Unlocking ECMP Programmability for Precise Traffic Control

Yadong Liu, Tencent; Yunming Xiao, University of Michigan; Xuan Zhang, Weizhen Dang, Huihui Liu, Xiang Li, and Zekun He, Tencent; Jilong Wang, Tsinghua University; Aleksandar Kuzmanovic, Northwestern University; Ang Chen, University of Michigan; Congcong Miao, Tencent

汇报人：陆俊安

2025.10.12

作者背景

- **工作经历**
  - 2024年博士毕业于美国西北大学，导师：Aleksandar Kuzmanovic
  - 2024年毕业后于密歇根大学担任研究员
  - 2025年7月加入香港中文大学担任助理教授
- **近期论文**
  - An RDMA-First Object Storage System with SmartNIC Offload
  - Exposing RDMA NIC Resources for Software-Defined Scheduling
  - Cloud Infrastructure Management in the Age of AI Agents
- **研究兴趣**
  - 计算机系统
  - 计算机网络
  - 网络安全
- 获得过APNet'25 最佳论文奖和ACM EuroSys'24最佳学生论文奖

Yunming Xiao

[肖蕴明]

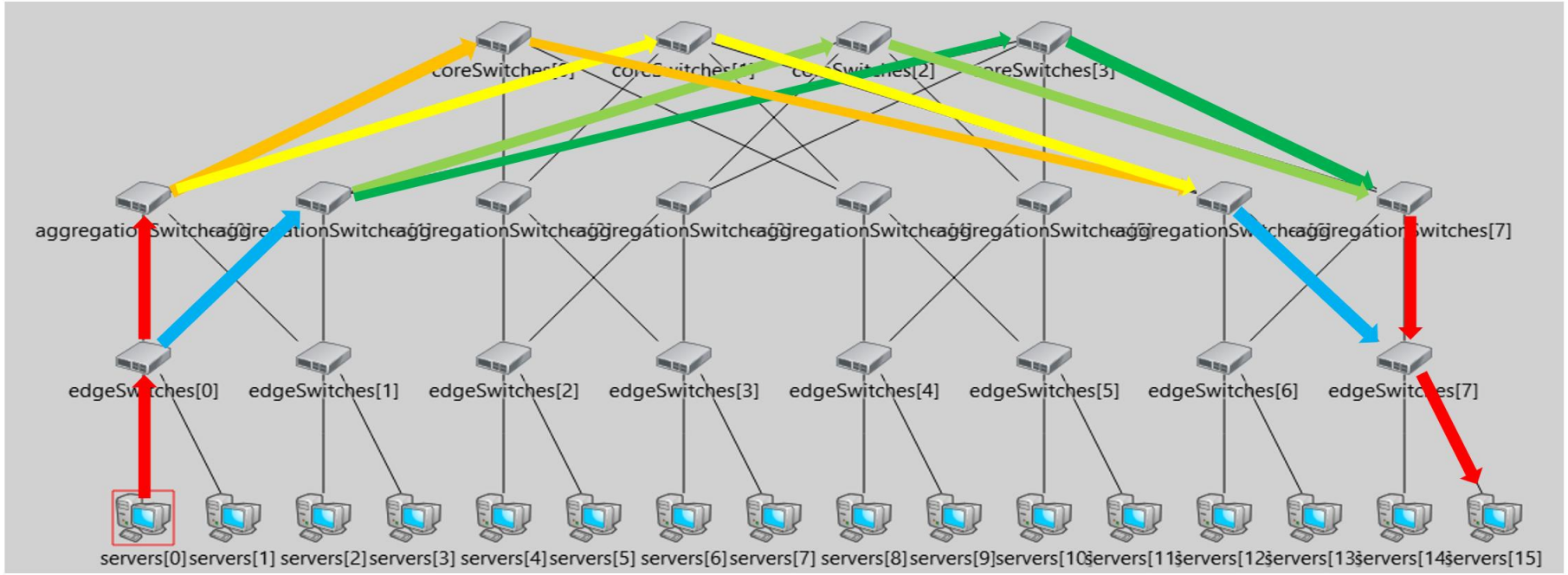
目录

研究背景

- **等价多路径（ECMP）的重要性和问题：**
  - **重要性**：现代数据中心的基础技术，在统计意义上优化了整体负载
  - **问题**：存在一类重要的精确流量控制（PTC）的任务，与ECMP的随机性本质相冲突。
  - **需求**：ECMP过于重要，需要找到一种能让PTC和ECMP共存的办法
- **之前是否有办法：**
  - 有，但是大部分方法粗暴且效率低下，对网络影响大，甚至会导致网络中断。
  - 根据在生产环境的统计，最坏情况甚至要试错几分钟，后续导致几十分钟网络波动。
- **解决问题的新方案：**
  - 本文提出P-ECMP（Programmability ECMP）
  - 通过挖掘并利用原本ECMP协议中未被充分利用的部分（ECMP组）实现PTC
  - 在保留ECMP的同时灵活的实现可编程的流量控制
  - 理论上可以兼容任何网络，任何交换机，额外开销少

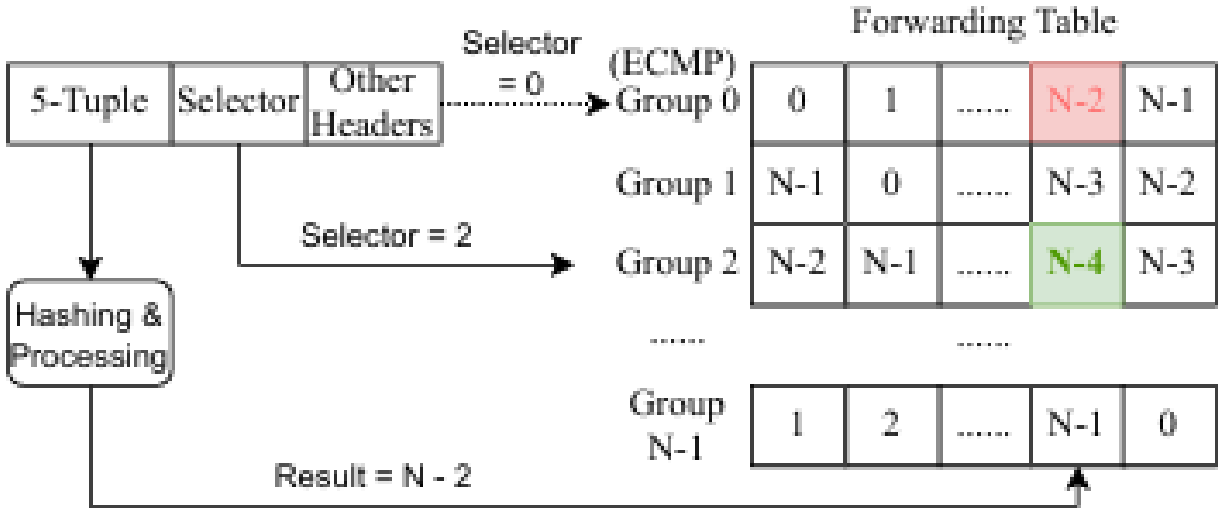
研究背景

- **等价多路径（Equal-Cost Multi-Path，ECMP）**
  - 是支撑当今几乎所有大型数据中心网络的核心技术
  - 核心思想：当数据包需要从源点到达某个目的地时，如果存在多条成本完全相同的传输路径，ECMP 将这些数据包分散到所有可用的等价路径上进行传输，而不是只选择其中一条。
  - 基本原理：流哈希



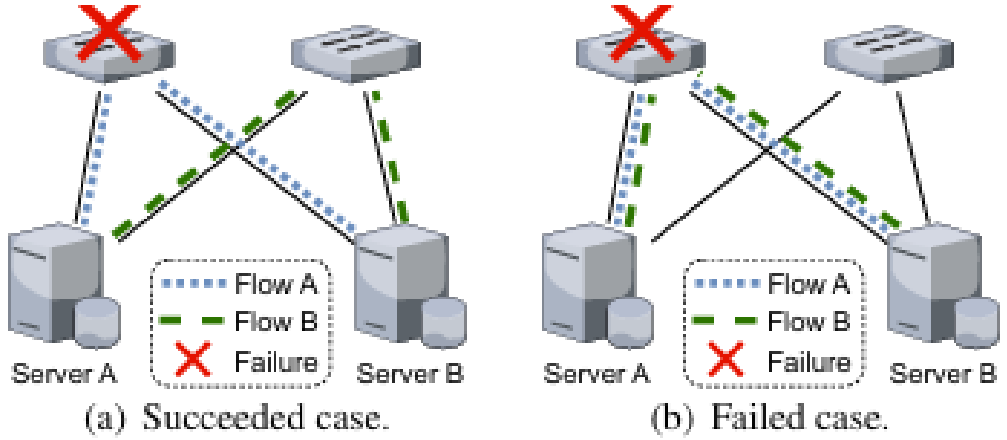
研究背景

- **ECMP组（ECMP groups）**
  - ECMP组是现代商用交换机中ECMP机制的一个标准，且早已存在
  - 它允许网络管理员将多个下一跳（next-hops）组织成不同的组，每个组对应一个ECMP组ID。
  - 路由查找时，先确定使用哪个ECMP组，再对该组内的成员进行哈希选择出端口
  - ECMP组最初的设计目标主要是为了解决路由聚合与灵活转发
  - 其设计初衷并非用于精确流量控制或可编程性



研究背景

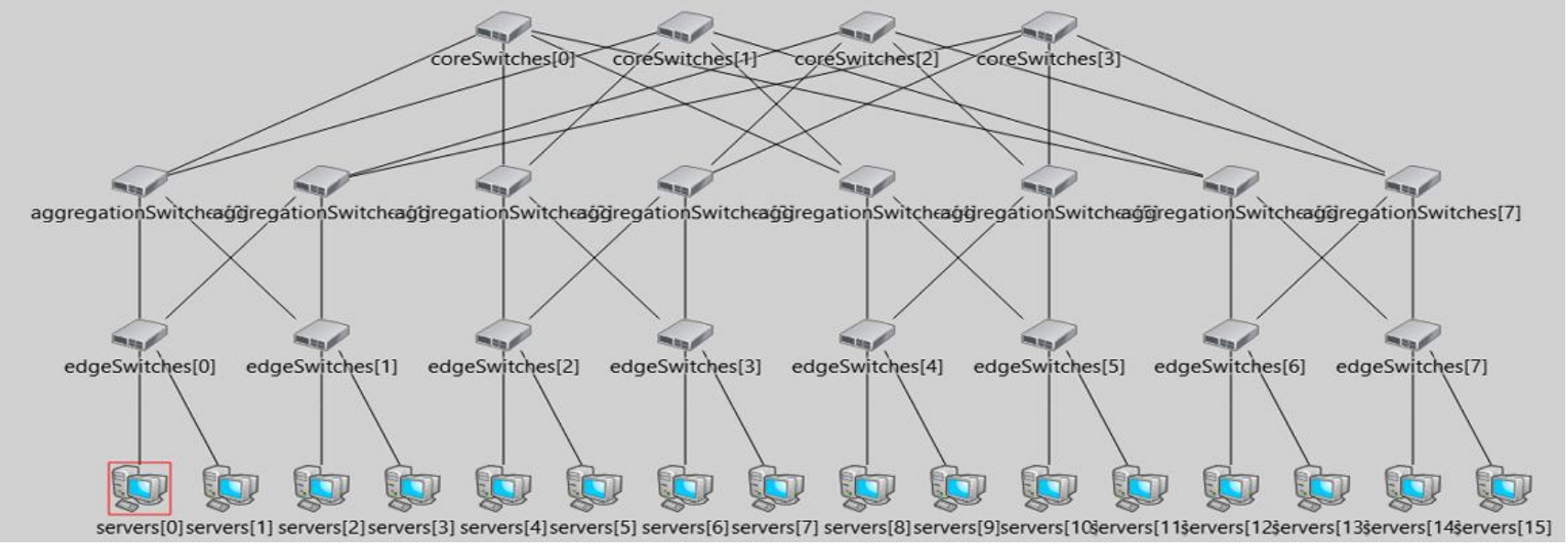
- **精确流量控制（Precise Traffic Control, PTC）**
  - 是一种网络流量管理技术，其核心目标是实现对数据流或数据包路径的细粒度、可预测、可编程的主动控制。
  - 关键特征：确定性，细粒度，主动性，可编程性
  - 典型应用场景：
    - 规避网络故障与实现快速故障恢复
    - 满足特定业务的性能与延迟要求
    - ……
  - 在ECMP基础上的实现方式
    - 路径偏移的精确流量控制（PTC of Offset）
    - 精确控制确切的下一跳（PTC of Hop）



研究背景

- **Fattree和Clos**

Fattree



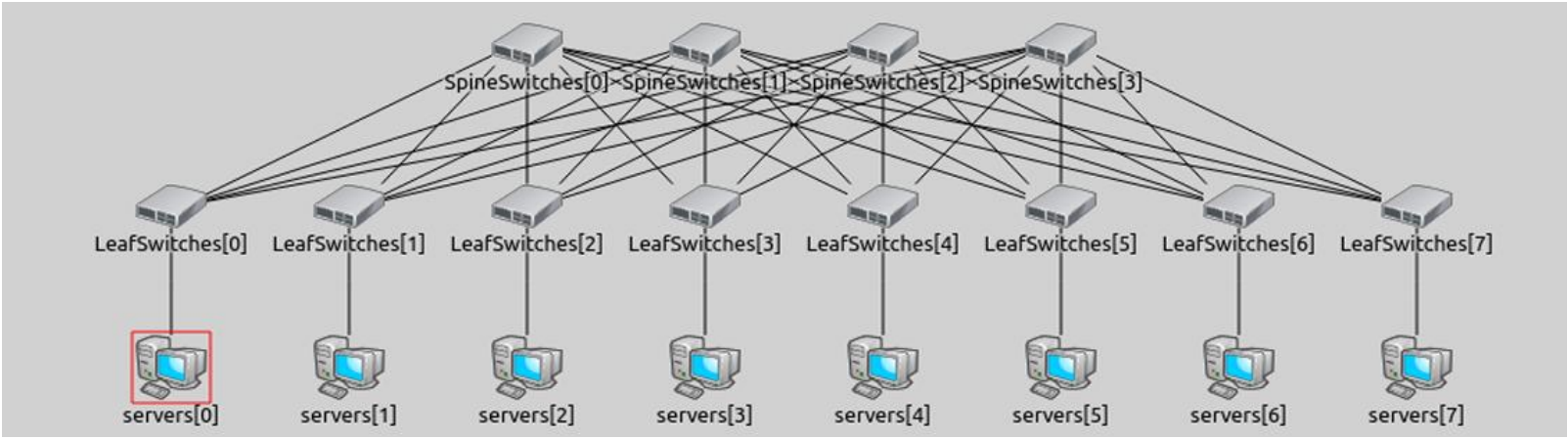
核心层

汇聚层

边缘层

终端

clos



spine层

leaf层

终端

设计实现

- **关键事实：**
  - 无论是网络发生故障（如超时、重传、性能下降），或是应用内部对通信服务质量的要求发生改变，终端总是能比控制平面更早感知到。
  - 由于数据中心网络拓扑大多有一定的对称性，其ECMP组缓存的利用率普遍小于1%
- **核心想法（key idea）：**
  - 将一部分控制权下放给终端
  - 重新利用ECMP组这一几乎没有被使用的资源
  - 通过合理的系统设计来使ECMP兼容PTC，并通过终端控制PTC
- **主要挑战：**
  - 在何处对ECMP进行编程？如何编程？
  - 顶层如何实现来保证实际部署时的可用性？
  - 如何在部署过程中保证一致性？

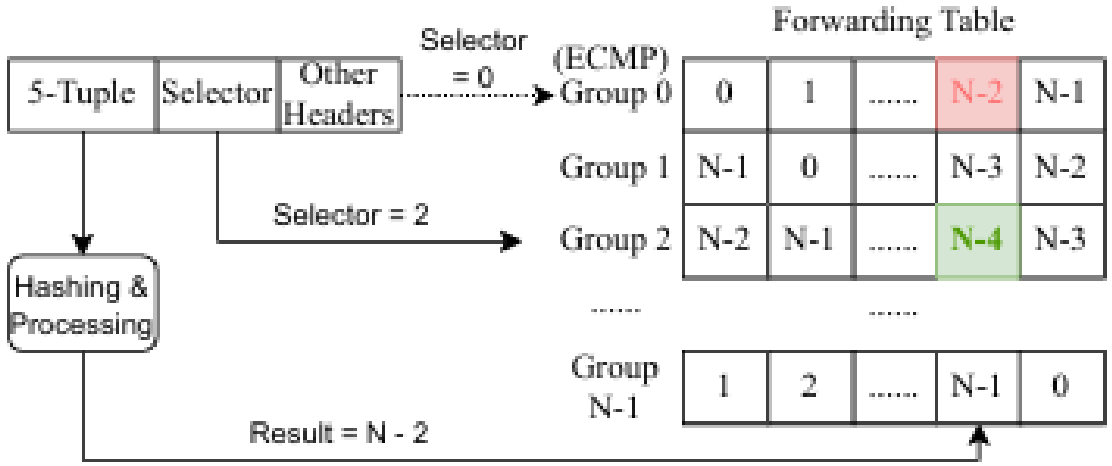
设计实现

- **应该在何处对ECMP进行编程?**
  - ECMP一般分三个阶段：输入，哈希，映射
  - 在输入阶段进行修改已有多项研究，然而，由于交换机哈希函数是黑盒，实用性欠佳
  - 更改哈希算法更不现实，几乎没有厂商的交换机支持更改算法
  - 修改映射又可以分为两种：重定向到同组内不同成员，或直接重定向到不同组
  - 第一种方法存在兼容性问题，因为许多交换机不支持对后处理结果执行运算
  - 综上，只能修改组

Stage		PTC of Offset	PTC of Hop	Compati-bility	Practi-cality
Mapping	Input	Yes*	Yes*	Yes	No
	Hashing	No	No	No	No
	Member Redirect	No	Yes	No	Yes
	Group Redirect	**Yes**	**Yes**	**Yes**	**Yes**

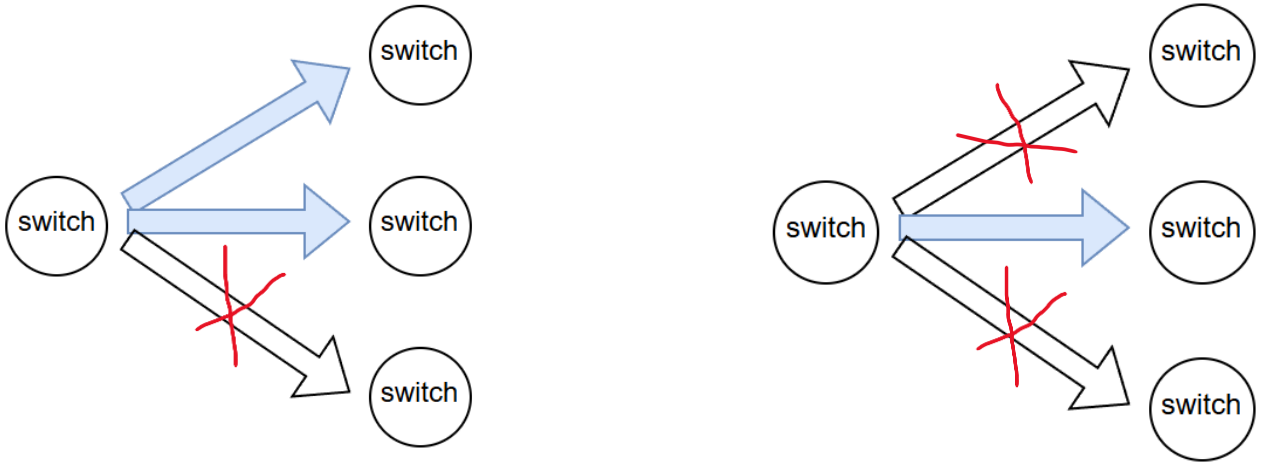
设计实现

- **PTC 的具体实现---ECMP组矩阵**
  - 令 l = [p_0, p_1, ⋯ , p_{N-1}] 表示一个 ECMP 组，其中每个 pi（i ∈ [0, N−1]）对应一个输出端口，N 是可用端口的总数。
  - P-ECMP 通过使用多个（M 个）ECMP 组替代单一 ECMP 组，将转发表扩展为一个 ECMP 矩阵，称为控制矩阵 C，其维度为 M×N
  - 我们假设用户可以通过在数据包头部嵌入一个选择器 s 来选择特定的 ECMP 组，即 ls
  - 则，在如右图所示的，通过循环左移得到的矩阵中，若S为0，则默认ECMP模式，否则S即为偏移量
  - 固定下一跳只需要继续添加只有一个元素的组即可。



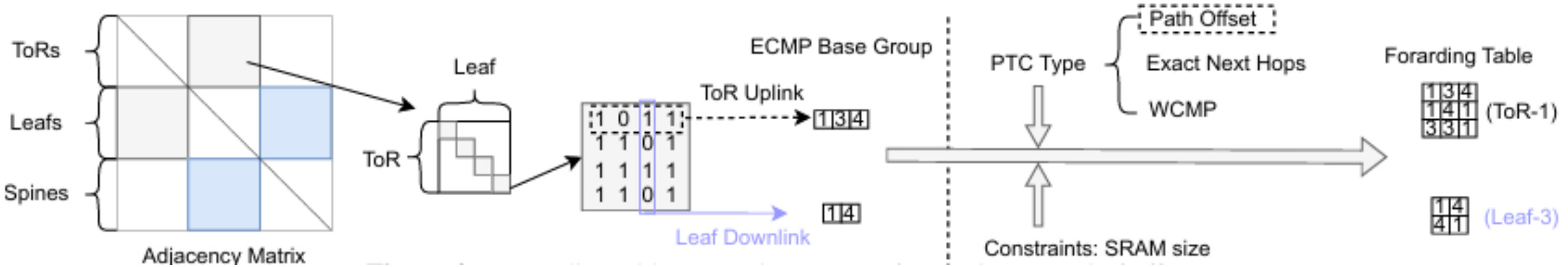
设计实现

- **为什么要有两种不同的PTC实现方式?**
  - 这两者的实现目的不同
    - 路径偏移的精确流量控制（PTC of Offset）主要是为了在规避故障的同时保证负载均衡
    - 精确控制确切的下一跳（PTC of Hop）主要是为了实现精确的流控制
  - 这两者的资源消耗不同
    - 路径偏移的精确流量控制（PTC of Offset）消耗报文头较小（大约6位）
    - 精确控制确切的下一跳（PTC of Hop）消耗报文头较多（十几位）



设计实现

- **编译实现（顶层实现）**
  - 为了部署前述的PTC，运维人员需要将控制矩阵作为转发表下发到每一台交换机。手动完成这一任务几乎不可能。为此，需要通过一个编译器来自动化这一过程。
  - 该编译器接收以下输入：PTC 类型、网络拓扑结构以及异构交换机的 SRAM 资源约束。编译器的输出则是网络中所有交换机的转发表。（如下图）



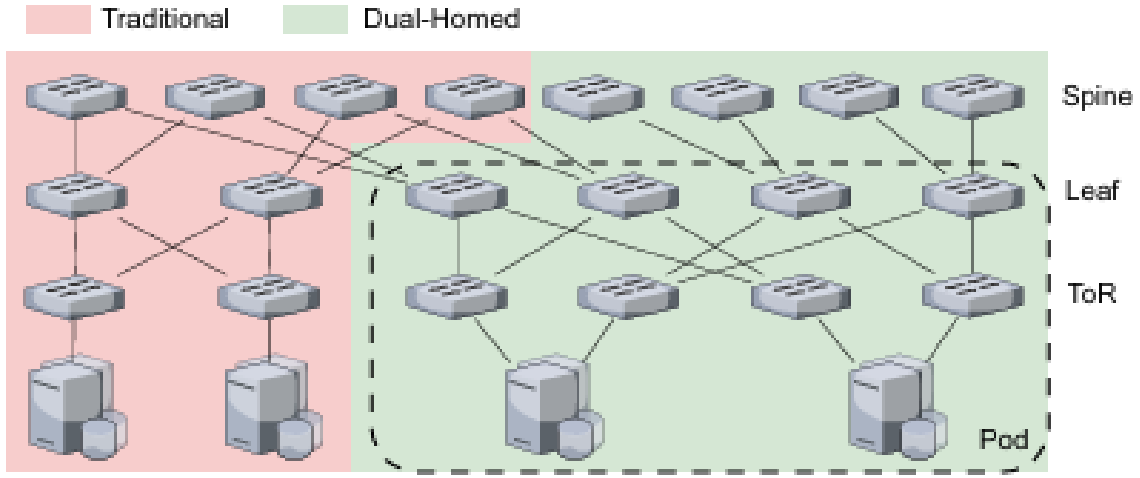
设计实现

- **更新机制**
  - 动态网络更新挑战
    - 动态拓扑变化需更新转发表，但是短暂不一致导致丢包或异常转发。
  - 运行时更新机制
    - 版本化转发表：每版本对应特定选择器范围，SRAM空间划为两半，对应两个版本
    - 创建新版本：更新时生成新版本，保持当前表不变，新表可用后平滑切换。
    - 切换机制：选择器值加⌊M/2⌋实现版本切换
  - 应用场景和机制
    - 静态拓扑变化：网络扩容、重构、长期故障或维护
    - 瞬态故障：无需更新，自行恢复
    - 编译与下发：系统几毫秒内完成配置编译并下发

实验测试

- **实验设置**
  - 整体采用spine-leaf拓扑，一种特殊的clos拓扑
  - 除了资源分析外，网络功能测试和生产环境测试均在#12上进行
  - 采用了一种“多宿主服务器架构”，在测试中表现为双宿主服务器绑定（dual-homed server bonding）

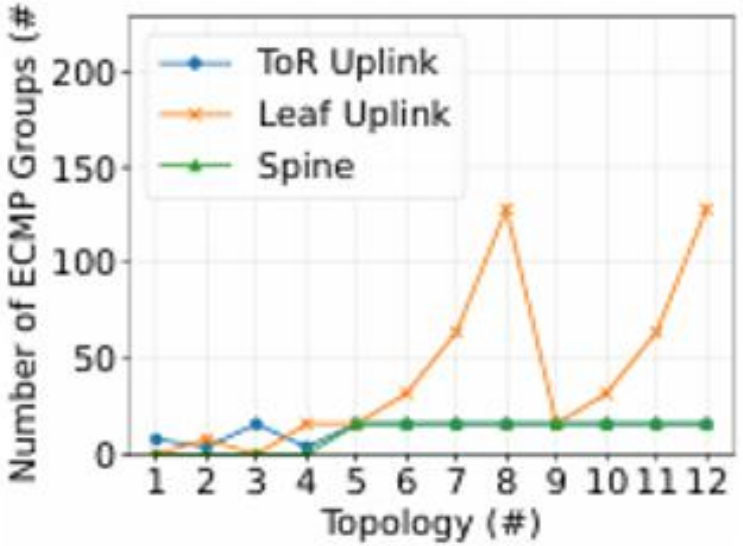
Topo	Dual-Homed	# Leaf Per Pod	# Spine	OSR	Max # Paths Between Servers
#1	N	4	N/A	N/A	4
#2	Y	4	N/A	N/A	8
#3	N	8	N/A	N/A	8
#4	Y	8	N/A	N/A	16
#5	N	8	8×8	8:1	64
#6	N	8	16×8	4:1	128
#7	N	8	32×8	2:1	256
#8	N	8	64×8	1:1	512
#9	Y	8	8×8	8:1	128
#10	Y	8	16×8	4:1	256
#11	Y	8	32×8	2:1	512
#12	Y	8	64×8	1:1	1,024



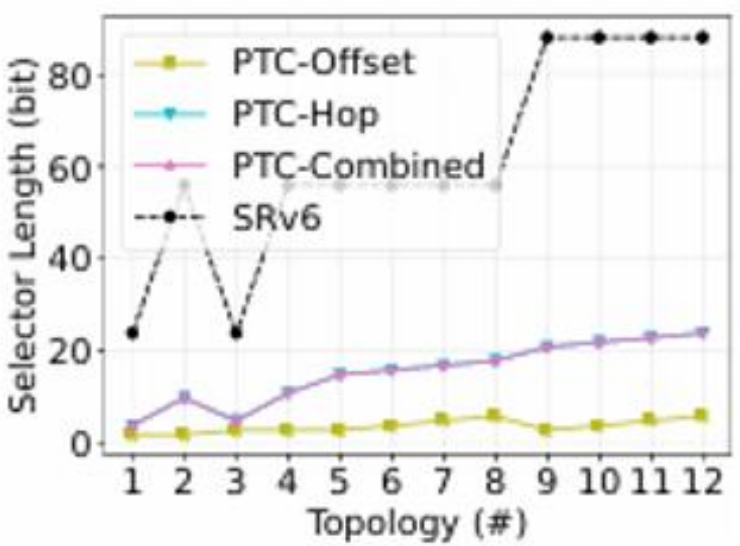
$$\mathrm{OSR} = \frac{\text{连接到服务器/终端设备的下行端口总带宽}}{\text{连接到网络上层（如上联交换机）的上行端口总带宽}}$$

实验测试

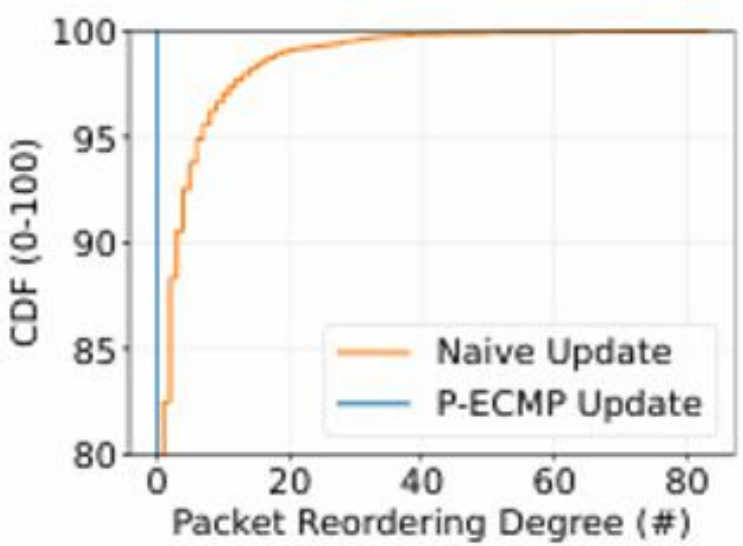
- **资源需求分析**
  - 图（a）:在不同拓扑层上，交换机为支持两种 PTC 功能所需的 ECMP 组数量
  - 图（b）:选择器字段所占比特位分析
  - 图（c）:事务性机制实用性分析



(a) Number of ECMP groups.



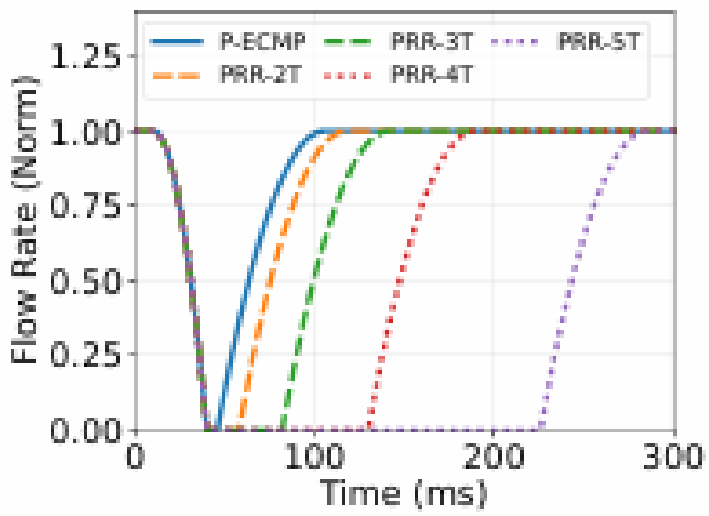
(b) Bit length of selector.



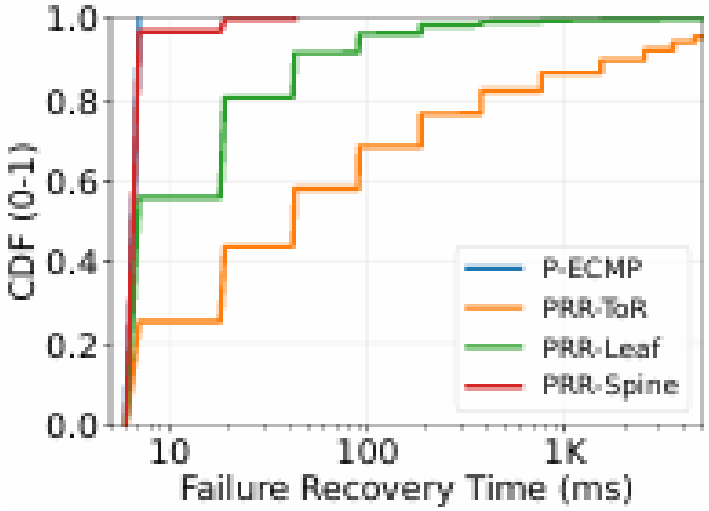
(c) Packet reordering degree.

实验测试

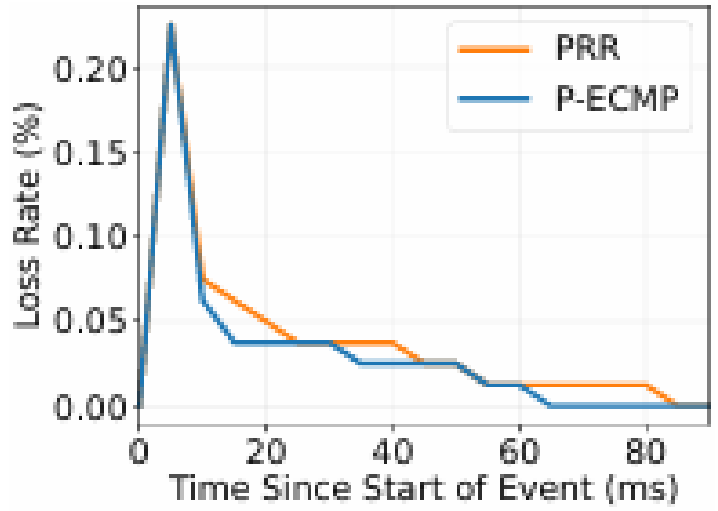
- **网络功能测试**
  - 图a：网络故障切换测试图
  - 图b：10000次网络故障切换测试后，累计概率图（竖轴为累计概率）
  - 图c：故障后丢包率随时间变化图



(a) Single-flow recover example.



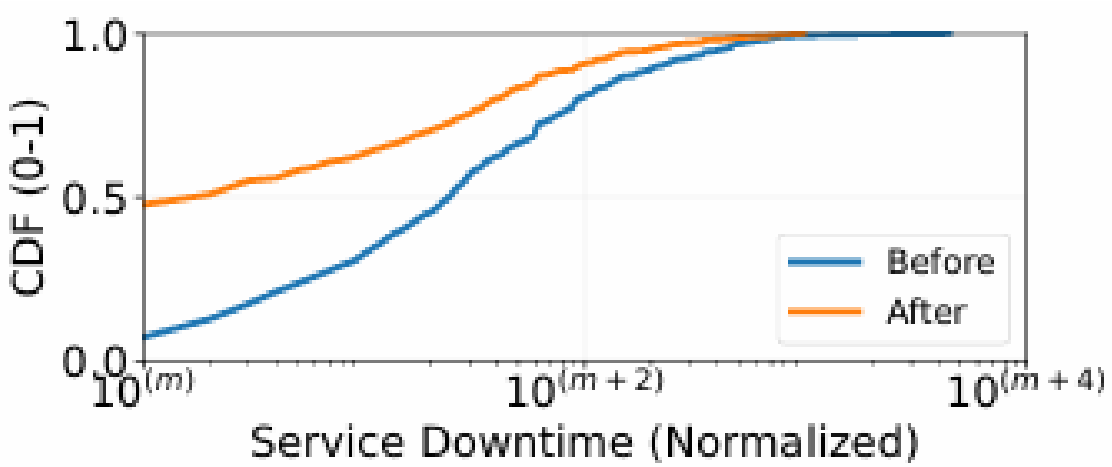
(b) Single-flow recovery duration.



(c) Failover from a link failure event.

实验测试

- **生产环境部署**
  - 已经在多个数据中心逐步部署了 P-ECMP 的“路径偏移” PTC 功能。整个部署过程平滑无缝
  - 图：部署 P-ECMP 前后，由网络故障引起的服务中断时间分布
  - 结论：P-ECMP 在生产环境中表现高效且符合预期



后续思考

- **这篇 paper 的工作能否进一步提高?**
  - 从被动故障恢复到主动流量控制
  - 摒弃被动的事后补救，在拥塞发生前预测其发生并把流重定向避免拥塞。
  - 是否可以训练一个AI模型，根据历史流量和当前状态，预测最优的PTC配置?
- **这篇 paper 能不能泛化?**
  - 核心思想：在保持现有基础设施稳定性的前提下，增加一层“可编程的、精确的”控制能力
  - 这个模式可以泛化到任何能强大但不够灵活的传统网络（设备）上

Q&A

汇报人：陆俊安

2025.9.29