



IP网络系列丛书

# 智能无损网络（HPC场景）

主编：陈乐

数据通信数字化信息和内容体验部 出品




# 版权声明

---

主编：陈乐  
主要参与人员：张帆、祝春荣、虞玲玲、姚成霞  
发布日期：2021-12-10  
发布版本：02

版权所有©华为技术有限公司 2021。保留一切权利。  
非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明

 和其他华为商标均为华为技术有限公司的商标。  
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 前言

---

## 主编简介

陈乐，华为数据中心网络资料工程师，2011 年加入华为，从事数据通信产品文档开发工作，曾担任《云数据中心网络架构与技术》一书的主编。

## 本书内容

本书重点介绍了华为智能无损网络解决方案在 HPC 高性能计算场景里产生的背景、带来的价值、使用的关键技术等内容。

## 读者对象

本书适合企业的中高层管理人员和网络工程师，对 HPC 高性能计算网络有所了解的读者阅读。



# 目录

第 1 章 HPC 高性能计算业务对网络提出更高的诉求 .....	1
第 2 章 HPC 高性能计算网络演进 .....	4
2.1 从 TCP 到 RDMA.....	4
2.2 从 IB 到 RoCE.....	5
第 3 章 基于以太网的智能无损网络技术.....	9
3.1 智能无损网络技术概览.....	9
3.2 智能无损网络技术架构.....	11
3.3 基于端口的流量控制技术 .....	12
3.4 基于流的拥塞控制技术.....	28
3.5 流量调度技术.....	48
3.6 应用加速技术.....	52
3.7 智能无损网络运维.....	54
第 4 章 华为智能无损网络方案介绍（HPC 场景） .....	63



4.1 组网架构.....63

4.2 核心部件.....65

4.3 交换机数量计算 .....65



# 第1章

# HPC 高性能计算业务对网络提出更高的诉求

## 摘要

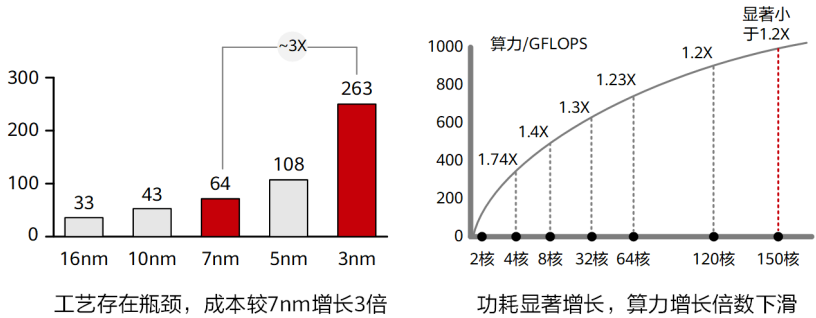
当今，数据中心正向算力中心演进，数据中心内计算集群规模不断扩大，对计算节点之间的互联网络的性能要求越来越高，数据中心网络成为数据中心算力的重要组成部分，计算和网络深度融合成为趋势。

随着 5G、大数据、物联网、AI 等新技术融入人类社会的方方面面，可以预见，在未来二三十年间人类将迈入基于数字世界的万物感知、万物互联、万物智能的智能社会。数据中心算力成为新的生产力，数据中心量纲也从原有的资源规模向算力规模转变，算力中心的概念被业界广泛接受。数据中心向算力中心演进，网络是数据中心大算力的重要组成部分，提升网络性能，可显著改进数据中心算力能效比。

为了提升算力，业界在多条路径上持续演进。单核芯片的工艺提升目前止步于 3nm；通过叠加多核提升算力，随着核数的增加，单位算力功耗也会显著增长，当 128 核增至 256 核时，总算力水平无法提升 1.2 倍。计算单元的工艺演进已经逼近基线，每 18 个月翻一番的摩尔定律即将失效，为了满足大算力的需求，HPC 高性能计



算成为常态。随着算力需求的不断增长，从 P 级向 E 级演进，计算集群规模不断扩大，对互连网络性能要求越来越高，计算和网络深度融合成为趋势。



HPC ( High Performance Computing, 高性能计算 ), 是指利用聚集起来的计算能力来处理标准工作站无法完成的科研、工业界最复杂的科学计算问题，包括仿真、建模和渲染等。由于需要大量的运算，一台通用计算机无法在合理的时间内完成工作，或者由于所需的数据量过大而可用的资源有限，导致根本无法执行计算，此时一种方式是通过使用专门或高端的硬件进行处理，但其性能往往依然很难达到要求同时较为昂贵。目前业界使用较多的方式是将多个单元的计算能力进行整合，将数据和运算相应地分布到多个单元中，从而有效地克服这些限制。

HPC 高性能计算的计算节点之间交互对网络性能的要求也是不同的，大致可以分为三类典型场景：

- 松耦合计算场景：在松耦合场景中，计算节点之间对于彼此信息的相互依赖程度较低，网络性能要求相对较低。一般金融风险评估、遥感与测绘、分子动力学等业务属于松耦合场景。该场景对于网络性能要求相对较低。
- 紧耦合场景：紧耦合场景中，对于各计算节点间彼此工作的协调、计算的同步以及信息的高速传输有很强的依赖性。一般电磁仿真、流体动力学和汽车碰撞等场景属于紧耦合场景。该场景对网络时延要求极高，需要提供低时延网络。
- 数据密集型计算场景：在数据密集型计算场景中，其特点是计算节点需要处理大量的数据，并在计算过程中产生大量的中间数据。一般气象预报、基因测序、图形渲染和能源勘探等属于数据密集型计算场景。由于该场景下计算节点处理大量数据的同时又产生了大量中间数据，所以该场景要求提供高吞吐的网络，同时对于网络时延也有一定要求。

总结一下 HPC 高性能计算对网络的诉求，高吞吐和低时延成为两个重要的关键词。同时为了实现高吞吐和低时延，业界一般采用了 RDMA（Remote Direct Memory Access，远程直接内存访问）替代了 TCP 协议，实现时延的下降和降低对服务器 CPU 的占用率。但 RDMA 协议对网络丢包非常敏感，0.01 的丢包率就会使 RDMA 吞吐率下降为 0，所以无损就成为网络的重要需求之一。关于无损网络的技术选择和技术演进，我们将在下一章进行讨论。





## 第2章

# HPC 高性能计算网络演进

---

### 摘要

传统的数据中心网络通常采用以太网技术组成多跳对称的网络架构，使用TCP/IP网络协议栈进行传输。然而传统TCP/IP网络虽然经过30年的发展技术日臻成熟，但与生俱来的技术特征使之不再适应高性能计算的业务诉求。RDMA技术逐渐代替了TCP/IP成为HPC高性能计算网络的首选协议。同时，RDMA的网络层协议选择也逐渐从昂贵的基于IB协议的无损网络向基于以太网的智能无损网络演进。本章节将为读者解释这些技术替代和演进的原因。

## 2.1 从 TCP 到 RDMA

传统的数据中心通常采用以太网技术组成多跳对称的网络架构，使用 TCP/IP 网络协议栈进行传输。但 TCP/IP 网络通信逐渐不适应高性能计算业务诉求，其主要限制有以下两点：

- 限制一：TCP/IP 协议栈处理带来数十微秒的时延

TCP 协议栈在接收/发送报文时，内核需要做多次上下文切换，每次切换需要耗费 5~10us 左右的时延，另外还需要至少三次的数据拷贝和依赖 CPU 进行协议封装，这导致仅仅协议栈处理就带来数十微秒的固定时延，使得在 AI 数据运算和 SSD 分布式存储等微秒级系统中，协议栈时延成为最明显的瓶颈。

- **限制二：TCP 协议栈处理导致服务器 CPU 负载居高不下**

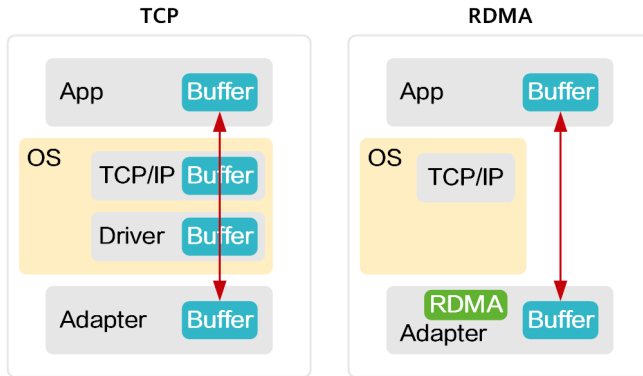
除了固定时延较长问题，TCP/IP 网络需要主机 CPU 多次参与协议栈内存拷贝。网络规模越大，网络带宽越高，CPU 在收发数据时的调度负担越大，导致 CPU 持续高负载。按照业界测算数据：每传输 1bit 数据需要耗费 1Hz 的 CPU，那么当网络带宽达到 25G 以上（满载），对于绝大多数服务器来说，至少 1 半的 CPU 能力将不得不用来传输数据。

为了降低网络时延和 CPU 占用率，服务器端产生了 RDMA 功能。RDMA 是一种直接内存访问技术，它将数据直接从一台计算机的内存传输到另一台计算机，数据从一个系统快速移动到远程系统存储器中，无需双方操作系统的介入，不需要经过处理器耗时的处理，最终达到高带宽、低时延和低资源占用率的效果。

## 2.2 从 IB 到 RoCE

如下图所示，RDMA 的内核旁路机制允许应用与网卡之间的直接数据读写，规避了 TCP/IP 的限制，将协议栈时延降低到接近 1us；同时，RDMA 的内存零拷贝机制，允许接收端直接从发送端的内存读取数据，极大的减少了 CPU 的负担，提升 CPU 的效率。举例来说，40Gbps 的 TCP/IP 流能耗尽主流服务器的所有 CPU 资源；而在使用 RDMA 的 40Gbps 场景下，CPU 占用率从 100%下降到 5%，网络时延从 ms 级降低到 10  $\mu$ s 以下。

图2-1 RDMA 与 TCP/IP 工作机制对比图



目前 RDMA 的网络层协议有三种选择。分别是 InfiniBand、iWarp ( internet Wide Area RDMA Protocol )、RoCE ( RDMA over Converged Ethernet )。

- InfiniBand 是一种专为 RDMA 设计的网络协议，由 IBTA ( InfiniBand Trade Association ) 提出，从硬件级别保证了网络无损，具有极高的吞吐量和极低的延迟。但是 InfiniBand 交换机是特定厂家提供的专用产品，采用私有协议，而绝大多数现网都采用 IP 以太网，采用 InfiniBand 无法满足互通性需求。同时封闭架构也存在厂商锁定的问题，对于未来需要大规模弹性扩展的业务系统，如果被一个厂商锁定则风险不可控。
- iWarp，一个允许在 TCP 上执行 RDMA 的网络协议，需要支持 iWarp 的特殊网卡，支持在标准以太网交换机上使用 RDMA。但是由于 TCP 协议的限制，其性能上丢失了绝大部分 RDMA 协议的优势。
- RoCE，允许应用通过以太网实现远程内存访问的网络协议，也是由 IBTA 提出，是将 RDMA 技术运用到以太网上的协议。同样支持在标准以太网交换机上使用 RDMA，只需要支持 RoCE 的特殊网卡，网络硬件侧无要求。目前 RoCE 有两个协议版本，RoCEv1 和 RoCEv2：RoCEv1 是一种链路层协议，允许在同一个广播域下的任意两台主机直接访问；RoCEv2 是一种网络层协议，可以实现路由功能，允许不同广播域下的主机通过三层访问，是基于 UDP 协议封装的。但由于 RDMA 对丢包敏感的特点，而传统以太网又是尽力而为存在丢包问题，所以需要交换机支持无损以太网。

三种协议的优势对比如下：

表2-1 RDMA 三种实现的对比

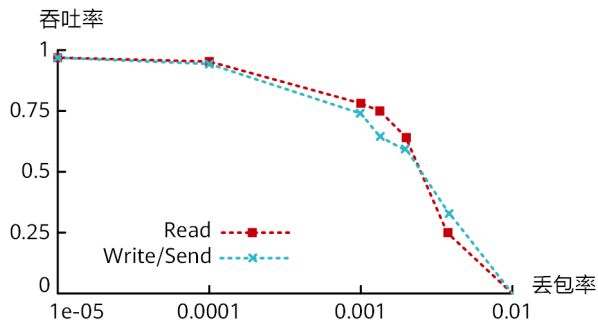
对比项	InfiniBand	iWarp	RoCE
标准组织	IBTA	IETF	IBTA
性能	最好	稍差	与 InfiniBand 相当
成本	高	中	低
网卡厂商	Mellanox-40Gbps	Chelsio-10Gbps	Mellanox-40Gbps Emulex-10/40Gbps

比较这三种技术，iWarp 由于其失去了最重要的 RDMA 的性能优势，已经逐渐被业界所抛弃。InfiniBand 的性能最好，但是由于 InfiniBand 作为专用的网络技术，无法继承用户在 IP 网络上运维的积累和平台，企业引入 InfiniBand 需要重新招聘专人的运维人员，而且当前 InfiniBand 只有很少的市场空间（不到以太网的 1%），业内有经验的运维人员严重缺乏，网络一旦出现故障，甚至无法及时修复，OPEX 极高。因此基于传统的以太网网络来承载 RDMA，也是 RDMA 大规模应用的必然。为了保障 RDMA 的性能和网络层的通信，使用 RoCEv2 承载高性能分布式应用已经成为一种趋势。

然而上文我们说过，RDMA 对于丢包是非常敏感的。TCP 协议丢包重传是大家都熟悉的机制，TCP 丢包重传是精确重传，发生重传时会去除接收端已接收到的报文，减少不必要的重传，做到丢哪个报文重传哪个。然而 RDMA 协议中，每次出现丢包，都会导致整个 message 的所有报文都重传。另外，RoCEv2 是基于无连接协议的 UDP 协议，相比面向连接的 TCP 协议，UDP 协议更加快速、占用 CPU 资源更少，但其不像 TCP 协议那样有滑动窗口、确认应答等机制来实现可靠传输，一旦出现丢包，RoCEv2 需要依靠上层应用检查到了再做重传，会大大降低 RDMA 的传输效率。

因此 RDMA 在无拥塞状态下可以满足速率传输，而一旦发生丢包重传，性能会急剧下降。如图 2-2 所示，大于 0.001 的丢包率，将导致网络有效吞吐急剧下降。0.01 的丢包率即使使得 RDMA 的吞吐率下降为 0，要使得 RDMA 吞吐不受影响，丢包率必须保证在 1e-05（十万分之一）以下，最好为零丢包。

图2-2 RDMA 网络丢包对吞吐的影响曲线图



RoCEv2 是将 RDMA 运行在传统以太网上，传统以太网是尽力而为的传输模式，无法做到零丢包，所以为了保证 RDMA 网络的高吞吐低时延，需要交换机支持无损以太网技术。而如何构建无损以太网技术我们将在下一章详细讨论。

# 第3章

# 基于以太网的智能无损网络技术

---

## 摘要

相比传统以太网技术，RDMA需要高吞吐、低时延的智能无损以太网技术。智能无损以太网可以从网络自身优化、网络与应用系统的融合优化以及运维便利性三方面去考虑网络优化问题。同时从技术上可以分为流量控制（无损）、拥塞控制（低时延与高吞吐）、流量调度（业务分层）、应用加速（性能提升）和运维五类，本章将——为读者讲解这些技术。

## 3.1 智能无损网络技术概览

总体来说，需要从以下几个方面来考虑如何提升应用性能：网络自身优化、网络与应用系统的融合优化和网络运维便利性。



## 网络自身优化

网络自身优化是指通过对网络包含设备的调整，实现网络无丢包、吞吐最高、时延最低，同时在数据中心中不同业务的优先级程度是不同的，对不同业务应该有不同的服务质量保障，从而使重要的业务能够获得更多的网络资源。所以网络自身优化主要分如下几个方面：

- 基于端口的流量控制：用于解决发送端与接收端的速率匹配问题，抑制上行出口端发送数据的速率，以便下行入口端来得及接收，防止交换机端口在拥塞的情况下出现丢包，从而实现网络无损。
- 基于流的拥塞控制：用于解决网络拥塞时对流量的速率控制问题，做到满吞吐与低时延。
- 流量调度：用于解决业务流量与网络链路的负载均衡性问题，做到不同业务流量的服务质量保障。

## 网络与应用系统的融合优化

网络与应用系统的融合优化主要在于发挥网络设备负责连通性的天然物理位置优势，与计算系统进行一定层次的配合，以提高应用系统的性能。在 HPC 高性能计算场景，当前主要的技术方向是网络设备参与计算过程，减少任务完成时间。

## 运维便利性

以太网相比于 IB 网络有其天生巨大的运维便利性优势，所以运维便利性并不是无损以太网主要需要解决的问题，其优势主要有以下几点：

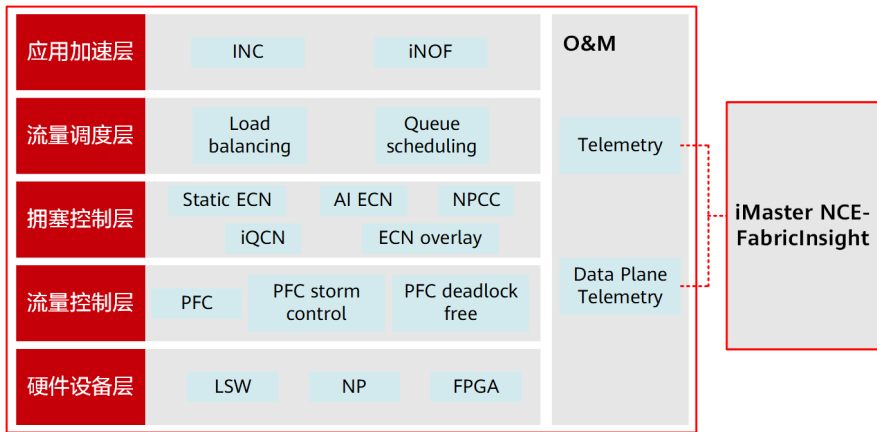
- 生态基础：以太网有强大的生态基础，而 IB 网络则相对比较小众，需要专业的运维人员，天生处于劣势。同时数据中心网络三张网均可以采用以太网构建，三网合一，只需要一个运维团队，统一的运维策略。这进一步扩大了以太网的运维便利性优势。
- 智能运维：随着计算、网络和存储的资源池化和自动化，智能运维开始成为以太网的重要运维手段，智能运维通过标准的 API 接口将网络设备的各种参数和指标发送至专业的分析器进行分析，实现自动化排除网络故障、自动化开局扩容等功能。



## 3.2 智能无损网络技术架构

基于上一个章节，智能无损网络的技术架构主要分为流量控制（无损）、拥塞控制（低时延与高吞吐）、流量调度（业务分层）、应用加速（性能提升）和运维，如下图所示。

图3-1 无损网络技术架构



- 流量控制层：主要包含 PFC、PFC 死锁检测和 PFC 死锁预防技术。PFC 是由 IEEE802.1Qbb 定义的一个优先级流控协议，主要用于解决拥塞导致的丢包问题。PFC 死锁检测和 PFC 死锁预防主要是为了解决和预防 PFC 风暴导致的一系列网络断流问题，提高网络可靠性。
- 拥塞控制层：主要包含静态 ECN、AI ECN 和 iQCN 技术。静态 ECN 是在 RFC3168（2001）中定义的一个端到端的网络拥塞通知机制，他允许网络在发生拥塞时不丢弃报文，而 AI ECN 是静态 ECN 的增强功能，可以通过 AI 算法实现 ECN 门限的动态调整，进一步提高吞吐和降低时延。iQCN 则是为了解决 TCP 与 RoCE 混跑场景下的一些时延问题。
- 流量调度层：主要为了解决业务流量与网络链路的负载均衡问题，做到不同优先级的业务流量可以获得不同等级的服务质量保障。



- 应用加速层：该层是可选的，需要根据相应的应用场景选择合适的加速技术，提升整体性能，在 HPC 高性能计算场景下，可以采用网算一体技术，让网络设备参与到计算过程中，减少任务完成时间。
- 运维：由于篇幅原因，本书主要介绍基于 Telemetry 技术的智能运维技术。

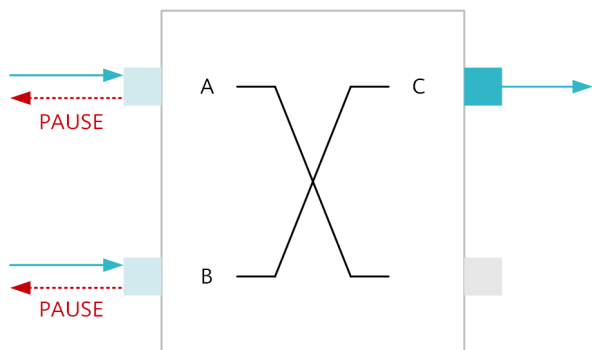
## 3.3 基于端口的流量控制技术

### 以太网 PAUSE 帧

传统的以太网通过以太 PAUSE 帧实现的流控（IEEE 802.3 Annex 31B）。当下游设备发现接收能力小于上游设备的发送能力时，会主动发 PAUSE 帧给上游设备，要求暂停流量的发送，等待一定时间后再继续发送数据。

如下图所示，端口 A 和 B 接收报文，端口 C 向外转发报文。如果端口 A 和 B 的收包速率之和大于端口 C 的带宽，那么部分报文就会缓存在设备内部的报文 buffer 中。当 buffer 的占用率达到一定程度时，端口 A 和 B 就会向外发送 PAUSE 帧，通知对端暂停发送一段时间。PAUSE 帧只能阻止对端发送普通的数据帧，不能阻止发送 MAC 控制帧。

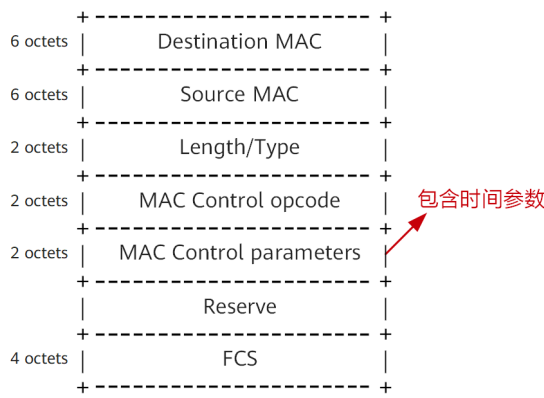
图3-2 以太 PAUSE 帧应用示意图



以上的描述有个先决条件，那就是端口 A 和 B 工作在全双工模式下，并且使能了流控功能，同时对端的端口也要开启流控功能。需要注意的是，有的以太网设备只能对 PAUSE 帧做出响应，但是并不能发送 PAUSE 帧。

以太 PAUSE 机制的基本原理不难理解，比较容易忽视的一点是：端口收到 PAUSE 帧之后，停止发送报文多长时间？其实，如下图所示，PAUSE 帧中携带了时间参数。

图3-3 PAUSE 帧格式



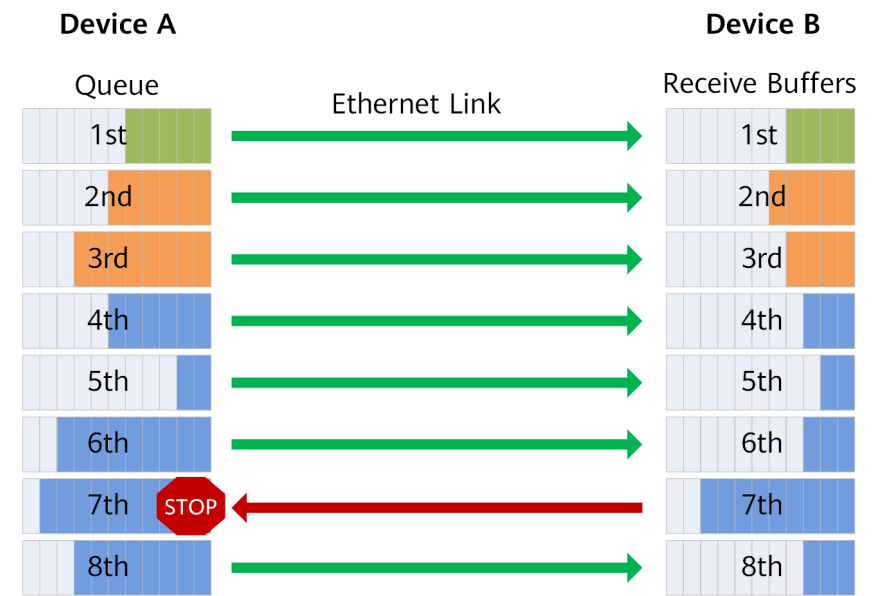
- PAUSE 帧的目的 MAC 地址是保留的 MAC 地址 0180-C200-0001，源 MAC 则是发送 PAUSE 帧的设备的 MAC 地址。
- MAC Control Opcode 域的值是 0x0001。其实，PAUSE 帧是 MAC 控制帧的一种，其他类型的 MAC 控制帧使用不同的 opcode 值。因此，通过 opcode，交换机可以识别收到的 MAC 控制帧是否是 PAUSE 帧。
- MAC Control Parameters 域需要根据 MAC Control Opcode 的类型来解析。对于 PAUSE 帧而言，该域是个 2 字节的无符号数，取值范围是 0~65535。该域的时间单位是 pause\_quanta，每个 pause\_quanta 相当于 512 比特时间。收到 PAUSE 帧的设备通过简单的解析，就可以确定停止发送的时长。对端设备出现拥塞的情况下，本端端口通常会连续收到多个 PAUSE 帧。只要对端设备的拥塞状态没有解除，相关的端口就会一直发送 PAUSE。

# PFC

基于以太 PAUSE 机制的流控虽然可以预防丢包，但是有一个不容忽视的问题。PAUSE 帧会导致一条链路上的所有报文停止发送，即在出现拥塞时会将链路上所有的流量都暂停，在服务质量要求较高的网络中，这显然是不能接受的。

PFC（Priority-based Flow Control，基于优先级的流量控制）也称为 Per Priority Pause 或 CBFC（Class Based Flow Control），是对 PAUSE 机制的一种增强。PFC 允许在一条以太网链路上创建 8 个虚拟通道，并为每条虚拟通道指定一个优先等级，允许单独暂停和重启其中任意一条虚拟通道，同时允许其它虚拟通道的流量无中断通过。这一方法使网络能够为单个虚拟链路创建零丢包类别的服务，使其能够与同一接口上的其它流量类型共存。

图3-4 PFC 的工作机制



如上图所示，DeviceA 发送接口分成了 8 个优先级队列，DeviceB 接收接口有 8 个接收缓存（buffer），两者一一对应（报文优先级和接口队列存在着——对应的映

射关系)，形成了网络中 8 个虚拟化通道，缓存大小不同使得各队列有不同的数据缓存能力。

当 DeviceB 的接口上某个接收缓存产生拥塞时，即某个设备的队列缓存消耗较快，超过一定阈值（可设定为端口队列缓存的 1/2、3/4 等比例），DeviceB 即向数据进入的方向（上游设备 DeviceA）发送反压信号“STOP”。

DeviceA 接收到反压信号，会根据反压信号指示停止发送对应优先级队列的报文，并将数据存储在本地图接口缓存。如果 DeviceA 本地图接口缓存消耗超过阈值，则继续向上游反压，如此一级级反压，直到网络终端设备，从而消除网络节点因拥塞造成的丢包。

“反压信号”实际上是一个以太网帧，其具体报文格式如下图所示。

图3-5 PFC 帧格式

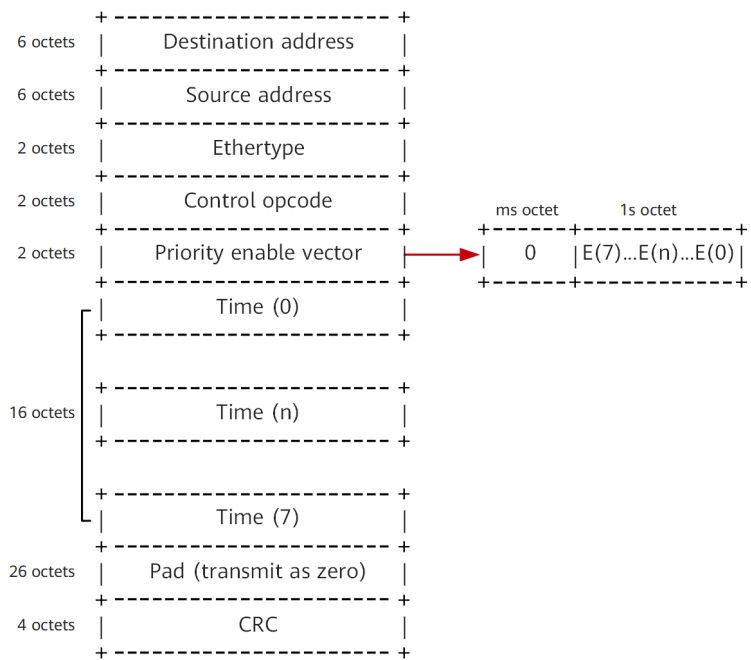


表3-1 PFC 帧的定义

项目	描述
Destination address	目的 MAC 地址，取值固定为 01-80-c2-00-00-01。
Source address	源 MAC 地址。
Ethertype	以太网帧类型，取值为 88-08。
Control opcode	控制码，取值为 01-01。



项目	描述
Priority enable vector	反压使能向量。 其中 $E(n)$ 和优先级队列 $n$ 对应，表示优先级队列 $n$ 是否需要反压。当 $E(n)=1$ 时，表示优先级队列 $n$ 需要反压，反压时间为 $Time(n)$ ；当 $E(n)=0$ 时，则表示该优先级队列不需要反压。
Time(0) ~ Time(7)	反压定时器。 当 $Time(n)=0$ 时表示取消反压。
Pad	预留。 传输时为 0。
CRC	循环冗余校验。

总而言之，设备会为端口上的 8 个队列设置各自的 PFC 门限值，当队列已使用的缓存超过 PFC 反压触发门限值时，则向上游发送 PFC 反压通知报文，通知上游设备停止发包；当队列已使用的缓存降低到 PFC 反压停止门限值以下时，则向上游发送 PFC 反压停止报文，通知上游设备重新发包，从而最终实现报文的零丢包传输。

由此可见，PFC 中流量暂停只针对某一个或几个优先级队列，不针对整个接口进行中断，每个队列都能单独进行暂停或重启，而不影响其他队列上的流量，真正实现多种流量共享链路。而对非 PFC 控制的优先级队列，系统则不进行反压处理，即在发生拥塞时将直接丢弃报文。

## PFC 线头阻塞

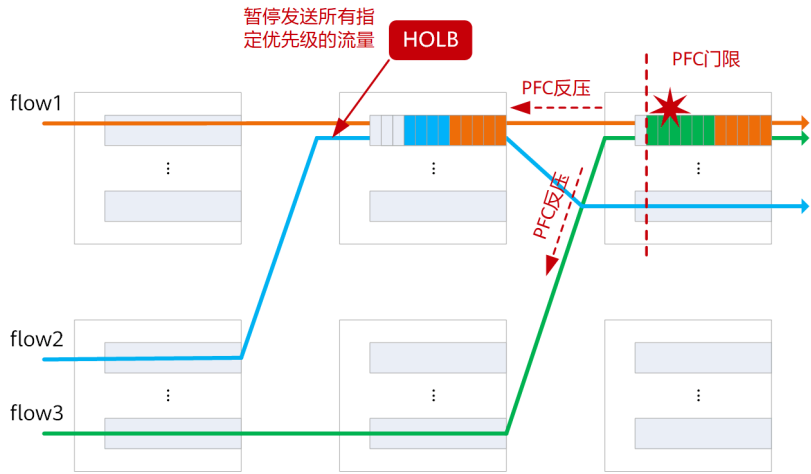
PFC 是一种有效避免丢包的流量控制技术，但由于 PFC 本质上是对以太 PAUSE 机制的一种增强，也会暂停一部分流量，这一技术应该作为最后的手段使用，否则频繁触发 PFC 会导致线头阻塞甚至死锁的问题。当交换机的某一个出口发生拥塞时，数据被缓存到备份里，并同时调用 PFC，由于 PFC 会阻止特定优先级的所有流量，所以流向其他端口的流量也有可能被阻隔，这种现象被称为线头阻塞（HOLB, head-of-line blocking）。

如下图所示，flow1、flow2、flow3 具有相同的优先级，走相同的队列。造成拥塞的是 flow1 和 flow3，flow2 在转发过程中并不存在拥塞。然而当下游端口缓存到



达 PFC 门限后，向上游端口发送的 PFC 反压信号会让上游端口停止发送所有对应优先级的队列，这样，对于 flow2 就是一个 HOLB 现象。

图3-6 PFC 线头阻塞示意图



线头阻塞可能会引起上游的额外阻塞。由于 PFC 隔离了所有流，即使是那些发往没有阻塞路径的流。这使得所有流必须在上游交换机处排队，产生的队列延时反过来又会引起上一级上游交换机的阻塞。如果上游交换机的缓存被填满，一个新的 PFC 信息会被调用并发送到网络，循环往复，造成更多的线头阻塞和阻塞现象，这被称为阻塞扩散。

为了避免线头阻塞，很有必要去尽早识别引起阻塞的流，并提供针对流特征（一般引起阻塞的流通常是大象流）的阻塞缓解技术。

## PFC 死锁

PFC 死锁，是指当多个交换机之间因为环路等原因同时出现阻塞，各自端口缓存消耗超过阈值，而又相互等待对方释放资源，从而导致所有交换机上的数据流都永久阻塞的一种网络状态。

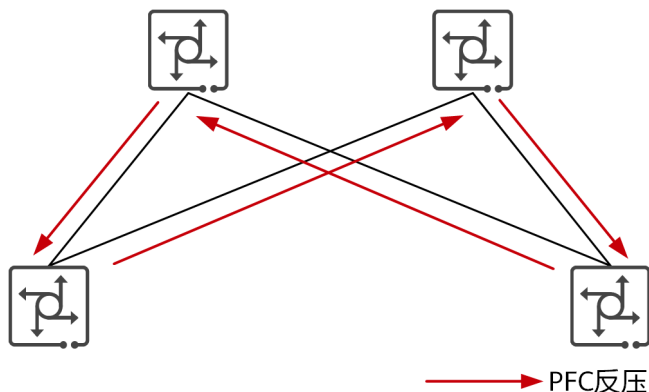
正常情况下，当一台交换机的端口出现拥塞并触发 XOFF 水线（PFC 反压帧触发门限）时，数据进入的方向（即下游设备）将发送 PFC 反压帧信号，上游设备接收到 PFC 反压帧后停止发送数据，如果其本地端口缓存消耗超过阈值，则继续向上游反压。如此一级级反压，直到网络终端服务器在反压帧中指定的暂停时间内暂停发送数据，从而消除网络节点因拥塞造成的丢包。

但在下面几个异常场景下，会出现 PFC 死锁情况。

### 场景 1：网络存在循环缓冲区依赖形成 PFC 死锁

特殊情况下，例如发生链路故障或设备故障时，BGP 路由重新收敛期间可能会出现短暂环路，会导致出现一个循环的缓冲区依赖。如下图所示，当 4 台交换机都达到 PFC 门限，都同时向对端发送 PAUSE 反压帧，这个时候该拓扑中所有交换机都处于停流状态，由于 PFC 的反压效应，整个网络或部分网络的吞吐量将变为零。即使是在无环网络中形成短暂环路时，也可能发生死锁。虽然经过修复短暂环路会很快消失，但它们造成的死锁不是暂时的，即便重启服务器中断流量，死锁也不能自动恢复。

图3-7 循环缓冲区依赖形成 PFC 死锁示意图

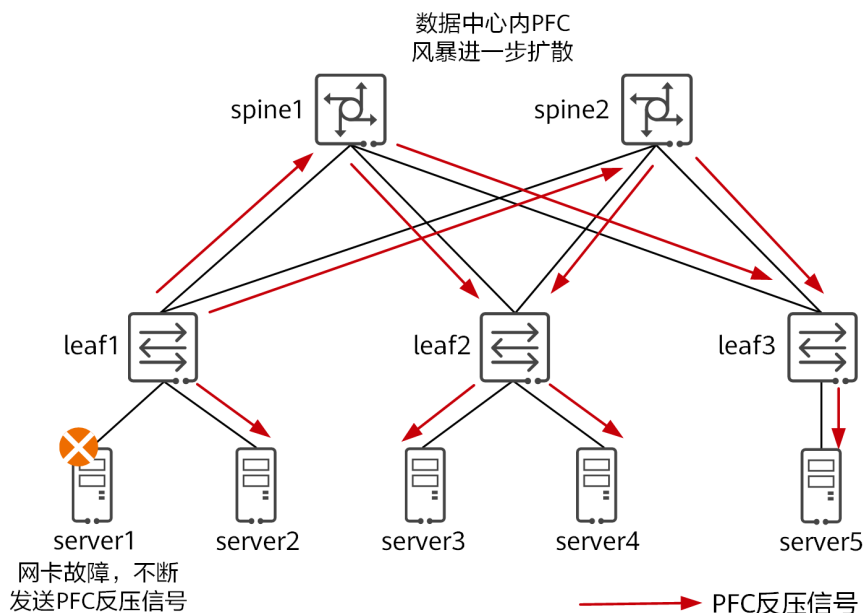


### 场景 2：服务器网卡故障引起 PFC 风暴导致 PFC 死锁

服务器网卡故障引起其不断发送 PFC 反压帧，网络内 PFC 反压帧进一步扩散，导致出现 PFC 死锁，最终将导致整网受 PFC 控制的业务的瘫痪，如下图所示。



图3-8 服务器网卡故障引起 PFC 风暴形成 PFC 死锁示意图



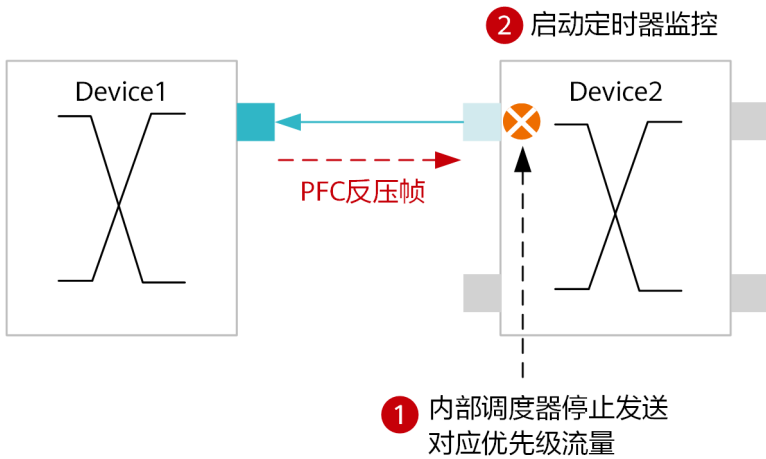
## PFC 死锁检测

由 PFC 死锁的各个场景可知，一旦出现 PFC 死锁，若不及时解除，将威胁整网的无损业务，PFC 死锁检测功能可以通过以下几个过程对 PFC 死锁进行全程监控，当设备在死锁检测周期内持续收到 PFC 反压帧时，将不会响应。

### 1. 死锁检测

Device2 的端口收到 Device1 发送的 PFC 反压帧后，内部调度器将停止发送对应优先级的队列流量，并开启定时器，根据设定的死锁检测和精度开始检测队列收到的 PFC 反压帧。

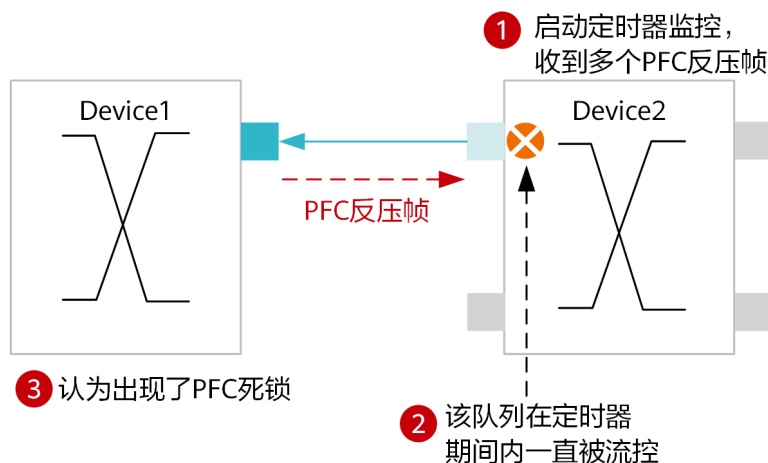
图3-9 开启死锁检测



## 2. 死锁判定

若在设定的 PFC 死锁检测时间内该队列一直处于 PFC-XOFF（即被流控）状态，则认为出现了 PFC 死锁，需要进行 PFC 死锁恢复处理流程。

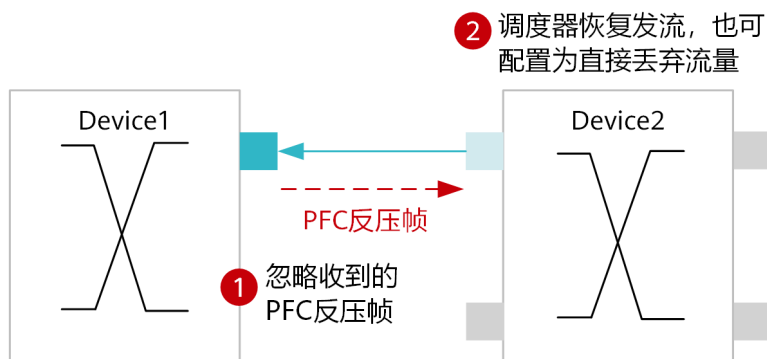
图3-10 判断出现了死锁



### 3. 死锁恢复

在 PFC 死锁恢复过程中，会忽略端口接收到的 PFC 反压帧，内部调度器会恢复发送对应优先级的队列流量，也可以选择丢弃对应优先级的队列流量，在恢复周期后恢复 PFC 的正常流控机制。若下一次死锁检测周期内仍然判断出现了死锁，那么将进行新一轮周期的死锁恢复流程。

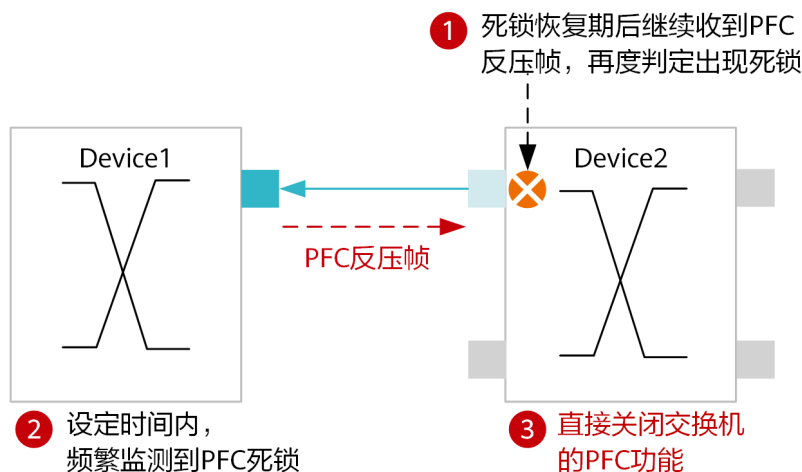
图3-11 进入死锁恢复流程



#### 4. 死锁控制

若上述死锁恢复流程没有起到作用，仍然不断出现 PFC 死锁现象，那么用户可以配置在一段时间内出现多少次死锁后，强制进入死锁控制流程。比如设定一段时间内，PFC 死锁触发了一定的次数之后，认为网络中频繁出现死锁现象，存在极大风险，此时进入死锁控制流程，设备将自动关闭 PFC 功能，需要用户手动恢复。

图3-12 频繁出现死锁可关闭 PFC 功能



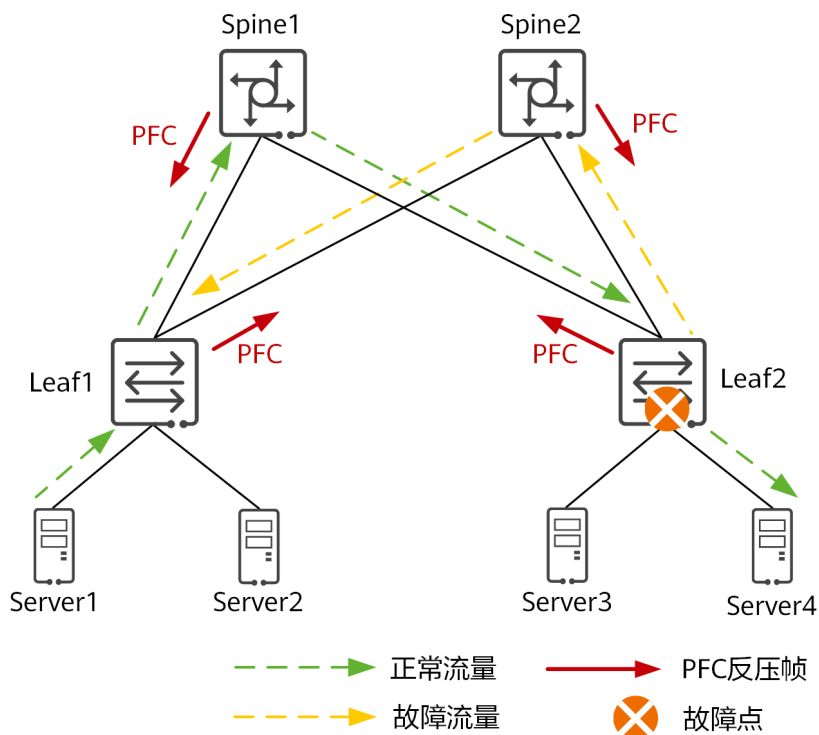
## PFC 死锁预防

PFC 死锁检测功能在死锁检测周期内持续收到 PFC 反压帧时，设备可以通过不响应反压帧的方式去解除 PFC 死锁现象。然而这种事后解锁的方式只能解决极低概率出现 PFC 死锁的场景，对于一些由于多次链路故障等原因出现环路的网络，在 PFC 死锁恢复流程后瞬间又会进入 PFC 死锁状态，网络吞吐将受到很大影响。

PFC 死锁预防正是针对数据中心网络典型的 CLOS 组网的一种事前预防的方案，通过识别易造成 PFC 死锁的业务流，修改队列优先级，改变 PFC 反压的路径，让 PFC 反压帧不会形成环路。

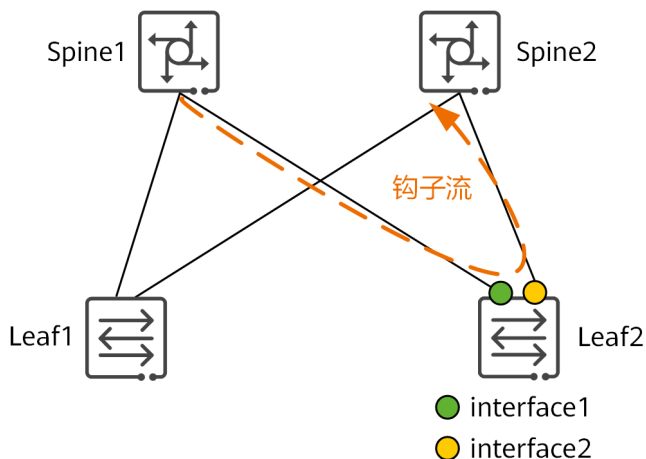
如下图所示，一条业务流的流向为：Server1-Leaf1-Spine1-Leaf2-Server4，这种正常的业务转发过程不会引起 PFC 死锁。然而若 Leaf2 与 Server4 间出现链路故障，或者 Leaf2 因为某些故障原因没有学习到 Server4 的地址，都将导致流量不从 Leaf2 的下游端口转发，而是从 Leaf2 的上游端口转发，这样 Leaf2-Spine2-Leaf1-Spine1 就形成了一个循环依赖缓冲区。当 4 台交换机的缓存占用都达到 PFC 反压帧触发门限，都同时向对端发送 PFC 反压帧停止发送某个优先级的流量，将形成 PFC 死锁状态，最终导致该优先级的流量在组网中被停止转发。

图3-13 CLOS 架构下的 PFC 死锁



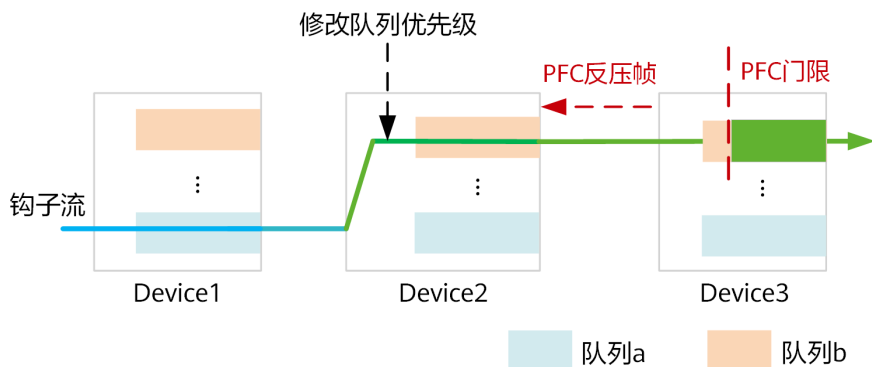
PFC 死锁预防功能中定义了 PFC 上联端口组，用户可以将一个 Leaf 设备上与 Spine 相连的接口，例如下图中的 interface1 与 interface2，都加入 PFC 上联端口组，一旦 Leaf2 设备检测到同一条业务流从属于该端口组的接口内进出，即说明该业务流是一条高风险的钩子流，易引起 PFC 死锁的现象。

图3-14 PFC 钩子流



如下图所示，Device2 识别到一条从 Device1 发过来的走队列 a 的流量为钩子流。此时 Device2 会修改该流的优先级并修改其 DSCP 值，使其从队列 b 转发。这样若该流在下游设备 Device3 引起了拥塞，触发了 PFC 门限，则会对向 Device2 的队列 b 进行反压，让 Device2 停止发送队列 b 对应优先级的流量，不会影响队列 a，避免了形成循环依赖缓冲区的可能，从而预防了 PFC 死锁的发生。

图3-15 PFC 死锁预防原理



## 总结

传统以太网由于其尽力而为的特性导致了网络丢包的存在，基于端口的流量控制技术主要是为了实现网络不丢包，传统以太网使用 PAUSE 帧实现基于端口的流量控制，但是该技术控制粒度较粗，无法完全防止丢包。基于此，提出了增强技术 PFC，PFC 通过反压机制可以实现网络零丢包，但是却存在死锁等问题，虽然有 PFC 死锁检测、死锁预防等功能可以缓解，但目前业界还没有能真正解决 PFC 死锁问题。PFC 死锁问题导致了网络吞吐下降，时延提升。所以目前业界比较多的做法是将 PFC 作为最后的终极手段，在没有必要的情况下尽量少的触发 PFC 反压。

所以 PFC 必须和基于流的拥塞控制技术共同使用，拥塞控制技术可以通过对流的控制尽量少的触发 PFC 反压，从而达到进一步提高网络吞吐和降低网络时延的目的。基于流的拥塞控制技术我们将在下一节做详细说明。



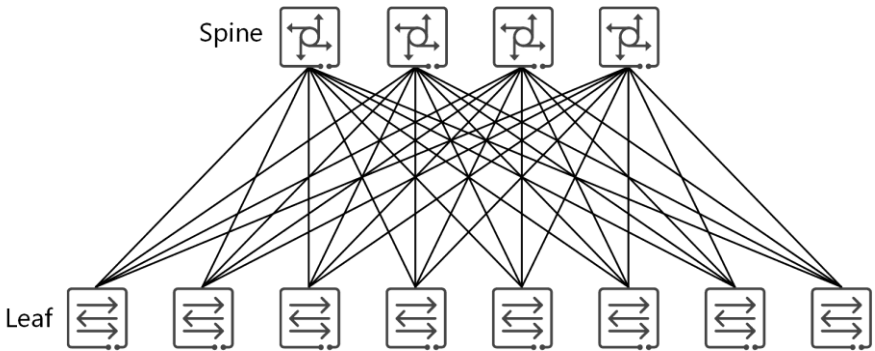
## 3.4 基于流的拥塞控制技术

### many-to-one 流量模型和 all-to-all 流量模型下的拥塞

在说明基于流的拥塞控制技术之前，笔者先介绍两种典型的拥塞场景：many-to-one 流量模型和 all-to-all 流量模型下的拥塞

图 3-16 显示了当今数据中心流行的 CLOS 网络架构：Spine+Leaf 网络架构。CLOS 网络通过等价多路径实现无阻塞性和弹性，交换机之间采用三级网络使其具有可扩展、简单、标准和易于理解等优点。除了支持 Overlay 层面技术之外，Spine+Leaf 网络架构的另一个好处就是，它提供了更为可靠的组网连接，因为 Spine 层面与 Leaf 层面是全交叉连接，任一层中的单交换机故障都不会影响整个网络结构。

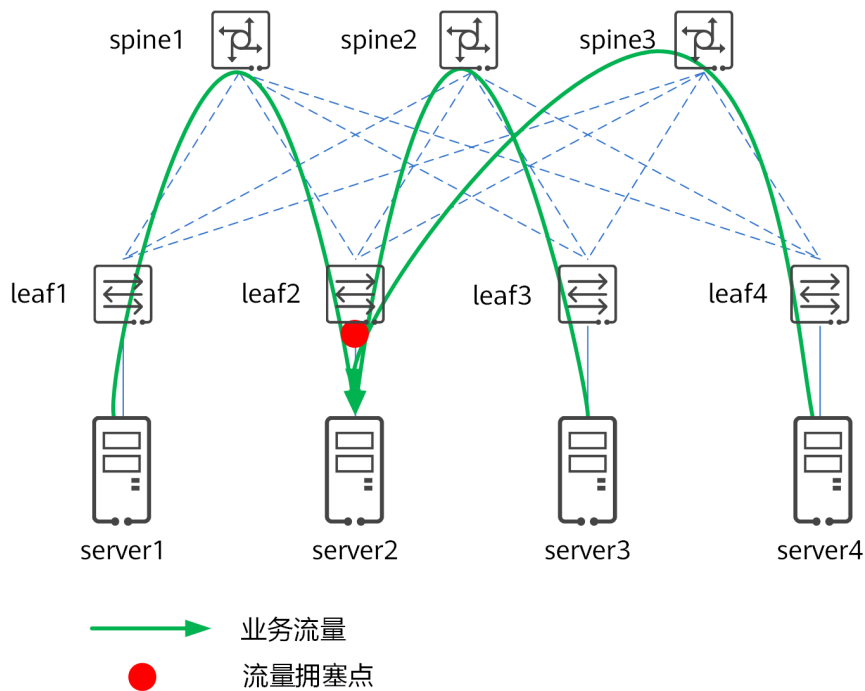
图3-16 Spine-Leaf 网络架构



然而，由于 CLOS 网络架构中的 many-to-one 流量模型和 all-to-all 流量模型，数据中心中无法避免的常常出现 Incast 现象，这是造成数据中心网络丢包的主要原因。

如图 3-17 所示，leaf1、leaf2、leaf3、leaf4 和 spine1、spine2、spine3 形成一个无阻塞的 CLOS 网络。假设服务器上部署了某分布式存储业务，某个时间内，server2 上的应用需要从 server1、server3、server4 处同时读取文件，会并发访问这几个服务器的不同数据部分。每次读取数据时，流量从 server1 到 server2、从 server3 到 server2、从 server4 到 server2，形成一个 many-to-one，这里是 3 打 1。整网无阻塞，只有 leaf2 向 server2 的方向出端口方向产生了一个 3 打 1 的 Incast 现象，此处的 buffer 是瓶颈。无论该 buffer 有多大，只要 many-to-one 持续下去，最终都会溢出，即出现丢包。

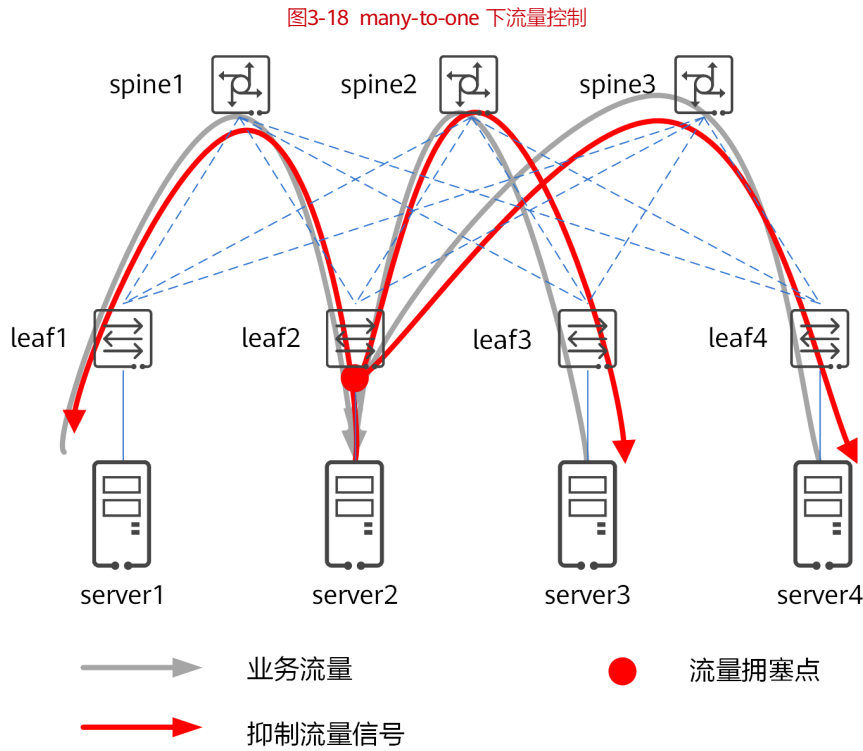
图3-17 many-to-one 流量模型示例



一旦丢包，会进一步恶化影响业务性能指标（吞吐和时延）。增加 buffer 可以缓解问题，但不能彻底解决问题，特别是随着网络规模的增加、链路带宽的增长，增加

buffer 来缓解问题的效果越来越有限。同时，大容量芯片增加 buffer 的成本越来越高，越来越不经济。

要在 many-to-one 流量模型下实现无损网络，达成无丢包损失、无时延损失、无吞吐损失，唯一的途径就是引入拥塞控制机制，目的是控制从 many 到 one 的流量、确保不超过 one 侧的容量，如图 3-18 所示。

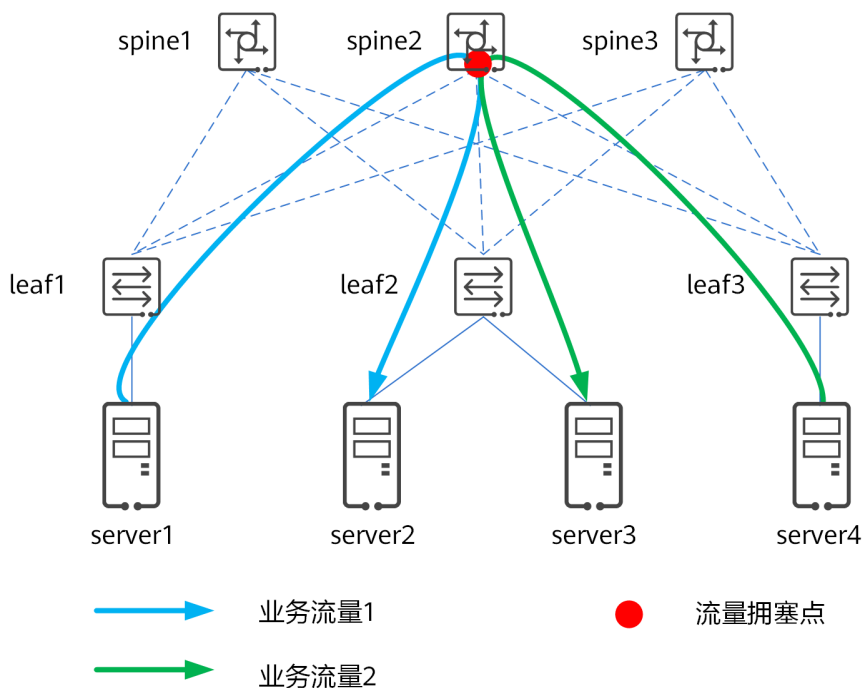


在图 3-18 中，为了保证不出现 buffer 溢出而丢包，交换机 leaf2 必须提前向源端发送信号抑制流量，同时交换机必须保留足够的 buffer 以在源端抑制流量之前接纳报文，这些操作由拥塞控制机制完成。当然，这个信号也可以由服务器 server2 分别发给 server1、server3 和 server4。

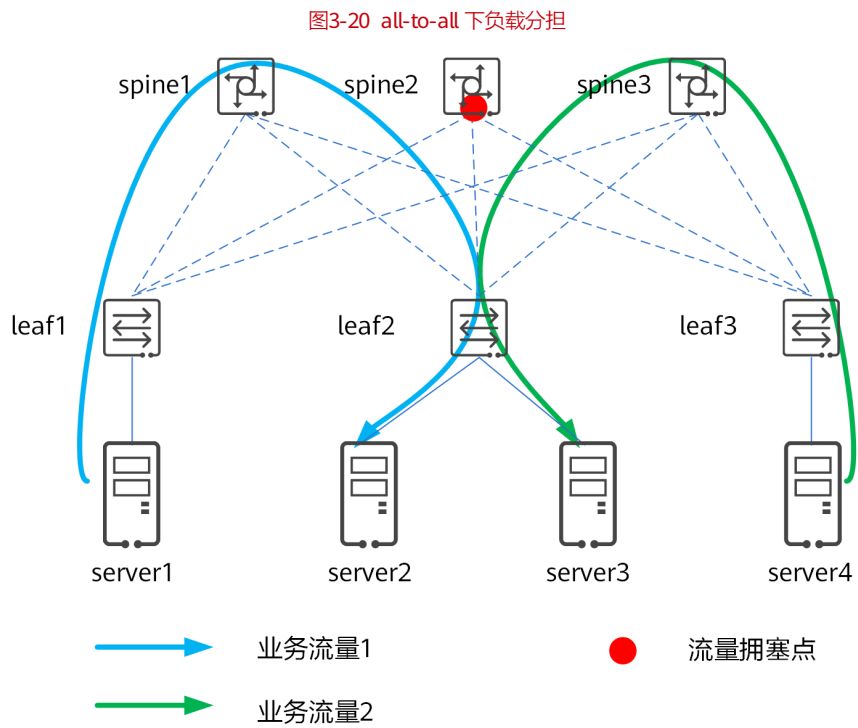
因为信号有反馈时延，为了确保不丢包，交换机是必须有足够的 buffer 以在源端抑制流量之前容纳排队的流量，buffer 机制没有可扩展性，这意味着除了拥塞控制机制之外，还需要链路级流量控制。

在图 3-19 中，leaf1、leaf2、leaf3 和 spine1、spine2、spine3 形成一个无阻塞的 CLOS 网络。假设服务器上部署了某分布式存储业务，server1 与 server4 是计算服务器，server2 与 server3 是存储服务器。当 server1 向 server2 写入数据、server4 向 server3 写入数据时，流量从 server1 到 server2、从 server4 到 server3，两个不相关的 one-to-one 形成一个 all-to-all，这里是 2 打 2。整网无阻塞，只有 spine2 向 leaf2 的方向出口方向是一个 2 打 1 的 Incast 流量，此处的 buffer 是瓶颈。无论该 buffer 有多大，只要 all-to-all 持续下去，最终都会溢出，即出现丢包。一旦丢包，会进一步恶化影响吞吐和时延。

图3-19 all-to-all 流量模型示例



要在 all-to-all 流量模型下实现无损网络，达成无丢包损失、无时延损失、无吞吐损失，需要引入负载分担，目的是控制多个 one 到 one 的流量不要在交换机上形成交叉，如图 3-20 所示，流量从 server1 到 spine1 到 server2、从 server4 到 spine3 到 server3，整网无阻塞。

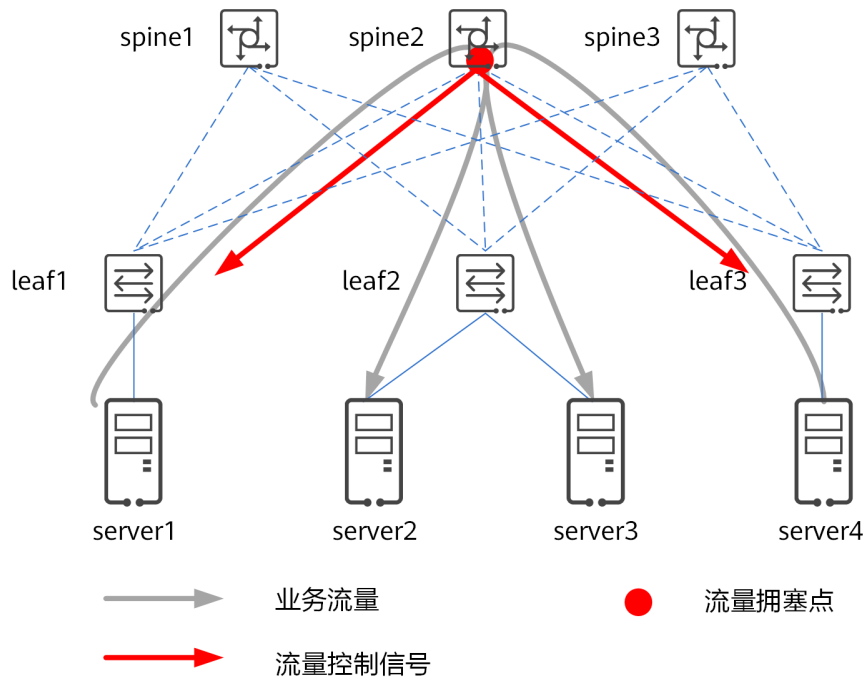


事实上，报文转发、统计复用就意味着有队列、有 buffer，不会存在完美的负载分担而不损失经济性。如果采用大 buffer 吸收拥塞队列，则成本非常高且在大规模或大容量下无法实现，比如这里单纯使用大 buffer 保证不丢包，spine2 的 buffer 必须是所有下接 leaf 的 buffer 总和。

为了整网不丢包，除了 buffer 以外，还得有流量控制机制以确保点到点间不丢包。如图 3-21 所示，all-to-all 流量模型下，采用的是“小 buffer 交换机芯片+流量

控制”，由小 buffer 的 spine2 向 leaf1 和 leaf3 发送流量控制信号，让 leaf1 和 leaf3 抑制流量的发送速率，缓解 spine2 的拥塞。

图3-21 all-to-all 下链路级流控



拥塞控制是一个全局性的过程，目的是让网络能承受现有的网络负荷。网络拥塞从根源上可以分为两类，一类是对网络或接收端处理能力过度订阅导致的 Incast 型拥塞，可产生在如 many-to-one 流量模型的数据中心网络，其根因在于多个发送端往同一个接收端同时发送报文产生了多打 1 的 Incast 流量；另一类是由于流量调度不均引起的拥塞，比如 all-to-all 流量模型的数据中心网络，其根因在于流量进行路径选择时没有考虑整网的负载分担使多条路径在同一个交换机处形成交叉。

解决 Incast 现象引起的拥塞，往往需要交换机、流量发送端、流量接收端协同作用，并结合网络中的拥塞反馈机制来调节整网流量才能起到缓解拥塞、解除拥塞的效果。

## ECN-显示拥塞通知

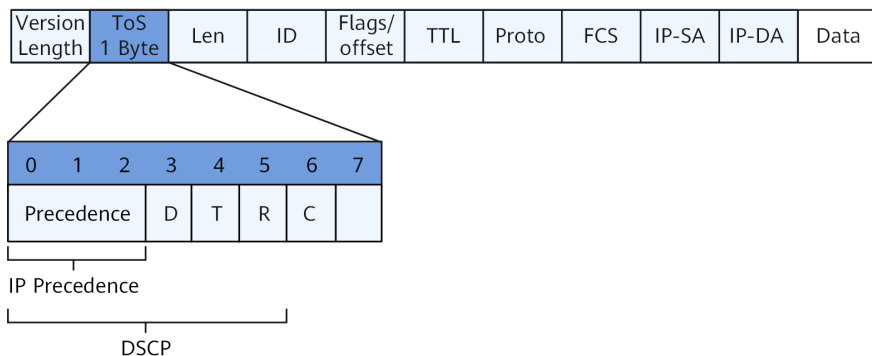
ECN (Explicit Congestion Notification) 是指流量接收端感知到网络上发生拥塞后，通过协议报文通知流量发送端，使得流量发送端降低报文的发送速率，从而从早期避免拥塞而导致的丢包，实现网络性能的最大利用，有如下优势：

- 所有流量发送端能够早期感知中间路径拥塞，并主动放缓发送速率，预防拥塞发生。
- 在中间交换机上转发的队列上，对于超过平均队列长度的报文进行 ECN 标记，并继续进行转发，不再丢弃报文。避免了报文的丢弃和报文重传。
- 由于减少了丢包，发送端不需要经过几秒或几十秒的重传定时器进行报文重传，提高了时延敏感应用的用户感受。
- 与没有部署 ECN 功能的网络相比，网络的利用率更好，不再在过载和轻载之间来回震荡。

那么，流量接收端是如何感知到网络上发生拥塞的呢？这里，需要先介绍一下 IP 报文中的 ECN 字段。

根据 RFC791 定义，IP 报文头 ToS (Type of Service) 域由 8 个比特组成，其中 3 个比特的 Precedence 字段标识了 IP 报文的优先级，Precedence 在报文中的位置如下图所示。

图3-22 IP Precedence/DSCP 字段



比特 0~2 表示 Precedence 字段，代表报文传输的 8 个优先级，按照优先级从高到低顺序取值为 7、6、5、4、3、2、1 和 0。优先级 7 和 6 一般用于承载各种协议报文，用户级应用仅能使用 0~5。

而比特 0~5 为 IP 报文的 DSCP，比特 6~7 为 ECN 字段。协议对 ECN 字段进行了如下规定：

- ECN 字段为 00，表示该报文不支持 ECN。
- ECN 字段为 01 或者 10，表示该报文支持 ECN。
- ECN 字段为 11，表示该报文的转发路径上发生了拥塞。

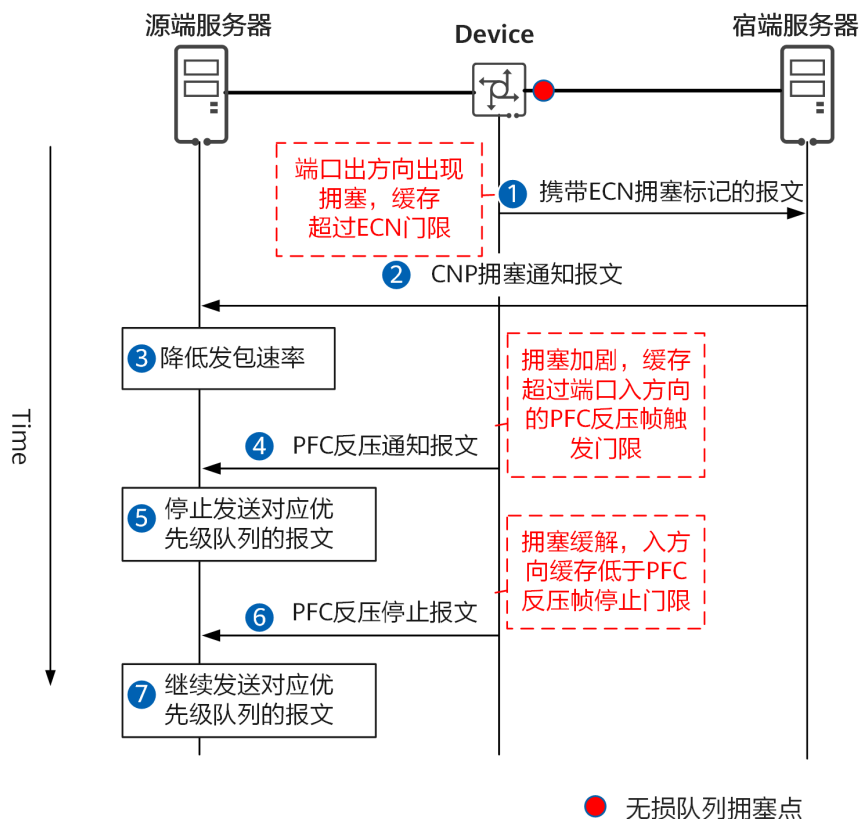
因此，中间交换机通过对将 ECN 字段置为 11，就可以通知流量接收端本交换机是否发生了拥塞。当流量接收端收到 ECN 字段为 11 的报文时，就知道网络上出现了拥塞。这时，它向流量发送端发送协议通告报文，告知流量发送端存在拥塞。流量发送端收到该协议通告报文后，就会降低报文的发送速率，避免网络中拥塞的加剧。

当网络中拥塞解除时，流量接收端不会收到 ECN 字段为 11 的报文，也就不会往流量发送端发送用于告知其网络中存在拥塞的协议通告报文。此时，流量发送端收不到协议通告报文，则认为网络中没有拥塞，从而会恢复报文的发送速率。

以上为 ECN 简单的原理，那么 ECN 门限和 PFC 门限是什么关系呢？下面以图 3-23 为例，介绍 PFC 门限和 ECN 门限的作用。



图3-23 PFC 门限和 ECN 门限减缓拥塞原理图



- 当 Device 设备的无损队列出现拥塞，队列已使用的缓存超过 ECN 门限时，Device 设备在转发报文中打上 ECN 拥塞标记（将 ECN 字段置为 11）。
- 宿端服务器收到携带 ECN 拥塞标记的报文后，向源端服务器发送 CNP 拥塞通知报文。源端服务器收到 CNP 拥塞通知报文后，降低发包速率。
- 当 Device 设备的无损队列拥塞加剧，队列已使用的缓存超过 PFC 反压帧触发门限时，Device 设备向源端服务器发送 PFC 反压通知报文。源端服务器收到 PFC 反压通知报文后，停止发送对应优先级队列的报文。

- 当 Device 设备的无损队列拥塞缓解，队列已使用的缓存低于 PFC 反压帧停止门限时，Device 设备向源端服务器发送 PFC 反压停止报文。源端服务器收到 PFC 反压停止报文后，继续发送对应优先级队列的报文。



本节中后续讨论的所有“PFC 门限”均指 PFC 反压帧触发门限，即 PFC-XOFF，PFC 反压帧停止门限 PFC-XON 不在本节讨论范围内。取值上，PFC-XON 应该小于 PFC-XOFF，确保已占用的缓存减少（拥塞已缓解）后再停止反压。

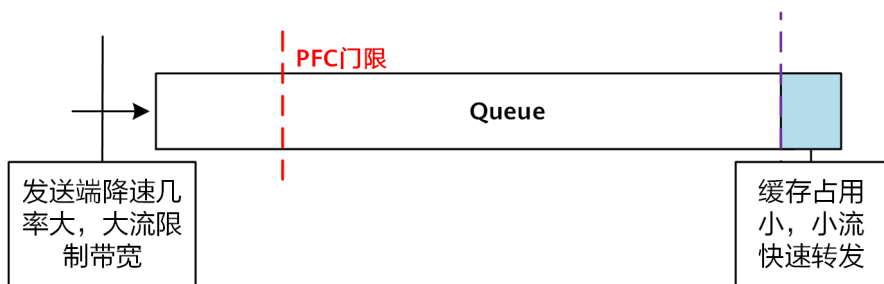
由上面的过程可以看出，从 Device 设备发现队列缓存出现拥塞触发 ECN 标记，到源端服务器感知到网络中存在拥塞降低发包速率，是需要一段时间的。在这段时间内，源端服务器仍然会按照原来的发包速率向 Device 发送流量，从而导致 Device 设备队列缓存拥塞持续恶化，最终触发 PFC 流控而暂停流量的发送。因此，需要合理设置 ECN 门限，使得 ECN 门限和 PFC 门限之间的缓存空间能够容纳 ECN 拥塞标记之后到源端降速之前这段时间发送过来的流量，尽可能的避免触发网络 PFC 流控。

## AI ECN

传统方式的 ECN 门限值是通过手工配置的，存在一定的缺陷，首先，静态的 ECN 取值无法兼顾网络中同时存在的时延敏感老鼠流和吞吐敏感大象流。ECN 门限设置偏低时，可以尽快触发 ECN 拥塞标记，通知源端服务器降速，从而维持较低的缓存深度（即较低的队列时延），对时延敏感的老鼠流有益。但是，过低的 ECN 门限会影响吞吐敏感的大象流，限制了大象流的流量带宽，无法满足大象流的高吞吐，如图 3-24 所示。

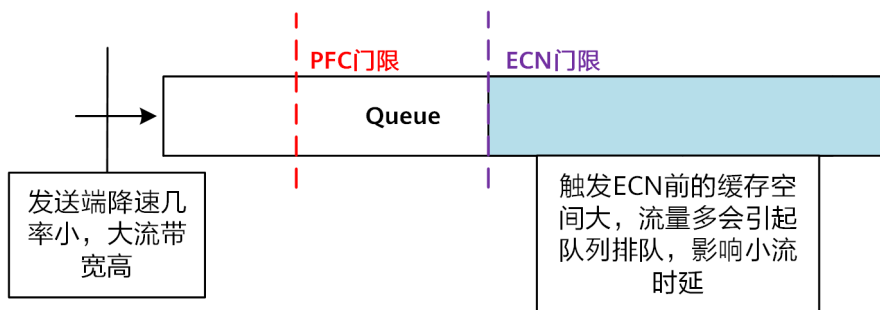


图3-24 ECN 门限偏低的影响



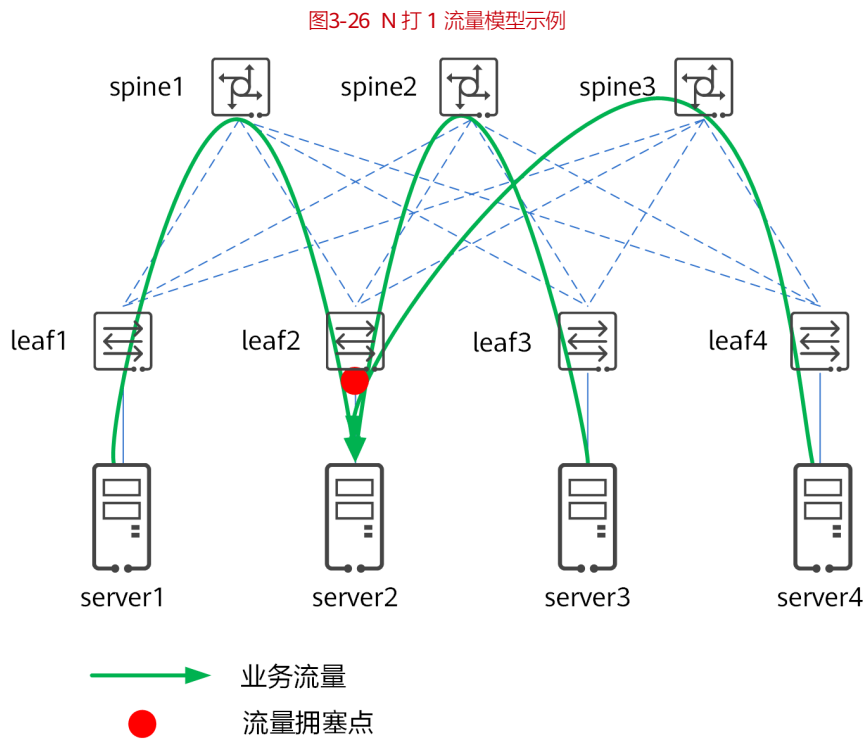
ECN 门限设置偏高时，可以延长触发 ECN 拥塞标记的时间，保障队列的突发吸收能力，满足吞吐敏感的大象流的流量带宽。但是，在队列拥塞时，由于缓存较大，会带来队列排队，引起较大的队列时延，对时延敏感的老鼠流无益。如图 3-25 所示。

图3-25 ECN 门限偏高的影响



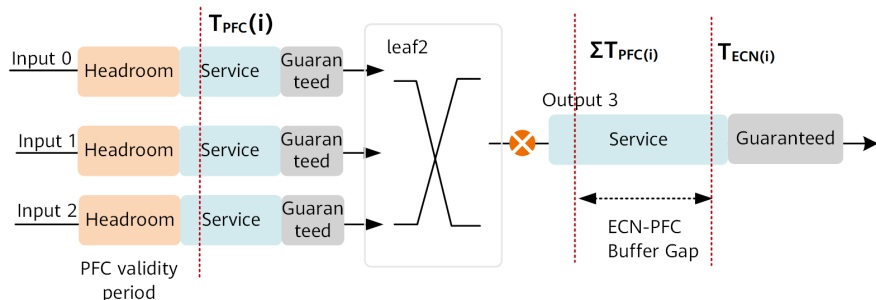
其次，固定的 ECN 门限值无法应对 Incast 场景下的突发流量。配置静态 ECN 时，为了保障无丢包，往往需要根据接入的最大链路带宽来设置 ECN 门限，然而对于高性能分布式应用，同一个交换机中的上下行队列容易出现 N 打 1 的 Incast 场景，ECN 门限需要与 PFC 门限、缓存空间大小相配合，才能满足无丢包、低时延、高吞吐的需求。

以一个 HPC 高性能计算为例，在每次任务完成同步结果时，会出现图 3-26 所示的 N 打 1 的流量模型，此时全网无阻塞，只有 leaf2 向 server2 的方向出端口方向产生了一个 3 打 1 的 Incast 现象，在 leaf2 交换机的出端口处形成了拥塞，易产生丢包。



为了避免拥塞丢包，leaf2 上引入了拥塞控制机制：PFC 功能和 ECN 功能，此时无损队列的 PFC 门限、ECN 门限和缓存的关系如图 3-27 所示（图中 T 表示 Threshold）。

图3-27 N打1场景下 leaf2 中无损队列缓存示意图



从上图中可以更清晰的获知队列缓存的作用：

- **Guaranteed**：用于保证队列的最基本转发能力，为每个优先级队列（包括无损和无损队列）提供最小可用缓存空间，不能被其他队列占用，因此各个门限都不能低于 **Guaranteed** 的缓存大小。
- **Service**：用于保证队列的突发流量转发能力，当 **Guaranteed** 的缓存已不够用时，每个队列都可以使用 **Service** 中的缓存。**Service** 的大小是扣除 **Guaranteed** 和 **Headroom** 缓存后的剩余缓存。
- **Headroom**：用于保证在 PFC 生效期间的不丢包能力，所以其大小应该大于等于 PFC 生效时间，与 PFC 反压帧的发送和接收延迟、传输链路距离和链路带宽等有关，可以通过计算获取一个相对固定的值。

为确保无丢包及不影响吞吐，N打1场景下，缓存空间和门限需要满足如下的配置要求：

1.  $\Sigma T_{PFC(i)} > T_{ECNI}$ ，即所有上行队列的 PFC 门限值相加的和应该大于下行队列的 ECN 门限值，这样设置有两个原因，首先是可以确保下行队列触发 ECN 拥塞标记之前，不会触发所有上行 PFC 流控，减少 PFC 流控的触发次数；其次，由于下行队列的流量较大，若 ECN 门限过小，会导致频繁触发 ECN，导致大量的 CNP 拥塞通知报文和过低吞吐，所以只需小于  $\Sigma T_{PFC(i)}$ 。
2.  $T_{OUT-Service} > \Sigma (T_{PFC(i)} + T_{HDM(i)})$ ，即下行队列的动态缓存门限（**Service** 缓存门限）需要大于所有上行队列的 PFC 门限值和 **Headroom** 缓存门限的和，这样设

置是为了确保所有上行触发 PFC 流控前，下行队列的缓存空间足够，不出现丢包。

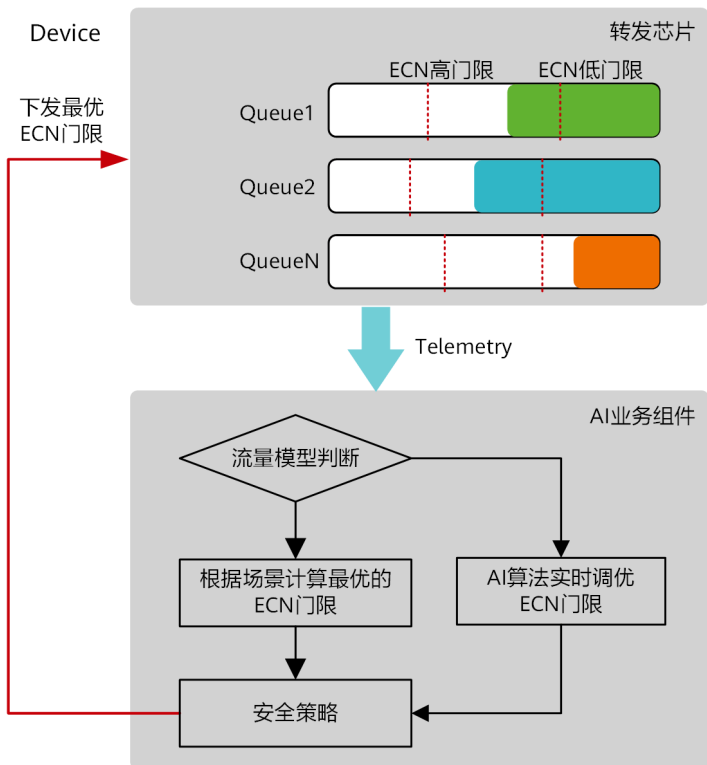
3. ECN-PFC Buffer Gap (ECN 与 PFC 的缓存间隔大小) 的取值需要合适，尽量确保从标记 ECN 到源端降速的时间差内 (也即 ECN 的生效时间)，流量不会触发 PFC，可以让 ECN 优先解决拥塞，避免或降低 PFC 的触发。因而这个 ECN-PFC Buffer Gap 将影响 PFC 触发的频率。

可见，ECN 门限的设置十分复杂，不仅与现网流量模型相关，而且与缓存空间、PFC 门限之间相互影响、相互联系，传统的静态 ECN 门限配置显然很难满足要求。这时，可以使用华为独有的 AI ECN 功能。

结合了 AI 算法的无损队列的 AI ECN 门限功能可以根据现网流量模型进行 AI 训练，对网络流量的变化进行预测，并且可以根据队列长度等流量特征调整 ECN 门限，进行队列的精确调度，保障整网的最优性能。

如图 3-28 所示，设备会对现网的流量特征进行采集并上送至 AI 业务组件，AI 业务组件将根据预加载的流量模型文件智能的为无损队列设置最佳的 ECN 门限，保障无损队列的低时延和高吞吐，从而让不同流量场景下的无损业务性能都能达到最佳。

图3-28 无损队列的 AI ECN 功能实现原理



1. **Device** 设备内的转发芯片会对当前流量的特征进行采集，比如队列缓存占用率、带宽吞吐、当前的 ECN 门限配置等，然后通过 **Telemetry** 技术将网络流量实时状态信息推送给 AI 业务组件。
2. AI 业务组件收到推送的流量状态信息后，将根据预加载的流量模型文件对当前的流量进行场景识别，判断当前的网络流量状态是否是已知场景。如果是已知场景，AI 业务组件将从积累了大量的 ECN 门限配置记忆样本的流量模型文件中，推理出与当前网络状态匹配的 ECN 门限配置。如果是未知的流量场景，AI 业务组件将结合 AI 算法，在保障高带宽、低时延的前提下，对当前的 ECN 门限不断进行实时修正，最终计算出最优的 ECN 门限配置。

3. 最后，AI 业务组件将符合安全策略的最优 ECN 门限下发到设备中，调整无损队列的 ECN 门限。
4. 对于获得的新的流量状态，设备将重复进行上述操作，从而保障无损业务的最佳性能。

无损队列的 AI ECN 门限功能可以根据现网流量模型进行 AI 训练，对网络流量的变化进行预测，并且可以根据队列长度等流量特征调整 ECN 门限，进行队列的精确调度，保障无损业务的最优性能。

同时，与拥塞管理技术（队列调度技术）配合使用时，无损队列的 AI ECN 门限功能可以实现网络中 TCP 流量与 RoCEv2 流量的混合调度，保障 RoCEv2 流量的无损传输的同时实现低时延和高吞吐。

## DCQCN

数据中心部署了满足高吞吐量、超低时延和低 CPU 开销的 RDMA 协议后，需要找到一个拥塞控制算法可以满足在该环境中保证有效运行，使网络零丢包可靠传输，因此提出了 DCQCN。DCQCN 拥塞控制算法融合了 QCN 算法和 DCTCP 算法，DCQCN 只需要可以支持 WRED 和 ECN 的数据中心交换机（市面上大多数交换机都支持），其他的协议功能在端节点主机的 NICs 上实现。DCQCN 可以提供较好的公平性，实现高带宽利用率，保证低的队列缓存占用率和较少的队列缓存抖动情况。

与 DCTCP 类似，DCQCN 算法也由三个部分组成：

- **交换机（CP, congestion point）**

CP 算法与 DCTCP 相同，如果交换机发现出端口队列超出阈值，在转发报文时就会按照一定概率给报文携带 ECN 拥塞标记（ECN 字段置为 11），以标示网络中存在拥塞。标记的过程由 WRED（Weighted Random Early Detection）功能完成。

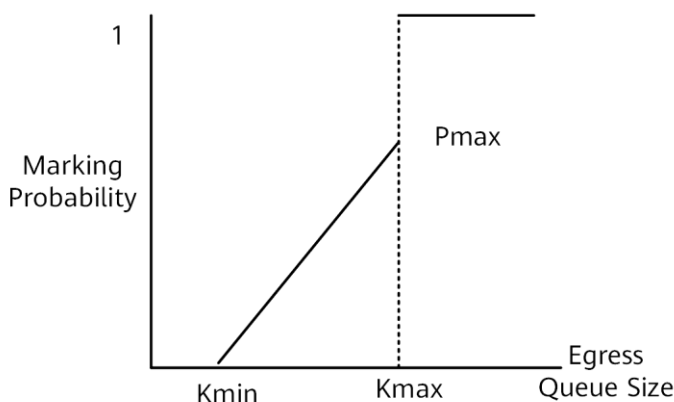
WRED 是指按照一定的丢弃策略随机丢弃队列中的报文。它可以区分报文的等级，为不同的业务报文设置不同的丢弃策略。WRED 在丢弃策略中设置了报文丢包的高/低门限以及最大丢弃概率，（该丢弃概率就是交换机对到达报文标记 ECN 的概率）。并规定：

- 当实际队列长度低于报文丢包的低门限值时，不丢弃报文，丢弃概率为 0%。



- 当实际队列长度高于报文丢包的高门限值时，丢弃所有新入队列的报文，丢弃概率为 100%。
- 当实际队列长度处于报文丢包的低门限值与高门限值之间时，随机丢弃新到来的报文。随着队列中报文长度的增加，丢弃概率线性增长，但不超过设置的最大丢弃概率。

图3-29 报文被标记的概率与队列长度关系



### ● 接收端（NP，notification point）

接收端 NP 收到报文后，发现报文中携带 ECN 拥塞标记（ECN 字段为 11），则知道网络中存在拥塞，因此向源端服务器发送 CNP 拥塞通知报文（Congestion Notification Packets），以通知源端服务器进行流量降速。

NP 算法说明了 CNPs 应该什么时间以及如何产生：如果某个流的被标记数据包到达，并且在过去的 N 微秒的时间内没有相应 CNP 被发送，此时 NP 立刻发送一个 CNP。NIC 每 N 微秒最多处理一个被标记的数据包并为该流产生一个 CNP 报文。

### ● 发送端（RP，reaction point）

当发送端 RP 收到一个 CNP 时，RP 将减小当前速率  $R_c$ ，并更新速率降低因子  $\alpha$ ，和 DCTCP 类似，并将目标速率设为当前速率，更新速率过程如下：

$$R_T = R_c$$

$$Rc = Rc * (1 - \alpha/2)$$

$$\alpha = (1 - g) * \alpha + g$$

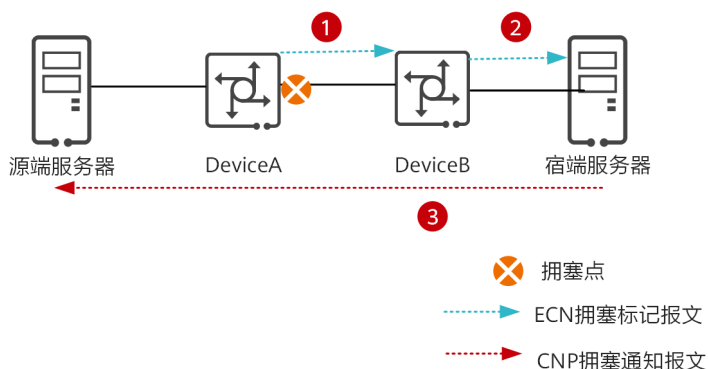
如果 RP 在 K 微秒内没有收到 CNP 拥塞通知，那么将再次更新  $\alpha$ ，此时  $\alpha = (1 - g) * \alpha$ 。注意 K 必须大于 N，即 K 必须大于 CNP 产生的时间周期。

进一步，RP 增加它的发送速率，该过程与 QCN 中的 RP 相同。

应用了 DCQCN 后，网络的处理流程如图 3-30 所示：

1. 转发设备 DeviceA 发现出端口队列缓存超出阈值，即认为出现了拥塞，因此在转发报文时就会按照一定概率给报文携带 ECN（Explicit Congestion Notification）拥塞标记（ECN 字段置为 11），以标示网络中存在拥塞。
2. 转发设备 DeviceB 收到报文后，发现报文的地址不是本机，则不对报文进行处理，正常转发给宿端服务器。
3. 宿端服务器收到报文后，发现报文中携带 ECN 拥塞标记（ECN 字段为 11），则知道网络中存在拥塞，因此向源端服务器发送 CNP（Congestion Notification Packets）拥塞通知报文，以通知源端服务器进行流量降速。

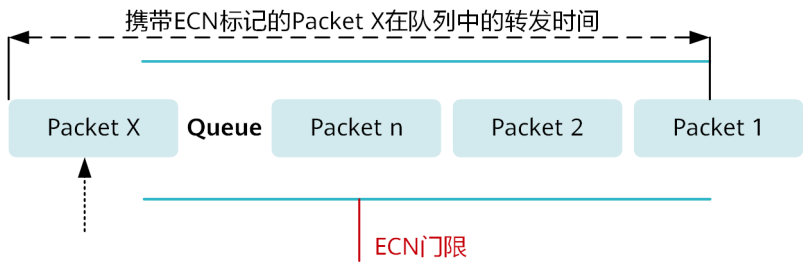
图3-30 DCQCN 处理流程图



从这里可以看出，拥塞发生点为 DeviceA，但是对拥塞进行反馈的设备却是网络尾部的宿端服务器，过长的 CNP 拥塞反馈路径使得源端服务器流量不能及时降速，从而导致转发设备缓存可能进一步拥塞恶化。

并且 DCQCN 中的 ECN 采用的是入队列标记方式，如图 3-31 所示，入队列标记方式是指报文在入队列时判断队列已使用的缓存是否超过 ECN 门限，若超过则在入队列的报文中打上 ECN 拥塞标记（将报文的 ECN 字段置为 11）。这样，宿端服务器收到携带 ECN 标记报文的时间为该报文在设备队列中的转发时间（从设备给报文打上 ECN 拥塞标记到设备将携带 ECN 拥塞标记的报文转发出去）+该报文在网络中转发的时间。在网络拥堵严重的情况下，这种入队列标记方式容易造成队列拥堵恶化。

图3-31 传统 ECN 拥塞标记方式图



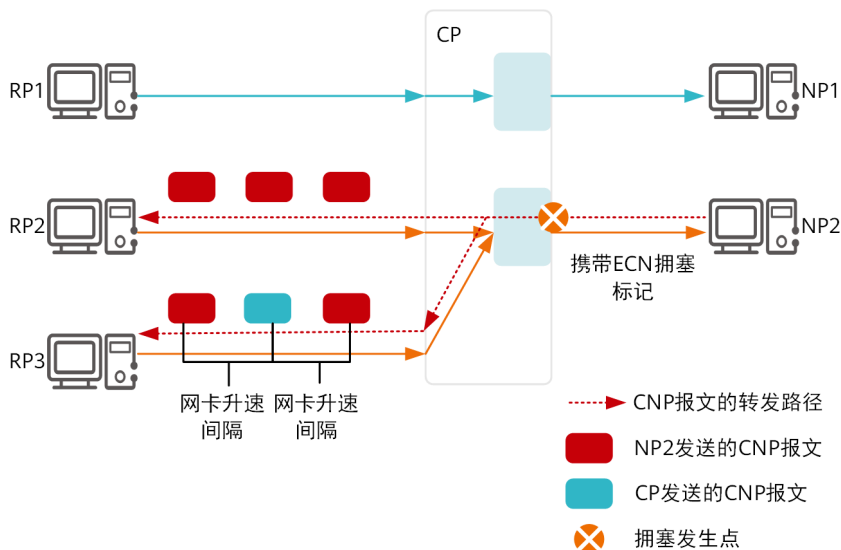
从以上内容可知，DCQCN 中过长的 CNP 拥塞反馈路径和入队列标记的 ECN 机制使得 DCQCN 控制回路时延过大，并且数据中心一般具有大规模的网络，拥有更多的跳数，因此 DCQCN 控制回路的往返时间（Round-Trip Time，RTT）会更长，在 ECN 标记生效前很难吸收突发流量，甚至导致拥塞加剧最终引发整网因 PFC 流控而暂停流量的发送。

DCQCN 引起的拥塞控制环路问题可以通过 iQCN 技术解决。

## iQCN

iQCN 功能是为了应对发送端网卡未及时收到 CNP 报文而提出的功能，iQCN 让转发设备可以智能识别网络拥塞状况，同时根据流量接收端返回 CNP 报文的时间间隔和网卡升速时间间隔，主动对 CNP 报文进行补偿发送，避免流量发送端未及时感知到 CNP 报文而升速导致拥塞加剧的情况。以下简单描述 iQCN 的工作原理。

图3-32 iQCN 工作原理



1. 报文从 RP1 发往 NP1，若 CP 没有发生拥塞，流量正常转发。
2. 报文从 RP2 和 RP3 发往 NP2，在 CP 的端口出方向发生了拥塞，CP 对报文中进行 ECN 拥塞标记后将报文转发给 NP2。
3. NP2 收到携带了 ECN 拥塞标记的报文后，获知网络中出现了拥塞，NP2 的网卡向 RP2 和 RP3 发送 CNP 拥塞通知报文，通知 RP2 和 RP3 的网卡降低发送报文的速率。若 CP 的出端口持续拥塞，则 NP2 将持续发送 CNP 拥塞通知报文。
4. 使能了 iQCN 功能的 CP 会对收到的 CNP 报文进行记录，维护包含 CNP 报文信息和时间戳的流表。同时 CP 会对本设备的端口拥塞程度进行持续监测，端口拥塞较为严重时，将收到 CNP 报文的时间间隔与网卡升速时间进行比较，若发现从 NP 收到 CNP 报文的时间间隔小于 RP 的网卡升速时间，判断网卡可以正常降速，CP 正常转发 CNP 报文。若发现从 NP 收到 CNP 报文的时间间隔大于 RP 的网卡升速时间，判断网卡不能及时降速且存在升速风险，CP 将会主动补偿发送 CNP 报文。

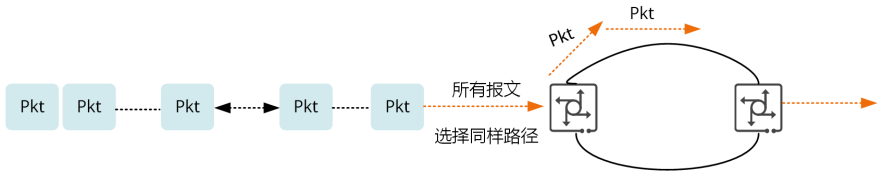
## 3.5 流量调度技术

流量调度技术主要用于解决业务流量与网络链路的负载均衡性问题，做到不同业务流量的服务质量保障。由上文可知，网络拥塞根因可以分为 Incast 流量引起的拥塞和流量调度不均引起的拥塞。实际上无损数据中心网络中流量调度不均的原因主要有两个：一个是通过报文特征字段进行流量选择的简单粗暴的逐流负载分担，容易引起等价多路径冲突；一个是网络中老鼠流和大象流交替分布，大象流容易把老鼠流堵住，大象流产生的等价多路径冲突后果更严重。

### 动态负载分担

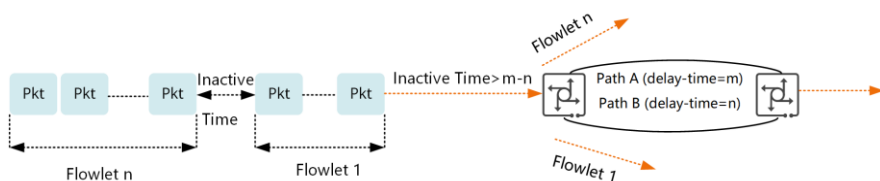
动态负载分担技术是为了解决传统静态逐流负载分担引起的等价多路径冲突问题。传统的静态哈希方式的逐流负载分担，可以看作是基于 Flow 的负载分担，Flow 是指一组具有相同特征字段的数据包。为避免报文乱序，同样的流选择同样的路径，不管流的带宽大小如何，如图 3-33 所示。

图3-33 基于流的负载分担



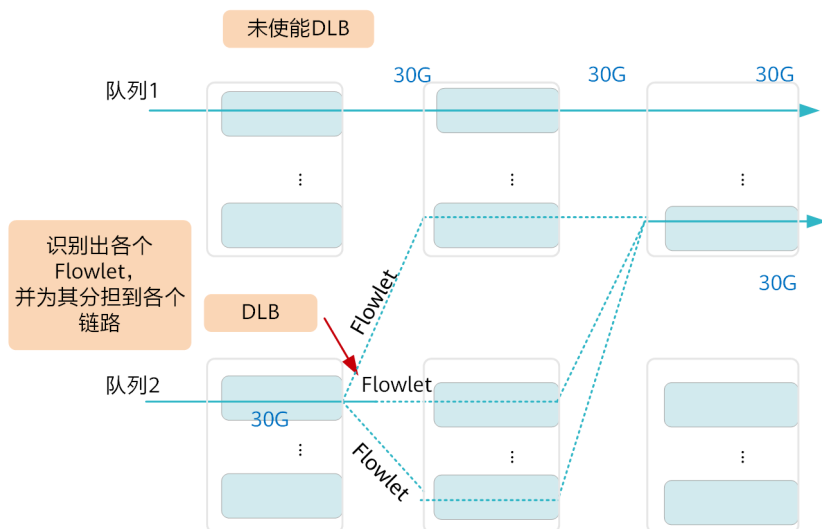
然而，智能无损网络里相当部分的流是突发流量，特征是同样的流，但为一段段的突发。显然，当两个突发之间的时间间隔大于 Path 路径间的时延差时，就可让新的一段突发重新选路而不会引起乱序。一段突发称为“Flowlet”，Flowlet 内的各个报文时间间隔会小于 Path 路径间的时延差，因此仍会选择同样的路径，以保证 Flowlet 内的各报文不乱序，如图 3-34 所示。

图3-34 Flowlet 示意



对应智能无损网络里的大象流，就可以利用 Flowlet 识别，将大象流“切割”成多个“老鼠流”，防止大象流对所选链路的带宽冲击，以 Flowlet 来选路，更加平衡各链路间的负载，如图 3-35 所示。

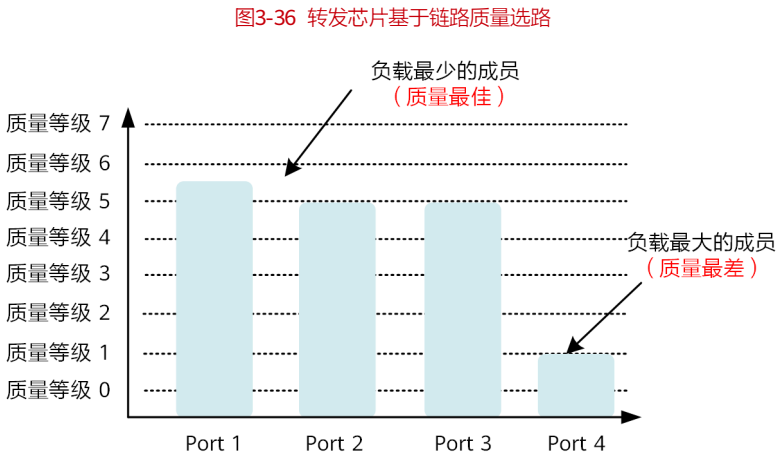
图3-35 大象流“切割”成多个老鼠流



转发芯片需要记录各个 Flowlet 所选择的成员链路，以便该 Flowlet 后面的报文仍选择同样的链路发送出去，确保不乱序。因此，转发芯片需要支持 Flowlet 流表的

学习和老化，并记录各个 Flow 的不同报文间的进入时间戳，以便区分出不同的 Flowlet。

与此同时，为了度量各个链路的拥塞状况，转发芯片需要基于各端口的缓存长度、端口带宽利用率等，来量化出各端口的“质量”。每当为 Flowlet 选链路时，选择“质量”最好拥塞最轻的链路来发送 Flowlet 报文，如图 3-36 所示。



负载分担这种根据拥塞程度来“动态”选择链路的方式，使得各链路利用更均衡，而能提升应用的性能（FCT、吞吐量）。

智能无损网络里，高性能计算应用一般都是老鼠流，谈不上 Flowlet。分布式存储应用的大象流也不一定出现能分割成多个 Flowlet 的时间间隔（可通过网卡配合“切分”出 Flowlet）。从而，对这些分布式应用的流量，就退化为基于 Flow 的动态负载分担。为了更好地分担均衡，需要基于 RDMA 流量的 IB.BTH 头字段参与哈希来学 Flow。

## 无损队列的大小流区分调度

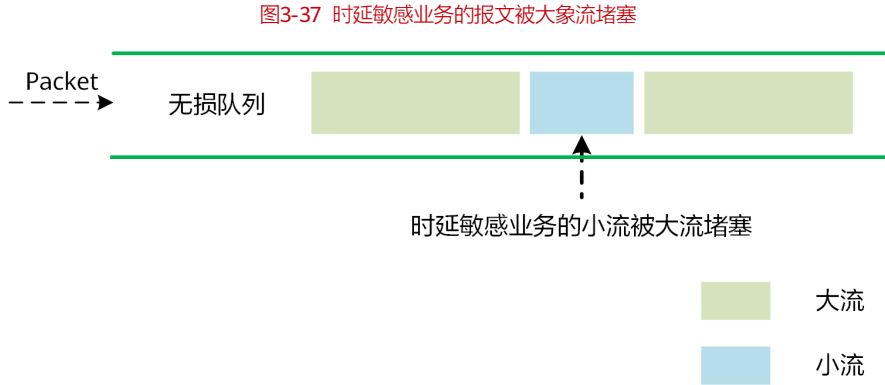
无损队列的大小流区分调度主要是为了解决大象流堵住老鼠流的问题。在设备上，每个接口的出方向都拥有 8 个队列，其队列索引分别为 0~7，其中 0 号队列优



优先级最低，7 号队列优先级最高。通常情况下，设备按照队列优先级的高低顺序对不同队列中的报文进行 PQ 调度，按照先进先出 FIFO（First In First Out）的策略对同一队列中的报文进行转发。

队列中转发报文的识别参数（例如报文的速率、长度等）大小不一，高于识别参数的报文称为大象流，低于识别参数的报文称为老鼠流。

如图 3-37 所示，在无损队列存在大象流、老鼠流混合的情况下，当队列发生拥塞时，会因为大象流引起的过深的队列长度使得老鼠流的队列时延加大，从而导致时延敏感老鼠流的流完成时间 FCT（Flow Completion Time）大大增加。更严重时，大象流会把队列堵满，后面的老鼠流会因为进入不了队列而丢包。



无损队列的大小流区分调度功能可以很好的解决上述问题。使能无损队列的大小流区分调度功能后，设备会根据大象流识别参数，来区分队列中的大象流和老鼠流。优先调度老鼠流的报文，使得老鼠流的时延不受大象流的影响，从而保障老鼠流的 FCT。

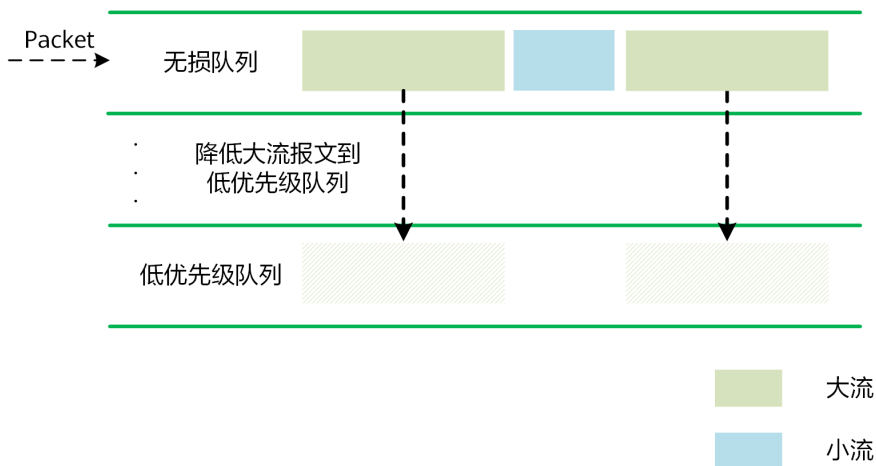
如图 3-38 所示，在无损队列上开启大小流区分调度功能后，设备按照如下方式对队列中的报文进行处理：

1. 将无损队列中的报文信息记录在流表表项中，并基于流表表项内容按照大象流识别参数识别出大象流。



2. 对识别出的大象流，降低其到低优先级队列进行转发；对于老鼠流，仍然保持其在原始优先级的队列进行转发。
3. 后续进入队列的报文，若为已识别出的大象流，则降低其到低优先级队列中进行转发；若不是已识别出的大象流，则重复上面的步骤进行处理。

图3-38 无损队列大小流区分调度实现原理



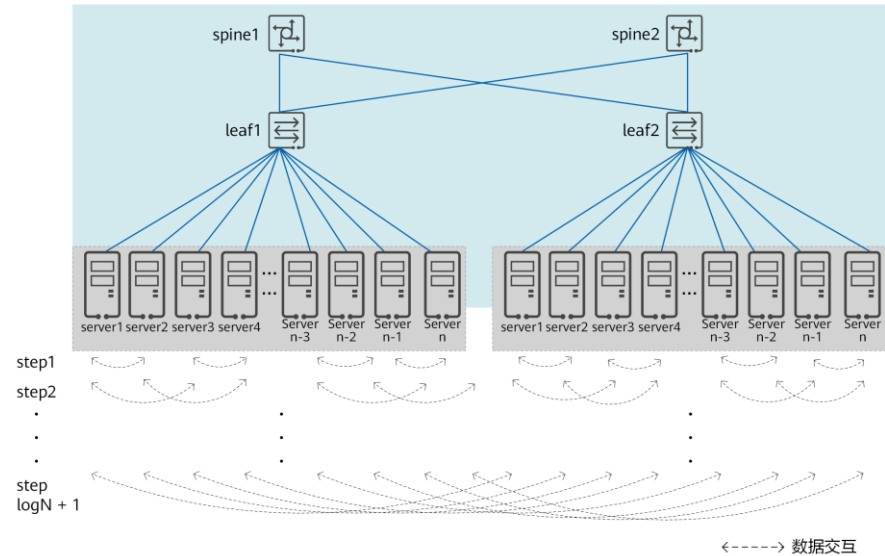
## 3.6 应用加速技术

在完成无损及高吞吐低时延的要求之后，基于以太网的智能无损网络还需要考虑应用性能问题。因为当前整体性能 IB 还领先以太网 20%左右。针对 HPC 高性能计算场景，当前可行的技术是网络能参与到计算过程中，为服务器承担一部分计算任务。华为针对 HPC 高性能计算网络提供了网算一体（Integrated Network and Computing，简称 INC）技术，网算一体技术可以提高并行计算中计算节点之间的交互效率。在开启了网算一体功能后，无损以太网的整体性能基于业务场景的不同可以提升 3%到 18%，性能可以和 IB 网络持平。

传统技术场景的数据处理和计算过程都是由服务器完成的，服务器根据指定的算法通过多次大量的数据通信交互，完成一项数据运算过程，以 Spine-Leaf 两层组网架构举例说明，其数据交互过程如下图所示。如果每台 Leaf 交换机挂接  $N$  台服务器，则每台服务器需要进行  $\log N + 1$  次通信交互才可以完成一项数据运算过程，其通信交互复杂度为  $O(\log N)$ 。由于 HPC 场景网络流量的典型特征是 80% 以上的流量是 payload 小于 16 字节的小字节报文，小字节报文的转发时延接近设备的静态时延，因此随着服务器规模和计算量的不断提升，服务器数据交互的次数会显著增加，转发时延不断增大，其对网络性能的压力也越来越大。

网算一体技术可以重点保障智能无损网络 HPC 小字节报文场景的低时延效果，降低任务完成时的通信等待时间，提升计算效率。

图3-39 传统技术场景下 MPI\_Allreduce 算法的数据交互示意图

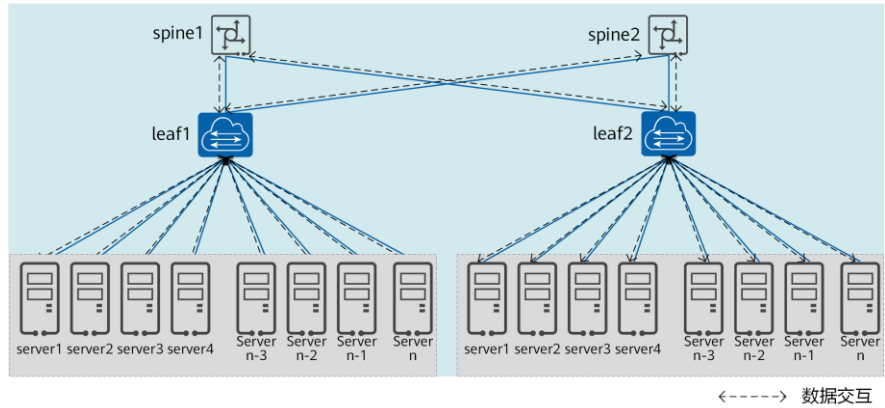


网算一体功能是指在交换机可以支持的计算能力范围内，将集合通信的部分计算过程从服务器转移到交换机设备，同时完成网络数据转发与高性能数据计算功能。



网算一体功能的数据处理过程如图 3-40 所示，最多可以支持两级组网同时启用网算一体功能。服务器将需要计算的数据封装成 MPI 报文发送到 Leaf 交换机，Leaf 节点对报文信息进行提取后由内部计算模块对数据进行一级计算；如果 Spine 交换机也配置了网算一体功能，则可以将由 Spine 交换机继续进行二级计算，计算结果再通过 MPI 报文转发到服务器节点，完成一项数据计算过程。网算一体功能的数据交互复杂度为  $O(C)$  ( $C$  表示 HPC 网络的层级数)，极大地减少了服务器集群间的通信交互过程，从而降低了 HPC 小字节报文场景下的网络时延，提升了计算效率。

图3-40 网算一体场景下 MPI\_Allreduce 算法的数据交互示意图



### 3.7 智能无损网络运维

由于以太网相比 IB 有天生的运维便利性优势，限于篇幅，本书不再详细说明网络运维的整体技术架构，仅针对基于 Telemetry 的智能运维技术做一些说明。

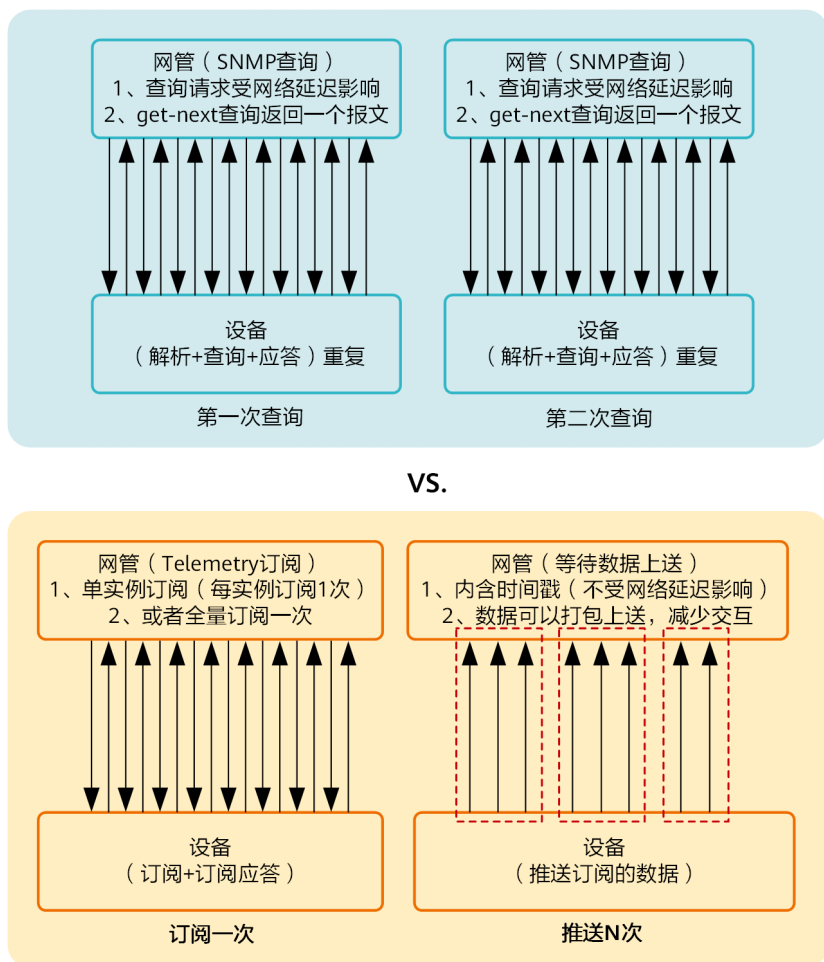
## Telemetry 主动推送机制

依据全局部署的智能分析平台 iMaster NCE-FabricInsight（以下简称为 FabricInsight），可以采集到 RoCEv2 网络状态数据，从而帮助智能无损网络从全局的视角实时修正网卡和网络的参数配置，以匹配应用的需求。

为了采集智能无损网络的流量状态，CloudEngine 系列交换机提供了可以向 FabricInsight 推送网络设备的各项高精度性能 Metrics 数据的 Telemetry 技术。

Telemetry 是一项从物理设备或虚拟设备上远程高速采集性能数据的网络监控技术。相比于传统的网络监控技术，Telemetry 通过推模式（Push Mode）高速且实时的向 FabricInsight 推送网络设备的各项高精度性能数据指标，提高了采集过程中设备和网络的利用率。

图3-41 SNMP 查询过程与 Telemetry 采样过程对比



如上图所示，与传统网络监控技术（SNMP-get）相比，Telemetry 具有如下优势：

- 通过推模式主动上送采样数据，扩大了被监控节点的规模

在传统网络监控技术中，网管与设备之间是一问一答式交互的拉模式。假设 1 分钟内需要交互 1000 次数据才能完成查询过程，则意味着设备解析了 1000 次的查询请求报文。第 2 分钟设备将再次解析 1000 次的查询请求报文，如此持续下去。实际上，第 1 分钟和第 2 分钟解析的 1000 次查询请求报文是一样的，后续设备每分钟都需要重复解析 1000 次的查询请求报文。查询请求报文的解析需要消耗设备的 CPU 资源，因此为了不影响设备的正常运行，则必须限制设备被监控节点的数量。

在 Telemetry 技术中，网管与设备之间采用的是推模式。在第 1 分钟内，网管向设备下发 1000 次的订阅报文，设备解析 1000 次的订阅报文，在解析订阅报文的过程中，设备记录下网管的订阅信息。后续每分钟内，网管不再向设备下发订阅报文，设备根据记录的订阅信息自动且持续的向网管推送数据。这样每分钟都节省了 1000 次订阅报文的解析，也就节省了设备的 CPU 资源，使得设备能够被监控更多的节点。

- **通过打包方式上送采样数据，提高了数据采集的时间精度**

在传统网络监控技术中，设备每分钟内都要解析大量的查询请求报文，且对于一个查询请求报文只上送一个采样数据。而查询请求报文的解析也需要消耗设备的 CPU 资源。因此为了不影响设备的正常运行，必须限制网管下发查询请求报文的频度，也就降低了设备数据采集的时间精度。通常来说，传统网络监控技术的采样精度为秒级。

在 Telemetry 技术中，只有第 1 分钟设备需要解析订阅报文，其他时间内设备都不需要解析订阅报文，且对于一个订阅报文可以通过打包方式上送多个采样数据，进一步减少了网管与设备之间交互报文的次数。因此，Telemetry 技术的采样精度可以达到毫秒级乃至亚秒级。

- **通过携带时间戳信息，提升了采样数据的准确性**

在传统网络监控技术中，采样数据中没有时间戳信息，由于网络传输时延的存在，网管监控到的网络节点数据并不准确。

在 Telemetry 技术中，采样数据中携带时间戳信息，网管进行数据解析时能确认采样数据的发生时间，从而避免了网络传输延迟对采样数据的影响。

通过 Telemetry 技术，可以对整个智能无损网络网络中各个无损队列的拥塞状态，比如 PFC 报文、ECN 报文、拥塞丢弃报文等等，进行实时监控，协助拥塞控制算法进行参数优化，实现 0 丢包、低时延、高吞吐的智能无损的智能无损网络网络，

最终帮助客户构建与传统以太网兼容的 RDMA，引领数据中心网络进入极速无损的高性能时代。

## RoCEv2 智能流量分析

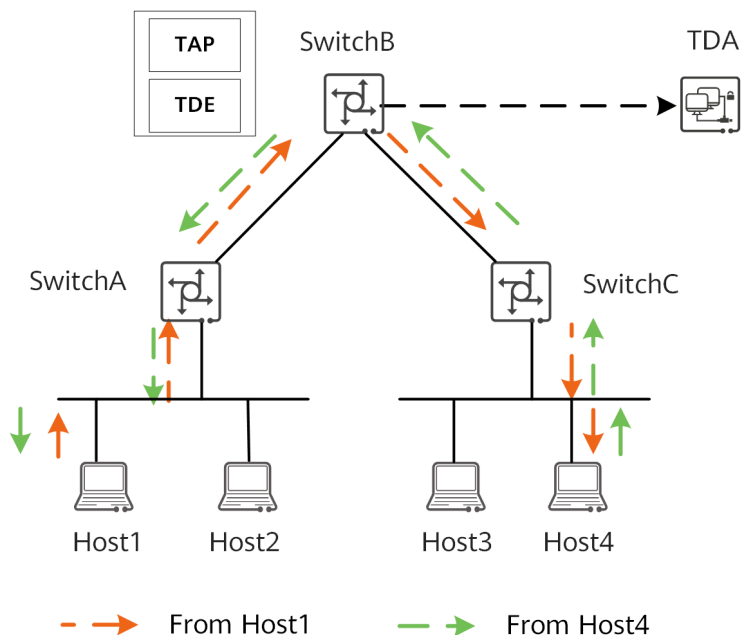
除了整网拥塞状态之外，网络管理员更关心每条 RoCEv2 流量的丢包状态、时延大小、吞吐和路径，因此针对 RoCEv2 流量实现了智能流量分析功能，将对设备经过的 RoCEv2 流量进行深度分析，并支持将分析结果发送至 FabricInsight 进行可视化展示。

一个典型的智能流量分析系统由流分析数据输出器 TDE ( Traffic-analysis Data Exporter )、流分析数据处理器 TAP ( Traffic-analysis Processor ) 和流分析数据分析器 TDA ( Traffic-analysis Data Analyzer ) 三部分组成。

- TDE：由使能了智能流量分析功能的设备承担，负责配置指定待检测的业务流，并上送到 TAP。
- TAP：由设备 CPU 内置芯片承担，对 TDE 上送的业务流进行处理和分析，并将分析结果输出至 TDA。
- TDA：表示一个网络流量分析工具，具有图形化用户界面，使用户可以方便地获取、显示和分析收集到的数据。

在实际的应用中，TDE 和 TAP 是集成在一台设备上。如下图所示，当一条指定的业务流往返方向路径相同，此时该流经过的每台设备都可以获取该流的双向流量，从而可以在这些设备上分析出该流的丢包、时延等各种指标。

图3-42 智能流量分析系统组成图



设备上使能 RoCEv2 智能流量分析功能后，TDE 会自动下发 ACL 匹配 RoCEv2 报文中的 Opcode 字段来捕获 RoCEv2 报文。通过 Opcode 字段，可以获知报文类型（是否是建链报文等），TAP 会依据 RoCEv2 建链报文中的四元组信息等关键值形成一条条的流，从而组成一个流表。

得到流表以后，TAP 根据 TDE 后续上送的 RoCEv2 数据报文，对流表中的一些关键字段进行统计，根据统计结果可以分析出 RoCEv2 流特征信息。流表中的统计内容支持在设备上查看，同时该统计结果会在流老化后输出至 FabricInsight，进一步的展示和分析。

RoCEv2 智能流量分析支持对 RoCEv2 建链报文按照四元组建流。四元组能够唯一确定一个 RoCEv2 会话，建流的四个关键值见下表。



表3-2 RoCEv2 流表 Key（关键值）

流表 Key	说明
ServerIP	指定 RoCEv2 流的服务器 IP 地址。目前仅支持 IPv4 地址。
ClientIP	指定 RoCEv2 流的客户端 IP 地址。目前仅支持 IPv4 地址。
ClientQP	指定客户端的 QP 值。该值存在于从服务器返回的 RoCEv2 流的 Dest QP 字段中。
ServerQP	指定服务器的 QP 值。该值存在于从客户端发出的 RoCEv2 流的 Dest QP 字段中。

建立 RoCEv2 智能流量分析流表后，TAP 会根据后续的 RoCEv2 数据报文统计流表中的字段，分析该流的特征信息。建议配置 1588v2 功能，提高 RoCEv2 智能流量分析功能分析的特征信息的精度。

TAP 能够分析的主要特征信息如表 3-3 所示。

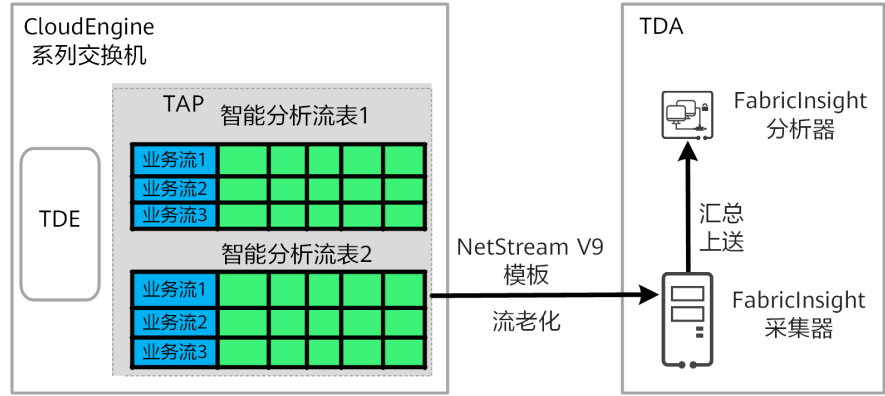
表3-3 RoCEv2 智能流量分析流特征信息

特征信息	详细内容
丢包	支持分别统计往返方向的 RoCEv2 流中 NAK 报文的数量，当值不为 0 时，说明发生了丢包。当 RoCEv2 报文出现丢包后，TAP 会记录丢包情况，添加时间戳信息后将丢包信息上送给 FabricInsight。
时延	支持分别统计双向的报文往返时延 RTT（Round Trip Time），该时延为基于双向报文计算的滑动平均时延，精度为纳秒级。
吞吐	支持统计单位时间内 Rocev2 流的吞吐率。对于 RoCEv2 报文来说，只有大象流可以计算吞吐率，老鼠流的 Rocev2 报文无法计算出对应的吞吐信息。
路径	支持统计 RoCEv2 报文的入端口信息并上送到 FabricInsight，全网配置 RoCEv2 智能流量分析功能后，即可在 FabricInsight 上看到相应的 RoCEv2 流在网络中的实际路径。

TAP 依据 TDE 上送的 RoCEv2 报文建立流表后，还需要把包含流分析结果的 RoCEv2 智能流量分析流表输出给指定的 FabricInsight，才能完成流信息的进一步加工和可视化。FabricInsight 分为 FabricInsight 采集器和 FabricInsight 分析器。

如下图所示，包含流分析结果的智能流量分析流表首先会被存储在设备的缓存区中，当缓存区中的智能流量分析流表达到老化条件时，设备会把缓存区中的智能流量分析流表输出给 FabricInsight 采集器，再由采集器将内容汇总上送给 FabricInsight 分析器，完成流特征信息的最终处理和展示。

图3-43 智能流量分析流表输出



流老化是智能分析流表输出到 FabricInsight 采集器的前提。具体来说，即流表在缓存区中到达了用户设置的老化（aging）时间或老化条件时，就会被设备发送到采集器。RoCEv2 智能流量分析流老化分为以下两类，在设备上同时配置多种老化方式后，当某一满足任一老化条件时，该流老化。

- 活跃流的老化：当一条智能流量分析流的活跃时间（从流创建时间到当前的时间）超过所设置的活跃老化时间时，设备认为该流处于活跃状态，该流将被周期性输出到 FabricInsight，输出周期正是设置的活跃老化时间。由于一旦检测到 NAK 报文数量的值不为 0 时即说明出现了丢包，当到达活跃老化时间后，TAP 还会将流表中的 NAK 报文的统计信息删除，为下一个活跃老化周期内的丢包检测做准备。若流表中的时延或吞吐信息在某个活跃老化周期内的统计信息不再变化，TAP 会将该统计信息删除，在下一个活跃老化周期内重新进行统计。
- 非活跃流的老化：由于网络上的流是短时间阵发的，在短时间内就会产生大量的流，而 TAP 的缓存空间容量是一定的，当一条 RoCEv2 智能流量分析流的非活跃时间（从流最后一个报文流过时间到当前的时间）超过所设置的非活跃老化时



间时，设备认为该流处于非活跃状态（流已经断了），这样就需要把当前的流表输出至 FabricInsight 并从缓存空间中删除，为后面到来的流提供空间，这个过程称为非活跃流老化。

该种老化方式主用于短时流量，流量停止则立即输出流表信息，节省内存空间。

## 第4章

# 华为智能无损网络方案介绍 ( HPC 场景 )

---

### 摘要

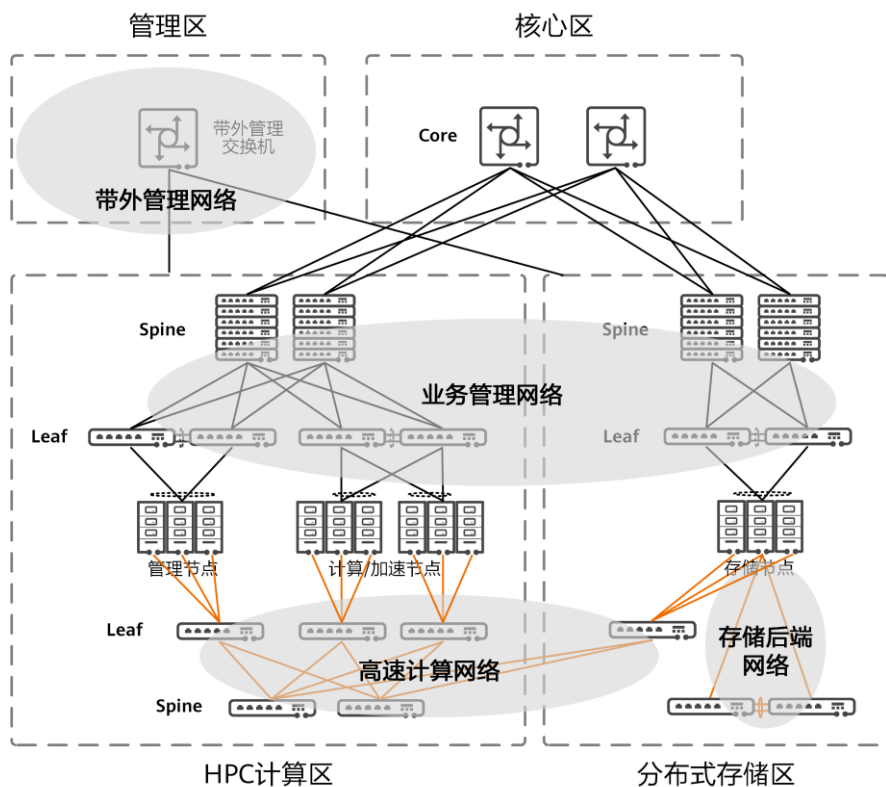
本章介绍了华为针对HPC高性能计算场景提供的数据中心智能无损网络方案，包括常见的组网架构、核心部件和不同计算规模下的网络侧设备规划。

## 4.1 组网架构

一个完整的 HPC 集群，包含高速计算网络、业务管理网络、带外管理网络、存储后端网络，如下图所示。



图4-1 HPC 组网架构简介



- **高速计算网络:** 用于计算节点之间、计算和存储节点之间，以及存储节点之间前台数据的通信。从网络角度，必须设计低延时、高带宽的互联网络来实现计算节点间的数据传输，提供足够的数据来满足单节点的计算能力。
- **业务管理网络:** 用于管理节点上集群管理软件收集集群各个节点状态信息（如 CPU 状态、内存使用率、磁盘使用率、在线状态等），并实现管理功能（如时间同步、集群部署、用户管理、作业调度等）。
- **带外管理网络:** 用于带外监控和管理集群中物理设备的状态。
- **存储后端网络:** 用于处理存储节点后台数据通信。

## 4.2 核心部件

华为智能无损网络推荐采用 CloudEngine 系列交换机进行组网，基于网络规模的不同，推荐的款型及交换机数量也不尽相同，不同网络规模款型推荐参见下表。

表4-1 不同网络规模设备款型推荐

接入端口规模	Leaf 推荐款型	Spine 推荐款型	部署说明
接入的 100GE 接口小于 2000 个	CloudEngine 8850-64CQ-EI	CloudEngine 8850-64CQ-EI	1. Leaf 交换机上下行收敛比建议 1:1，32 个接口上行，32 个接口接入。  2. Spine 交换机与 Leaf 交换机 100GE 全互联，无上行口。
接入的 100GE 接口在 2000~4000 个	CloudEngine 8850-64CQ-EI	CloudEngine 9860-4C-EI	
	CloudEngine 8851-32CQ8DQ-P	CloudEngine 9860-4C-EI	
接入的 100GE 接口大于 4000 个	CloudEngine 8851-32CQ8DQ-P	CloudEngine 16800 (CEL36DQHG-P)	

补充说明如下：

1. 服务器网卡推荐 Mellanox ConnectX-5 系列 100GE 网卡。
2. 实际项目中对超过 4K 小于 12K 接入节点的组网，优先选择 CE16808，可以进一步降低成本。
3. CE8851 与 CE9860 之间优先推荐 400G 互联，组网规模较大时可选择 100G 互联（此时 Leaf 交换机上行 400G 口一分四）。

## 4.3 交换机数量计算

假设 HPC 集群计算节点数 P，存储节点数 Q，计算节点采用 100G 单上行，存储节点采用 100G 双上行接入；Leaf 交换机 32 口下行，32 口上行，与 Spine 交换机 100G 互联，构建 1:1 收敛比无阻塞网络。



计算方法如下：

- 接入端口数  $N=P+2Q$ ;
- Leaf 交换机数量  $Y=(N/\text{Leaf 交换机下行端口数 } 32)$  再向上取整数;
- Spine 交换机数量  $X=(\text{Leaf 交换机上行口数量} \times Y / \text{Spine 交换机接口数量})$  再向上取能被 32 整除的数;
- Leaf 交换机连接 Spine 交换机 100GE 线缆数量  $=32 \times Y$ ;
- 服务器连接 Leaf 交换机 100GE 线缆数量  $=N$ ;

表4-2 不同规模 RoCE 网络交换机数量参考表

接入端口数 N	Spine 交换机 数量 X	Spine 交换机选型	Leaf 交换机数量 Y	Leaf 交换机选型
<65	0	NA	1	CloudEngine 8850-64CQ-EI
65~128	2	CloudEngine 8850-64CQ-EI	ROUNDUP ( N/32,0 )	CloudEngine 8850-64CQ-EI
129~256	4	CloudEngine 8850-64CQ-EI	ROUNDUP ( N/32,0 )	CloudEngine 8850-64CQ-EI
257~512	8	CloudEngine 8850-64CQ-EI	ROUNDUP ( N/32,0 )	CloudEngine 8850-64CQ-EI
513~1024	16	CloudEngine 8850-64CQ-EI	ROUNDUP ( N/32,0 )	CloudEngine 8850-64CQ-EI
1025~2048	32	CloudEngine 8850-64CQ-EI	ROUNDUP ( N/32,0 )	CloudEngine 8850-64CQ-EI
2049~4096	32	CloudEngine 9860-4C-EI	ROUNDUP ( N/32,0 )	CloudEngine 8850-64CQ-EI/ CloudEngine 8851-32CQ8DQ-P



接入端口数 N	Spine 交换机 数量 X	Spine 交换机选型	Leaf 交换机数量 Y	Leaf 交换机选 型
4096~9216	8	CloudEngine 16808 ( CEL36DQHG-P )	ROUNDUP ( N/32,0 )	CloudEngine 8851- 32CQ8DQ-P
9217~18432	8	CloudEngine 16816 ( CEL36DQHG-P )	ROUNDUP ( N/32,0 )	CloudEngine 8851- 32CQ8DQ-P

补充说明如下：

1. 实际项目中评估时需要综合考虑整体成本，尽量避免接入端口数 N 刚超过区间临界值的情况。比如在满足算力需求的前提下，可以通过调整计算集群中 CPU/GPU 节点比例调整接入端口数量，避免因增加几个端口要多增加一倍 Spine 交换机。
2. Spine 交换机台数合法取值为 2 的幂次方：2，4，8 等，最大不超过 Leaf 交换机上行接口数；非 2 的幂次方，存在 HASH 不均，不推荐。
3. Spine 交换机优先推荐 CloudEngine 16808，超过 9216 个 100G 接入端口的超大规模网络用 CloudEngine 16816 部署。





**联系我们**

[networkinfo@huawei.com](mailto:networkinfo@huawei.com)

**获取更多 IP 网络系列丛书**

<https://e.huawei.com/cn/solutions/enterprise-networks/ip-ebook>

