同的安全问题、建立需要的管理和监控功能，可以使用商用以太网(commodity Ethernet)，把 RDMA 安全地部署在大规模数据中心中。

4. 背景

我们的数据中心网络是一个基于以太网的多层 Clos 网络。20-40 个服务器连接到一个 top-of-rack（ToR，架顶交换机）交换机，数十个 ToR 连接到 Leaf 交换机层。Leaf 交换机反过来连接到数十到上百个 Spine（脊）交换机的层上。大多数链路是 40Gb/s，我们计划在不久的将来更新到 50GbE 和 100GbE，所有的交换机都是使用 IP 路由。

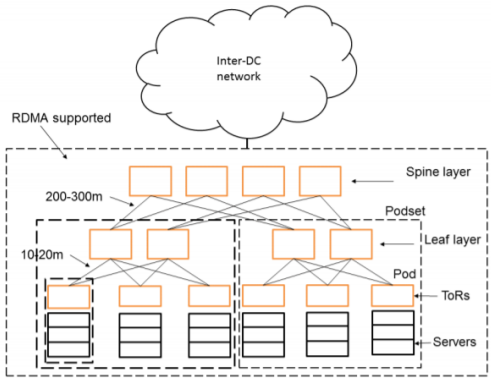


Figure 1: Our goal is to support RDMA for intra data center (intra-DC) communications.

服务器和使用大约 2 米的铜电缆连接到 ToR 交换机，ToR 交换机和叶子交换机之间有 10-20 米的距离，leaf 和 spine 交换机之间有 200-300 米。三层交换机将数以万计的服务器连接到一个数据中心，本篇论文中，我们的关注点是同一个 spine 交换机层下的若干服务器之

间的支持 RDMA。

RoCEv2：部署 RoCEv2 是基于技术和经济的两个原因。RoCEv2 在一个 Ethernet/IPv4/UDP 数据包中封装一个 RDMA 传输包，使得 RoCEv2 和我们现有的网络基础设施相兼容。基于 ECMP 的多路径路由需要 UDP 头部，目的 UDP 端口通常设置为 4791，源 UDP 端口对每个 QP(queue pair)是随机选择的。中间交换机使用标准的五元组哈希。因此，属于同一个 QP 的流量有相同的路径，而不同 QP 中的流量可以有不同的路径（甚至在同一对通信终端之间）。

PFC 和 buffer 预留：RoCEv2 使用 PFC 来防止缓冲区溢出。PFC 标准指定了 8 个优先级种类，来减少 head-of-line 阻塞问题，事实证明还是存在。但是，在我们的网络中，RDMA 只能使用 8 个中的两个优先级。原因如下：

1） PFC 是一个在两个以太网节点之间的逐跳协议。发送者的出端口把数据发送到接收者的入端口，出端口把数据包放到最多 8 个队列中进行排队，每个队列对应一个优先级，入端口把数据包缓存到对应的接收队列中。在我们网络中使用的是共享缓冲区的交换机，一个接收队列被作为一个 counter 简单地实现，所有的数据包共享一个通用的 buffer 池。

一旦接收队列的长度达到了一定阈值（XOFF），交换机会发送 PFC 暂停帧到对应的上流的发送队列。在发送队列接收到暂停帧的时候，就会停止发送数据包。暂停帧中包含了需要暂停的优先级队列和暂停时间。一旦接收队列的长度小于另一个阈值（XON），交换机会发送一个 0 持续时间的暂停帧给发送队列来恢复传输。XOFF 的值必须能够保证没有 buffer overflow，XON 来保证没有缓冲区 buffer underflow。（overflow 表示缓冲区已满，不能再写入了，underflow 表示缓冲区空的，不能读取数据）。

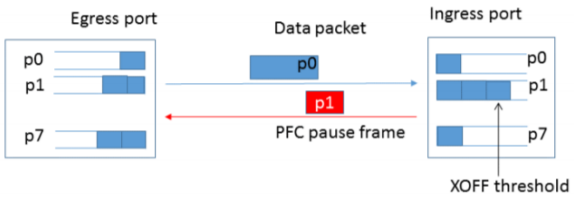


Figure 2: How PFC works.

暂停帧到达上游出端口会消耗一定时间，交换机反应也需要一定时间，所以在这段时间内，上游流量端口会继续发送数据包，所以接收端口必须给每个优先级预留出缓冲区空间来存储“gray period”（灰色时期）收到的数据包，这个预留的 buffer 叫做 *headroom*，其大小取决于 MTU、出端口的 PFC 反应时间，以及最重要的，发送者和接收者之间的传播时延。

传播时延取决于发送者和接收者之间的距离。在我们的网络环境中，二者的距离最大是 300 米。ToR 和 leaf 交换机有 shallow buffers（9MB 或者 12MB），即使交换机支持 8 个流量类别，我们只能给两个无损的流量类别预留充足的 *headroom*。一个无损类别进行实时流量传输，另一个无损类别进行批量大数据传输。

需要拥塞控制：PFC 是逐跳工作的，源和目的服务器之间可能有多跳，如果有持续的网络拥塞，PFC 暂停帧会从阻塞点传播并返回到源，这就会导致诸如 unfairness 和 victim flow 的问题【42】。

为了减少这些额外的问题，包括 QCN、DCQCN 和 TIMELY 在内的基于流量的拥塞控制机制应运而生。我们选择使用 DCQCN，本质是利用 ECN 进行拥塞警告，之所以选择 DCQCN，是因为它能直接对中间交换机的队列长度进行响应，并且所有的交换机都支持 ECN。小的队列长度减少了 PFC 的产生和传播几率。尽管 DCQCN 帮助减少了 PFC 暂停帧的数量，但是是 PFC 在保护数据包不被丢弃。PFC 提出了一些安全问题，正的本论文的重点内容，第

4 部分会进行讨论。我们相信，从本论文学到的经验教训同样适用于使用 TIMELY 的网络。

RDMA 和 TCP 共存：本论文中，RDMA 是为 intra-DC 通信设计的，而 inter-DC(DC 之间)和延迟应用仍然需要 TCP。TCP 使用一个不同的流量类（不是无损），不同的流量类别将 TCP 和 RDMA 的流量隔离开来。

5. DSCP-BASED PFC

在本小节中，我们测试了原始的基于 VLAN 的 PFC 面对的问题，并提出了基于 DSCP 的 PFC 方案。基于 VLAN 的 PFC 暂停帧中，VLAN TAG 中包含了数据包优先级和 VLAN ID，但是优先级和 VLAN ID 引发了两个严重的问题，因此提出了基于 DSCP 的 PFC 方案。

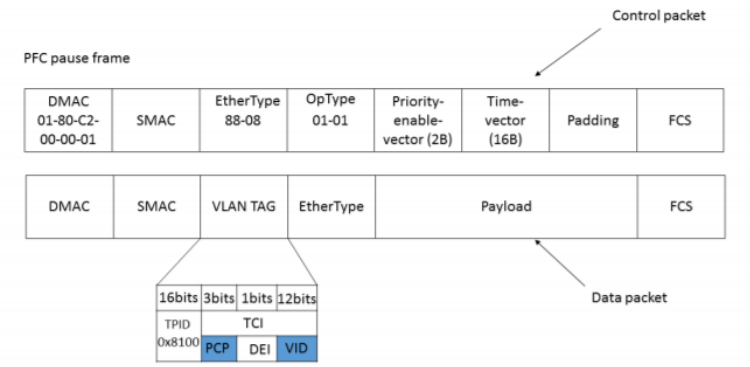
传统的基于 VLAN 的 PFC 的 PAUSE 帧和数据包格式：暂停帧是一个二层帧，并没有 VLAN 标签，数据包的 VLAN 标签有四部分，TPID 被固定为 0x8100，DEI（Drop Eligible Indicator 丢弃符合条件的指标），PCP（Priority Code Point）包含数据包的优先级，VID（VLAN identifier）是数据包的 VLAN ID。

尽管我们只需要 PCP，但是 PCP 和 VID 是不可分离的，因此，为了支持 PFC， 我们必须在服务器端和交换机端都配置 VLAN。为了使得交换机端口支持 VLAN，我们需要把面向交换机端口的服务器设置为 trunk 模式（支持 VLAN 标记的数据包），而不是 access 模式（发送和接收的都是没有标记的数据包）。基本的 PFC 功能都是使用这种配置，但是会引发两个问题。

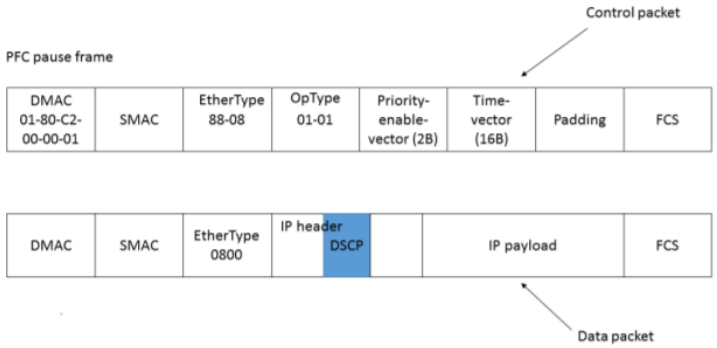
1） 交换机的 trunk 模式和操作系统提供的服务有不利的交互。OS provisioning 是一个基本服务，当服务器 OS 需要更新或者安装，或者当服务器需要被修复和供应的时候，需要运行这个基本服务。在我们的数据中心中，OS provisioning 必须自动完成。我们使用 PXE boot 来安装 OS。当服务器经过 PXE boot 的时候，它的 NIC 没有 VLAN 配置，结果就是不能发送和接收带有 VLAN 标签的数据包，但是由于面向服务器端口配置成了 trunk 模式，这些交换机端口只能发送带有 VLAN 标签的数据包，因此服务器之间的 PXE boot 通信和 OS provisioning 服务就崩溃了。我们尝试了一些“黑客”方式进行问题修复，比如基于猜测的服务器状态改变交换机端口配置，让 NIC 接收所有的数据包，不管是否有 VLAN 标签，但是这些方式都是很复杂并且不可靠的，或者说，都不是标准的。

2） 我们已经从二层 VLAN 移开了，我们所有的交换机，包括 ToR，都运行着三层 IP 交付，而不是基于 MAC 的二层桥接。一个三层网络有很多优势，比如扩展性、更好的管理和监控、更好的安全性、所有的协议都是公开而标准的。但是，在三层网络中，当穿越子网边界时，没有标准的方式实现 VLAN PCP 跨 L3 网络传输(VLAN 是一个 L2 协议)。

分析这两个问题，可以发现产生的原因都是，基于 VLAN 的 PFC 不必要的数据包优先级和 VLANID 对，因此提出了基于 DSCP 的 PFC，替换掉 PCP 和 VID。我们的关键观点就是 PFC 暂停帧并不包含 VLAN 标签，数据包中的 VLAN 标签只是用来携带数据包优先级，但是在 IP 中，我们有更好的方式传输优先级信息，那就是 IP 头部的 DSCP 域。



(a) VLAN-based PFC.



(b) DSCP-based PFC.

解决办法就是把数据包优先级从 VLAN 标签中移到 DSCP，这个改变是很小的，并且只涉及到了数据包（data packet）的格式，PFC 暂停帧并没有改变。基于 DSCP 的 PFC 中，没有 VLAN 标签，所以就没有上述两个问题，因为上述两个问题的起因就是因为 VLAN 标签中的优先级和 VLAN ID。面向服务器端口不用配置成 trunk 模式了，也就意味着 PXE boot 不会有任何问题。与此同时，数据包的优先级会以 DSCP 值的形式，在子网中通过 IP 路由正确传输。当然，基于 DSCP 的 PFC 并不能为工作在二层的设计提供服务，但是对我们来说也没任何问题，因为数据中心中没有二层网络。(比如说 FCoE 以太网光纤信道)

DSCP-based PFC 要求 NIC 和 Switch 能够正确分类不同 DSCP 值的包并把他们分队列。在每个 NIC 和 Switch 的端口都维持了 8 个 PG（Priority Group），这 8 个 PG 可以标记为无损或者有损。当某 PG i(i=0,1,,,,,7)被标记为无损时，一旦它的接收缓冲到达了 XOFF，对应的 i 优先级的 PAUSE 帧就会被发送到上一跳。DSCP 值和 PFC 优先级的映射关系是灵活的，甚至可以多对一，本论文中，简单地将 DSCP 值 i 映射成了 PFC 优先级 i。

基于 DSCP 的 PFC 是公开可用的，并且所有的供应商都支持，我们坚信基于 DSCP 的 PFC 提供了一种比原始基于 VLAN 的策略更简单、扩展性更好的解决方案。

6. 安全性挑战

6.1 RDMA-Live lock

RDMA 传输协议的设计有一个假设前提，就是在网络拥塞的时候，数据包不会被丢弃，在 RoCEv2 中是通过 PFC 来实现的，但是，丢包现象仍然会发生，比如 FCS 错误（帧校验码错误）或者交换机软件、硬件层面的 bug。理想情况下，我们希望 RDMA 性能在存在此

类错误的情况下能够尽可能地不会损失太多。但是我们发现，即使丢包率很低，RDMA 性能也会大幅下降。我们用下面简单的实验来说明这个问题。

用一台交换机 W 连接两个服务器 A 和 B，分别进行 RDMA SEND，WRITE，READ 实验。第一个实验中，A 执行 RDMA SENDs 以最快速度将每块 4M 的数据块发送到 B；第二个实验中，A 执行 RDMA WRITE 操作，其余和第一个实验相同；第三个实验中，B 使用 RDMA READ 以最快的速度从 B 读取 4MB 数据块。实验环境中，丢包率是 1/256（0.4%）。（IP ID 字段低 8 位累加，对于 0xff 的数据包 丢弃）

我们发现，即使丢包率很低，应用端的实际吞吐能力还是为 0。换句话说，系统处于活锁状态(live lock)，虽然链路被充分利用，但应用程序没有任何进展。根源在于 go-back-0 算法，这个算法用来进行 RDMA 传输的丢失恢复。假设 A 给 B 发送消息，这个消息被分段成了数据包 0，1，…，m，假设数据包 i 丢失了，那么 B 会给 A 发送 NAK（i），A 收到之后就会从数据包 0 重新发送该消息，因此 go-back-0 就导致了活锁。一个 4MB 的消息被分成 4000 个数据包，因为丢包率是 1/256，假设第一个 256 个数据包会丢失一个数据包，那么发送者就会重新发送，发送的时候又会丢包，所以会一直发送，但没有任何进展。表面上看起来一直在发送，但实际上并没有有效进展。

TCP 和 RDMA 在网络上有不同的假设。TCP 是最大努力交付，允许数据包丢失，并且通过诸如 SACK（）等复杂的重传机制解决丢包问题。RDMA 假设网络是无损的，因此供应商选择使用简单的 go-back-0 方法，在这种方法中，发送者不需要维护重传状态。

这个实验清晰地展示了在大规模网络环境中（比如我们的数据中心网络），尽管使用了 PFC，但是仍然会有是丢包现象发生，因此还是需要一个复杂的重传机制。因为 RDMA 的传输是在网卡上进行的，NIC 的资源限制意味着无法实现 SACK(Selective Acknowledgement，选择性确认)那样复杂的重传机制，同时，SACK 也没有必要，因为 PFC 已经几乎消除了网络拥塞造成的丢包现象，丢包现象很少，没必要使用如此复杂的机制。

我们的解决办法是用 go-back-N 代替 go-back-0 机制，在 go-back-N 中，重传从第一个丢失的数据包开始，之前已经接收到的数据包不需要重传。在最坏的情况下，因为一个丢包导致的带宽资源浪费最多是 RTT*C(C 是带宽)。Go-back-N 机制几乎和 go-back-0 一样简单，同时避免了活锁。自从使用了 go-back-N，我们没有遇到过活锁，因此建议使用 go-back-N。

6.2 RDMA-Deadlock

ToR 交换机连接的多个 servers 共同构成一个子网。ToR 交换机分发数据包要查询两个表。ARP 表和 MAC 地址表，ARP 表根据 IP 查找对应的 MAC 地址，MAC 地址表根据 MAC 地址查找对应的端口。两个表都使用超时机制来清除过期的记录。默认的 ARP 的超时值是 4 小时，MAC 表的超时值是 5 分钟。

不同的更新时间会造成”incomplete 记录的问题”,也就是 ARP 中有关于一个 MAC 地址的记录但是 MAC 地址表中已经被删除了(比如说 server 宕机)，当有一个 packet 的目的 MAC 是这个地址的时候，就会找不到对应的转发端口。这个时候交换机的做法是，把这个 packet 转发到所有端口上，也就是 flood。

下面是 DeadLock 的一个场景：

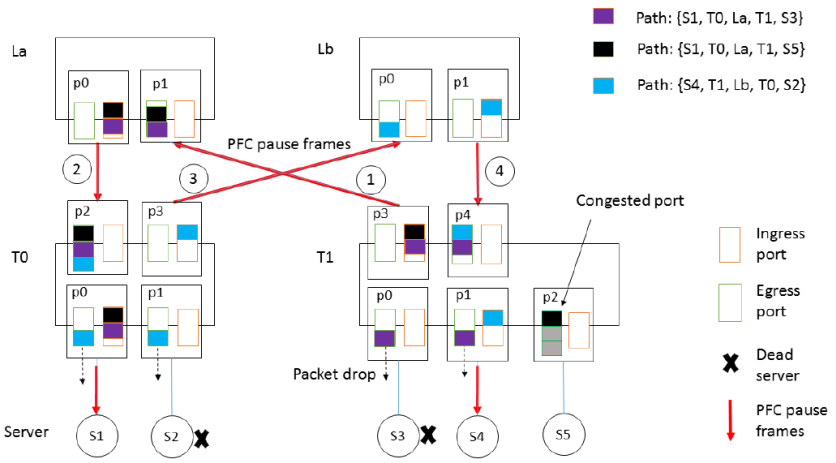


Figure 4: An example to show that the interaction between Ethernet packet flooding and PFC pause frame propagation can cause deadlock.

1. S1 发送数据包给 S3,S5，通过路径{T0,La,T1}。紫色数据包->S3 黑色数据包->S5 现在 S3 宕机，所以紫色数据包会被 flood 到所有端口。在紫色数据包还没有到达 T1.p4 的出队列队头之前，紫色数据包会一直加入队列。T1.p2 端口也会阻塞。所以黑色的数据包会在 T1 中排队，最后造成阻塞。导致 T1.p3 会向 La.p1 发 PAUSE 帧，PAUSE 帧一直反向传播到 S1.
2. 这时候如果 S4 开始按照{T1,Lb,T0}向 S2 发数据。S2 这时候也宕机，导致发送到 S2 的数据包会 flood 到其他端口。因为 T0.p1 的数据包因为 PAUSE 的影响没有办法发送数据包，所以最后 T0.p3 会向 Lb.p0 发 PAUSE。
3. 最终的结果就是，Lb.p1 的入端口开始阻塞 T1.p4 而 T1.p1 的入端口开始阻塞 S4

在四个交换机中会形成一个 pause 环，出现 dead lock。一旦 deadlock 所有流都阻塞，即使重启 sever 也不会解决。

有几个可选择的方法来解决 deadlock:（1）包的转发交给交换机的 CPU 决定（2）将 MAC 表的超时时间设置的比 ARP 表要长，这样就不会出现 MAC 查不到的情况（3）对于标记为 lossless 的数据包，当查不到它的 MAC 时，就丢弃，而不是转发。

我们选择第三个解决方案：

我们从 PFC 死锁中吸取的教训是广播和多播对于无损网络是危险的。为防止发生死锁，我们建议广播和多播数据包不应放入无损类。

6.3 NIC PFC pause frame storm

PFC 帧原本设计用来防止 lossless 包丢包，但是会引起 head-of-the line blocking。最坏的情景下：

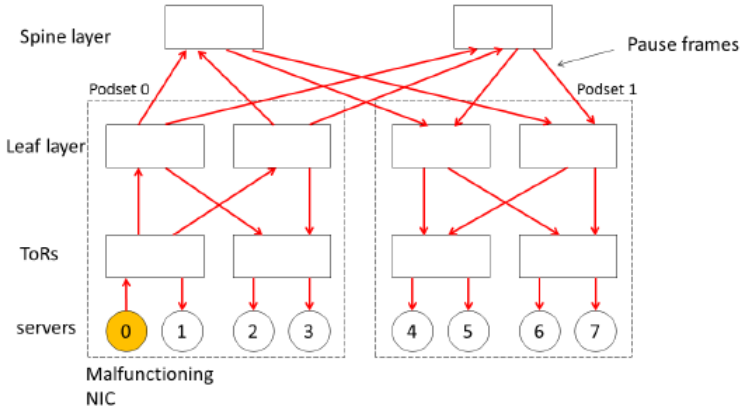


Figure 5: The PFC pause frame storm caused by the malfunctioning NIC of one single server (server 0).

1. S0 网卡故障，导致发 PAUSE 给 TOR0,然后反向传播给 Leaf Spine Leaf ToR Servers
2. 最终会导致所有的 server、交换机都会被 PAUSE，发生了 PFC 帧风暴

解决方案，在 NIC 和 ToR 上都设置两个看门狗：

NIC 上的看门狗：NIC 上有一个独立的微控制器可以用来监测 NIC 接收端的流水线 pipline 工作情况，如果检测到流水线停止工作超多 100ms（NIC 故障）就会禁止这个网卡，使得他不会再往上流发送 PAUSE 帧，阻断其对整个网络的影响。

交换机上的看门狗：监控面向 server 的 egress port 的队列不断增长，数据没有被发送，与此同时接收到了 PAUSE 帧。如果出现这种情况，就会关闭改端口对应的 lossless 模式，丢弃所有的从 NIC 过来的无损数据包。一旦发现 PAUSE 帧消失，200ms 以后会再次恢复 ToR 中的 lossless 设置。

6.4 The Slow-receiver symptom

关于 6.3 的 PFC 风暴问题，一开始没有定位 NIC 发送 PAUSE 的根本原因。因为对于 RDMA 是 NIC 的数据通过总线直接发送到内存。微软的数据中心中的网卡是 40Gb/s，而 PCI 总线的带宽是 64Gb/s 理论上网卡到内存不会发生阻塞

原因：NIC 的存储空间有限，所以对于 RDMA 中的大多数 QPC(Queue Pair Context)和 WQE（Work Queue Element）是存储在内存中的，只是缓存了一部分在 NIC 的存储中。NIC 具有一个将虚拟内存转换为物理内存的内存转换表（MTT）。MTT 只有 2K 条目。对于 4KB 的页面大小，2K MTT 条目只能处理 8MB 存储空间。如果 WQE 中的虚拟地址未在 MTT 中映射，则会导致缓存未命中，NIC 必须替换新虚拟地址的一些旧条目。 NIC 必须访问服务器的主内存以获取新虚拟地址的条目。这时候接受的 pipline 就会一直等待，导致 buffer 的数据超过阈值，引发 PAUSE。

解决方案：

交换机上各端口的缓冲共享（动态的），而不是静态的分配，这样就给了一些端口更多的缓冲，当 NIC 发送 PAUSE 时，端口也有更大的容量缓存，防止影响到剩下的网络。

NIC 上内存页设置为 2MB 而不是 4KB。这样 MTT 对应的存储空间就会大一点，减少未命中的概率。

7. 生产环境中部署 RDMA

RDMA 环境的管理和监控功能。包括：前一节提到的各种安全问题，各种错误事件的监控，对无损数据流的、PAUSE 帧的管理监控。

8. 实验验证

降低延时：

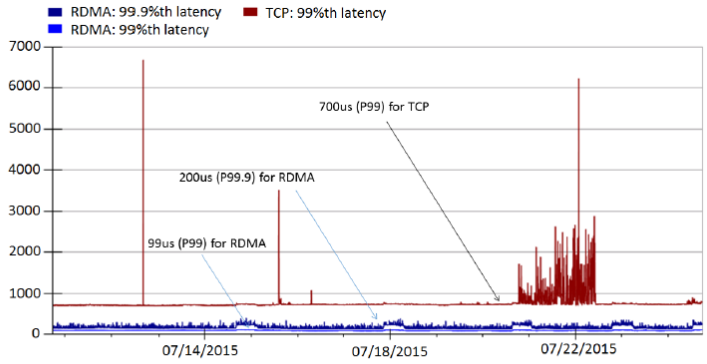
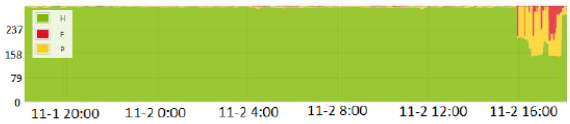


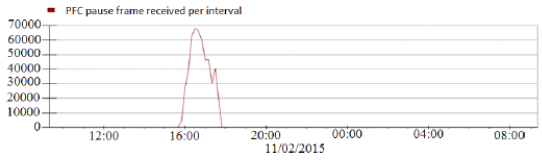
Figure 6: The comparison of the measured TCP and RDMA latencies for a latency-sensitive service.

8.1 事件

PAUSE frame storm



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



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

Figure 9: An incident caused by the NIC PFC storm problem of a single server.

当在每个 server 和 switch 上都部署了看门狗之后，PAUSE 帧风暴就再也没有出现过。

Switch 各端口的 buffer 动态分配：

由∝控制，只要满足∝× UB > $B_{p,i}$ UB 是还未分配的 buffer 空间，Bp,i 是端口 p 的 i 优先级流的分配空间。∝大的话就是分配的多，有利于减少 pause frame storm。但是可能会造成

不公平。

经过试验∝被设置为$\frac{1}{16}$

9. 总结

在本文中，我们介绍了我们在 Microsoft 数据中心大规模部署 RoCEv2 的实践和经验。我们的做法包括引入基于 DSCP-based PFC，将 RoCEv2 从第 2 层 VLAN 扩展到第 3 层 IP，并逐步实施部署流程。 我们的经验包括 RDMA live lock, dead lock, PAUSE 风暴，接受者延迟现象的发现与解决。 随着 RDMA 管理和监控的到位，我们的一些高度可靠的延迟敏感服务已运行 RDMA 超过一年半。

9.1 Future work

接下来的步骤有几个方向。 PFC 的逐跳距离限制在 300 米以内。 因此，RoCEv2 仅适用于同一个 Spine 交换机层下的服务器。 为此，RoCEv2 像 TCP 那样通用。 我们需要解决关于如何扩展 RDMA 以进行 inter-DC 通信的想法。

测量显示 ECMP 只能达到 60％的网络利用率。 对于 TCP，有 MPTCP 【29】和 per-packet routing 【10】以提高网络利用率。 如何使这些设计在无损网络环境下适用于 RDMA 将是一个有趣的挑战。

在本文中发现的 dead lock，数据中心的 dead lock 可能值得进行更系统的研究。 尽管 Clos 网络中的上下路由可以防止死锁，但像 F10【23】这样的设计可能会通过引入本地重路由来打破假设。许多其他网络拓扑【20】甚至没有上下路由属性。 在这些设计中如何避免死锁？

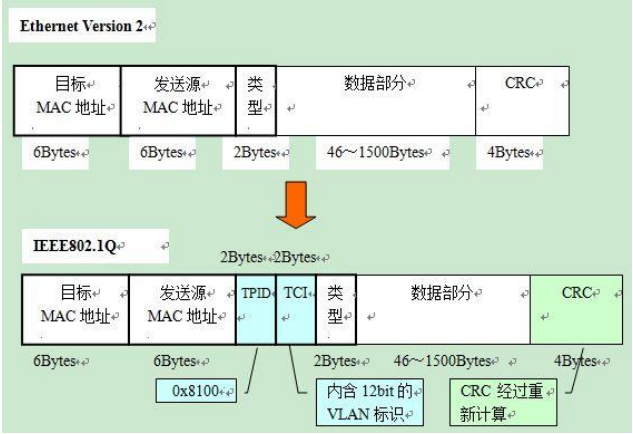
已经证明，通过消除操作系统内核数据包处理开销以及依靠无损网络，RDMA 可提供低延迟和高吞吐量。 然而，无损网络并不能保证低延迟。 当网络拥塞发生时，队列建立并且 PFC 暂停帧可能仍然生成。 队列和 PFC 暂停帧都会增加网络延迟。 如何在 RDMA 的同时实现低网络延迟和高网络吞吐量仍然是一个悬而未决的问题。

10.相关概念解释

10.1 VLAN

虚拟局域网，是一组逻辑上的设备和用户，这些设备和用户并不受物理位置的限制，可以根据功能、部门及应用等因素将它们组织起来，相互之间的通信就好像它们在同一个网段中一样，一个 VLAN 就是一个广播域。虚拟 LAN（局域网）是一个逻辑子网络，它可以将来自不同物理局域网的设备集合在一起。

VLAN 的实现是 IEEE 802.1Q，与传统以太网帧的区别：



该字段的前 16 位包含硬编码的数字 0x8100，它触发以太网设备识别框架属于 802.1 Q VLAN。这个字段的最后 12 位包含 VLAN ID，一个数字在 1 到 4094 之间。

10.2 large segment offload

利用网卡分割大数据包, 并分包发出，减小 CPU 负荷的一种技术

10.3 checksum offloading

网卡负责计算需要发送或者接收到的 TCP 消息的校验和，从而节省 CPU 的计算开销

10.4 receive side scaling

是一项网卡的新特性，俗称多队列。具备多个 RSS 队列的网卡，可以将不同的网络流分成不同的队列，再分别将这些队列分配到多个 CPU 核心上进行处理，从而将负荷分散，充分利用多核处理器的能力。

10.5 interrupt moderation

中断调控的意思，如果是节制模式会使网络硬件在收到新数据包时不会立即中断进行处理，而是继续收包，直至超时(超出预设时间)再中断，即能有效减少中断次数，发挥出硬件更佳的性能。

10.6 Clos 网络

多级交换架构
Leaf-Spine: 叶脊拓扑网络

10.7 RoCEv2:(RDMA)

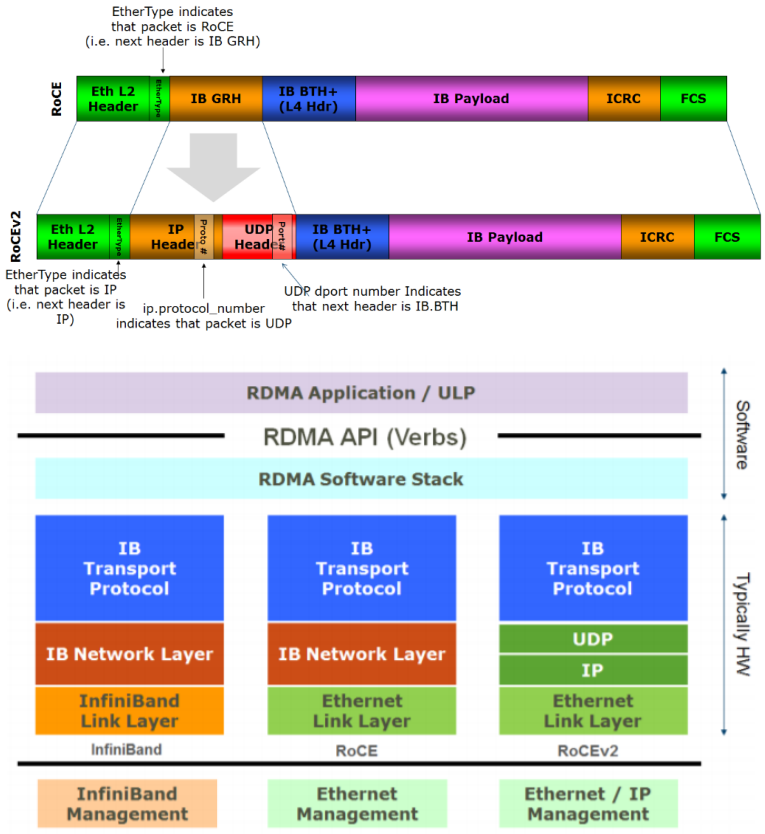
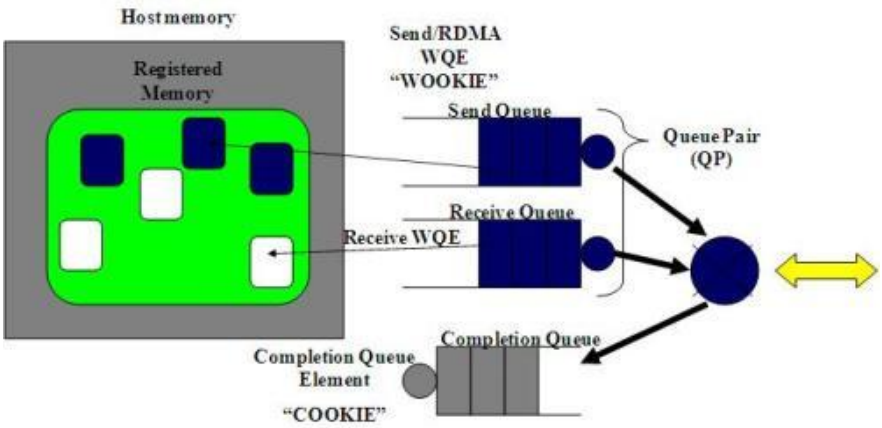


Figure 4 RoCEv2 Protocol Stack

RoCE 的传输过程：



Channel-IO/QP: 消息服务建立在通信双方 本端和远端应用之间创建的 Channel-IO 连接之上 。当应用需要通信时，就会创建一条 Channel io 连接，每条 Channel 的首尾端点是两对 Queue Pairs(QP)，每对 QP 由 Send Queue(SQ)和 Receive Queue(RQ)构成 ，这些队列中管理着各种类型的消息。

Work Queue (WQ): RDMA 提供的一套软件传输接口，方便用户创建传输请求 Work Request(WR) ，WR 中描述了应用希望传输到 Channel 对端的消息内容，WR 通知 QP 中的某个队列 Work Queue(WQ)。在 WQ 中，用户的 WR 被转化为 Work Queue Element (WQE)的格式，等待 RNIC 的异步调度解析，并从 WQE 指向的 Buffer 中拿到真正的消息发送到

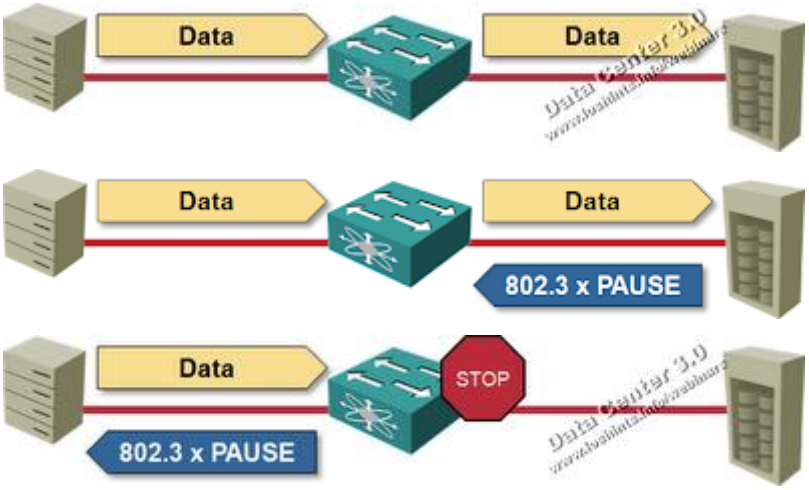
Channel 对端.

10.8 ECMP(Equal-cost multi-path routing)

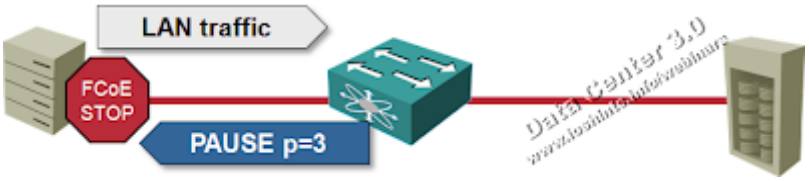
路由算法，即存在多条到达同一个目的地址的相同开销的路径。当设备支持等价路由时，发往该目的 IP 或者目的网段的三层转发流量就可以通过不同的路径分担，实现网络的负载均衡，并在其中某些路径出现故障时，由其它路径代替完成转发处理，实现路由冗余备份功能。
传统的路由技术，发往该目的地址的数据包只能利用其中的一条链路，其它链路处于备份状态或无效状态，并且在动态路由环境下相互的切换需要一定的时间
缺点：网络中各条路径的带宽、时延和可靠性等不一样，把 Cost 认可成一样，不能很好地利用带宽

10.9 PFC

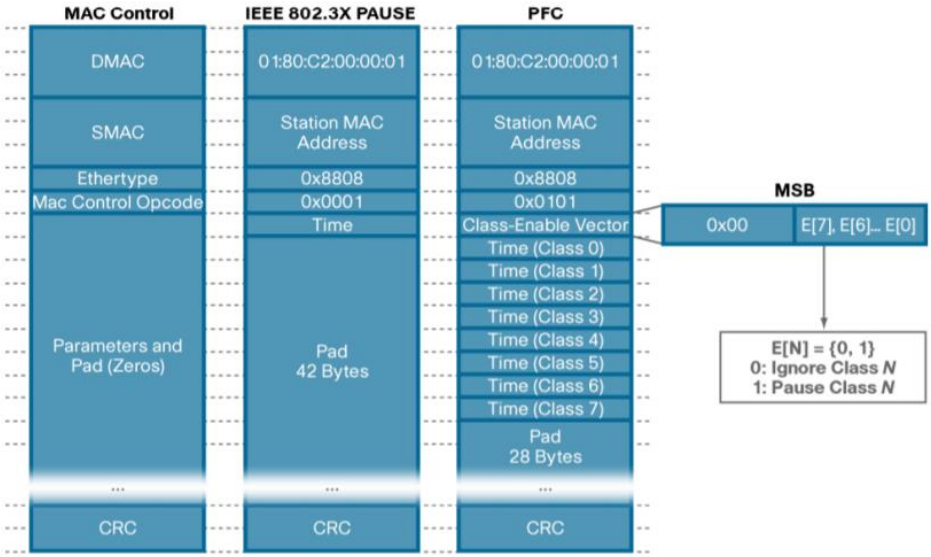
为了实现无损以太网的一些扩展
基于优先级的流量控制（PFC），IEEE 标准 802.1 Qbb，是一个链路级流控制机制。流控制机制类似于 IEEE 802.3 x 以太网暂停，但它是根据优先级进行操作的。PFC 允许你根据它的类有选择性地暂停流量，而不是在一个链接上暂停所有的流量。
IEEE802.3x：暂停机制是以太网（802.3）标准的一部分，允许在点对点的以太网链路上的接收器停止相邻的发送方，从而防止缓冲区溢出和数据包丢失。



一个链路被暂停以后，不能发送任何数据包。应用 IEEE 802.3 x 暂停使以太网段不适合承载多个流，这些流需要不同的 QoS 等级
802.1Qbb：



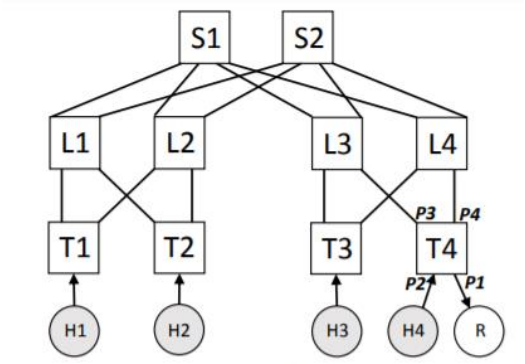
802.3x，802.1Qbb 都是使用 64 字节的 PAUSE 帧来实现的



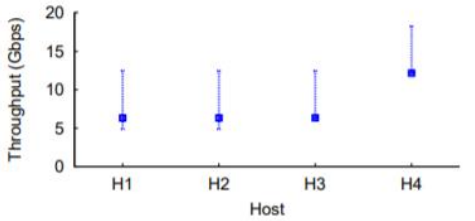
一共有 8 个优先级，以及 8 个类别对应的暂停持续时间
暂停机制在每个端口（和优先级）的基础上运行 - 而不是基于每个流。!!!

10.10 PFC-Unfairness

(sigcomm15 Congestion Control for Large-Scale RDMA Deployments)



(a) Topology

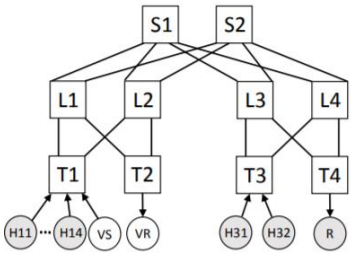


(b) Throughput of individual senders

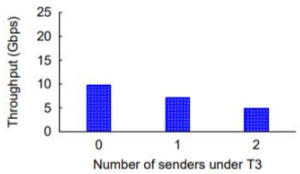
Figure 3: PFC Unfairness

H1,H2,H3,H4 同时发送数据给 R(使用相同的优先级)，如果 T4->R 阻塞，就会向 P2,3,4 发 PAUSE 帧，因为 ECMP 的原因，造成了 H4 的带宽更大，因为 H1,2,3 共享了部分的链路

10.11 Victim Flow



(a) Topology



b) Median throughput of victim flow

Figure 4: Victim flow problem

H11，…H14 发送数据给 R，VS 发送数据给 VR。因为暂停帧可以有级联效果，一个流

可能会受到甚至不发生在自己的路径上的拥塞的影响。因为 PAUSE 的级联效果，当 H1,..H14 到 R 的流在路径上发生拥塞，那么可能会影响到 VS 到 VR 的流，及时 VS 到 VR 的流的路径上并没有发生拥塞。

10.12 head-of-the line blocking

暂停机制在每个端口（和优先级）的基础上运行 - 而不是基于每个流。所以还是会有 head-of-the line 问题。

H11，…H14 发送数据给 R，如果瓶颈 T4 发生 overflow 那么 PAUSE 会一直反向传播到 T1，T1 上的上行数据不管目的地是否是 R 都会被影响。

10.13 DSCP(Differentiated Services Code Point)

它由 IP 分组报头中的 6 位组成，使用的是 ToS 字节，因此在使用 DSCP 后，该字节也被称为 DSCP 字节。其在字节中的位置如下：

DS5 DS4 DS3 DS2 DS1 DS0 CU CU

DSCP 优先级值有 64 个（0-63），0 优先级最低，63 优先级最高.

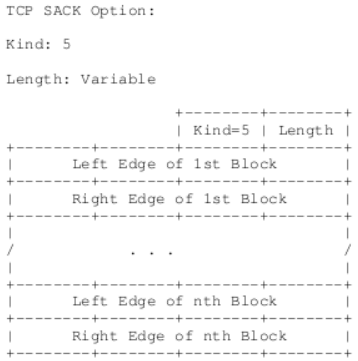
剩下的两位一般做 ECN

10.14 live lock

系统处于活锁状态(live lock)，虽然链路被充分利用，但应用程序没有任何进展。根源在于 go-back-0 算法，这个算法用来进行 RDMA 传输的丢失恢复。假设 A 给 B 发送消息，这个消息被分段成了数据包 0，1，…，m，假设数据包 i 丢失了，那么 B 会给 A 发送 NAK（i），A 收到之后就会从数据包 0 重新发送该消息，因此 go-back-0 就导致了活锁。一个 4MB 的消息被分成 4000 个数据包，因为丢包率是 1/256，假设第一个 256 个数据包会丢失一个数据包，那么发送者就会重新发送，发送的时候又会丢包，所以会一直发送，但没有任何进展。表面上看起来一直在发送，但实际上并没有有效进展。

10.15 SACK

SACK 选项可以告知发包方收到了哪些数据，发包方收到这些信息后就会知道哪些数据丢失，然后立即重传丢失的部分。



该选项参数告诉对方已经接收到并缓存的不连续的数据块，注意都是已经接收的，发送方可根据此信息检查究竟是哪个块丢失，从而发送相应的数据块。

Left Edge of Block:不连续块的第一个数据的序列号

Right Edge of Block:不连续块的最后一个数据的序列号之后的序列号

10.16 看门狗

一种监控软件，看门狗就是定期的查看芯片内部的情况，一旦发生错误就向芯片发出重启信号的电路。看门狗命令在程序的中断中拥有最高的优先级。防止程序跑飞。也可以防止程序在线运行时候出现死循环。

11．Refs

- **QoS in RoCE**
- IP 头部