中国科学技术大学
University of Science and Technology of China

B4: Experience with a Globally-Deployed Software Defined WAN

Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, et al.

ACM SIGCOMM, 2013

授课教师：赵功名

中国科大计算机学院

2025年秋·高级计算机网络

Outline

I. Introduction

II. Background

III. Design

IV. Traffic Engineering

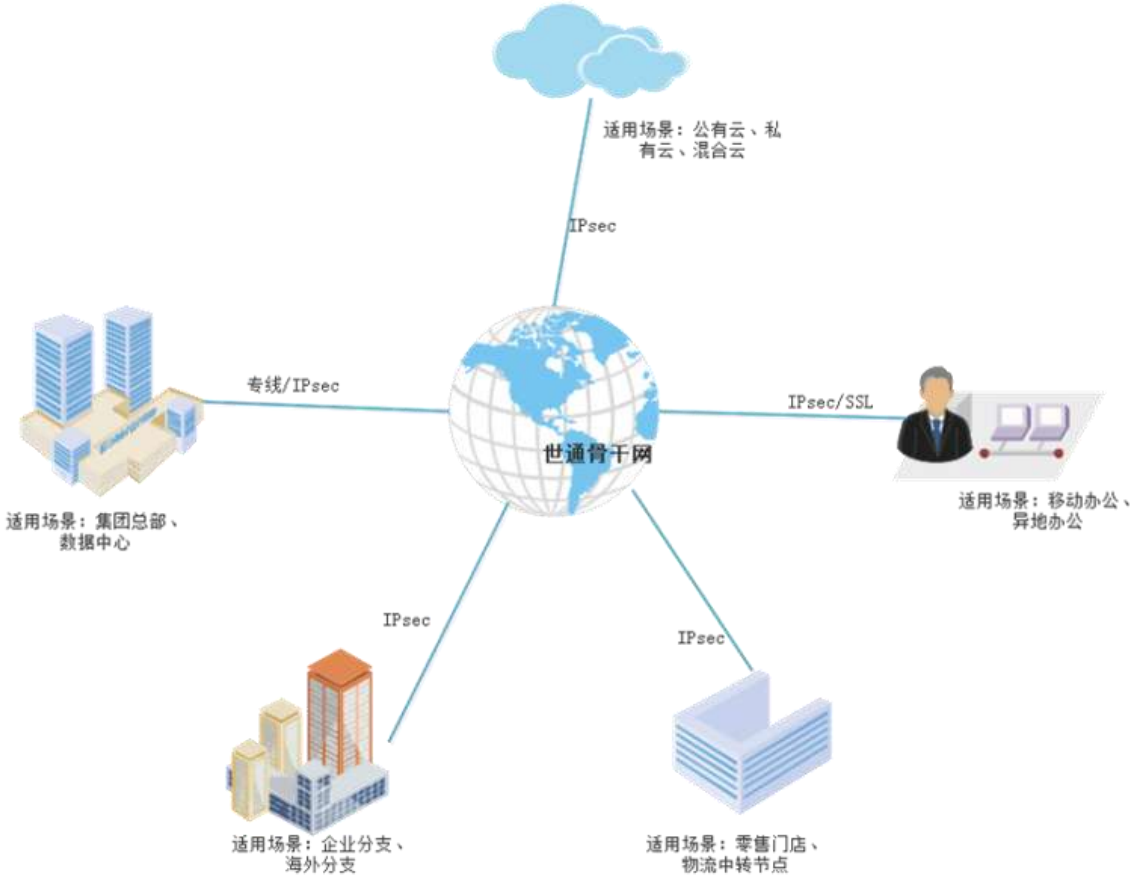
V. TE Protocol and OpenFlow

VI. Evaluation and Experience

VII. Review

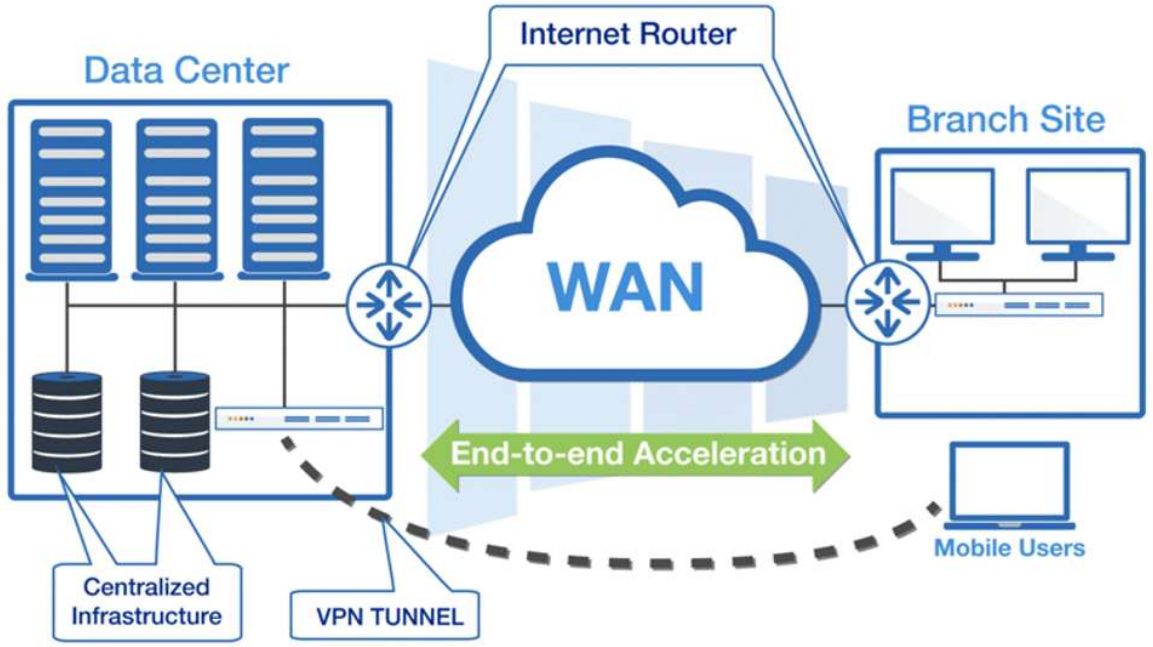
The Importance and Limitations of WAN

- 现代广域网（WAN）是互联网性能与可靠性的核心基础，承担着数以千计链路上的数Tbps流量
- 广域网通常对所有位进行相同的处理
- 这种策略在链路故障时平等对待所有应用程序，无法区分不同应用对带宽的敏感度



The Cost of Traditional Over-provisioning

- 由于链路昂贵且丢包不可接受，广域网路由器由高端的专用设备组成
- 为掩盖故障带来的影响，传统WAN通常将链路平均利用率控制在30–40%
- 过度预留带来了较好的可靠性，但代价是2–3倍的带宽冗余和昂贵的专用硬件，从而造成高昂的建设与运营成本



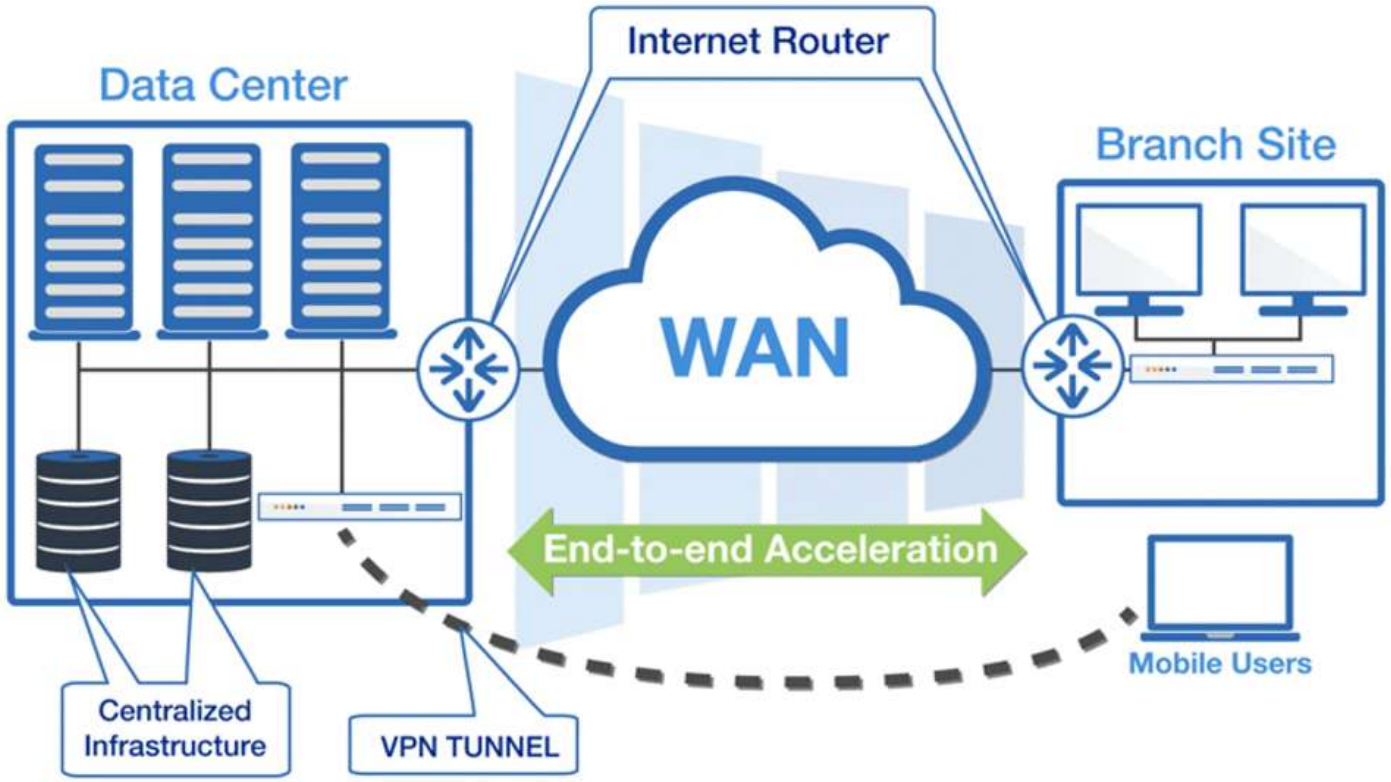
Bandwidth Challenges Faced by Google

- Google全球数据中心互联WAN的特殊特征：
  - 数据中心间通信带宽需求巨大
  - 传输任务高度弹性且可控
  - 公司同时掌控应用、服务器与边缘网络，使得可在网络边缘实施速率控制与需求测量

- 这些特性使得采用集中式、可编程的软件定义网络成为可能

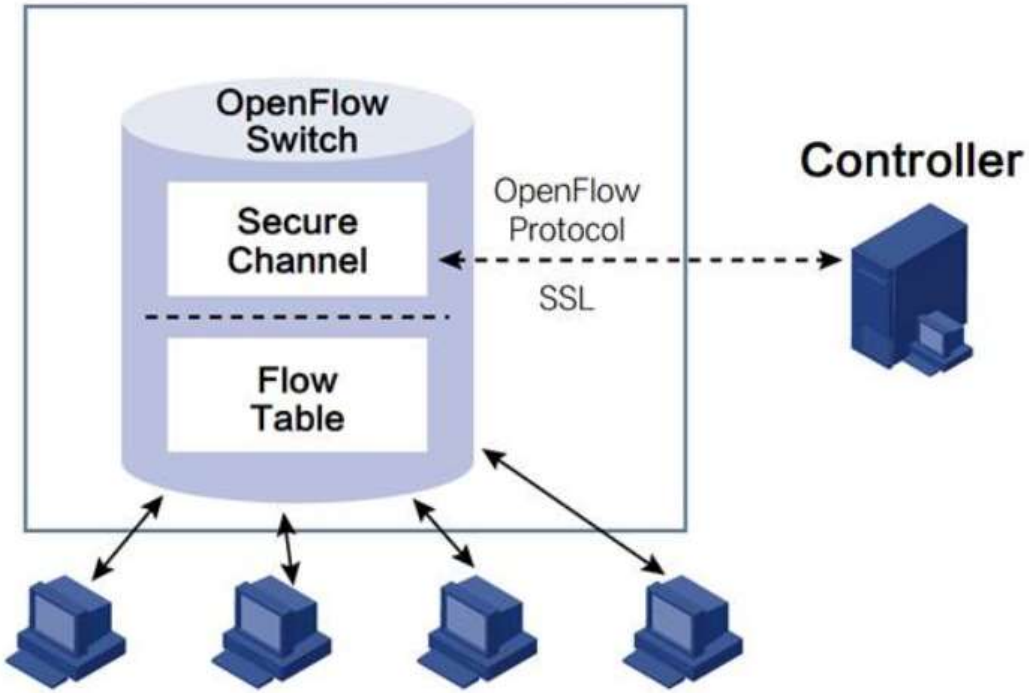
The Design Principles of Adopting SDN

- 接受故障的不可避免性，并将其显式暴露给应用层
- 使用可由中心控制的简单交换机硬件，通过软件编程实现灵活的转发逻辑
- 这样能在服务器上快速迭代新协议、调度与监控机制，从而提高整体网络效率



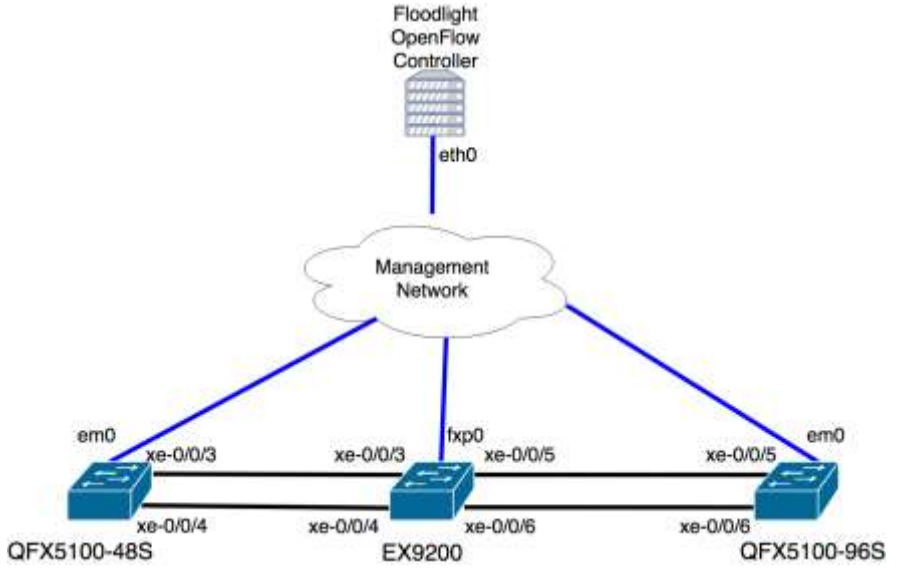
Overall Goals and Achievements of the B4 System

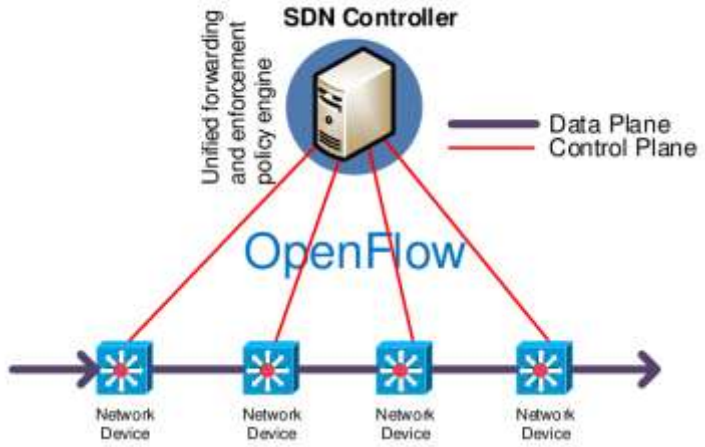
- B4系统是Google的私有数据中心WAN
- 基于OpenFlow实现集中式流量工程（Traffic Engineering, TE）
- B4的TE服务：
  - 在资源受限时依据应用优先级调度带宽
  - 使用多路径转发以充分利用容量
  - 在故障与需求变化时动态重新分配带宽。许多链接能在接近100%利用率下运行，平均利用率达70%，相比已有方案提升2-3倍



Deployment Scale and Research Value

- B4已部署超过三年，传输流量超过Google面向公众的WAN，是迄今最大规模的SDN/OpenFlow系统之一
- B4实现了高效的带宽利用、快速的控制功能迭代与应用层的动态适应性
- SDN并非万能——系统也经历过一次重大故障，揭示了大规模网络管理与SDN设计的挑战，为未来研究提供了经验启示





Outline

I. Introduction

II. Background

III. Design

IV. Traffic Engineering

V. TE Protocol and OpenFlow

VI. Evaluation and Experience

VII. Review

Overview of Google WAN

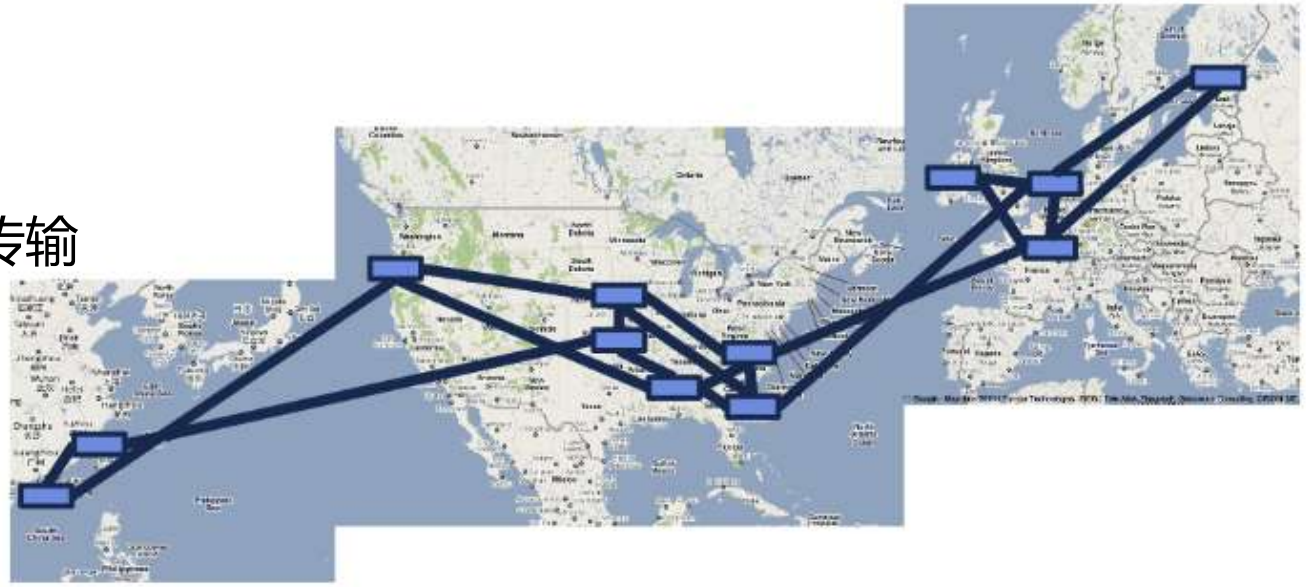
- Google的全球广域网是互联网中规模最大的之一
- 支撑搜索、视频、云计算和企业级应用等服务
- 架构由分布于全球的数据中心和边缘缓存节点组成，用于为全球用户提供低延迟、高可靠性的服务交付

Two WANs of Google

- Google运营两张彼此独立的广域网：
  - 用户面网络（User-facing WAN）
    - 负责与外部互联网互联，承载用户请求与内容分发
    - 参考链接：http://peering.google.com
  - 内部网络B4

Two WANs of Google

- Google运营两套独立的WAN系统：
  - 用户面网络（User-facing WAN）
  - 内部网络B4
    - 用于连接数据中心之间
    - 支撑异步数据复制
    - 索引推送及冗余存储
    - 约90%以上的内部流量都通过B4传输
- 这种双网络分离设计是为了应对不同的连接密度、协议要求和可用性目标



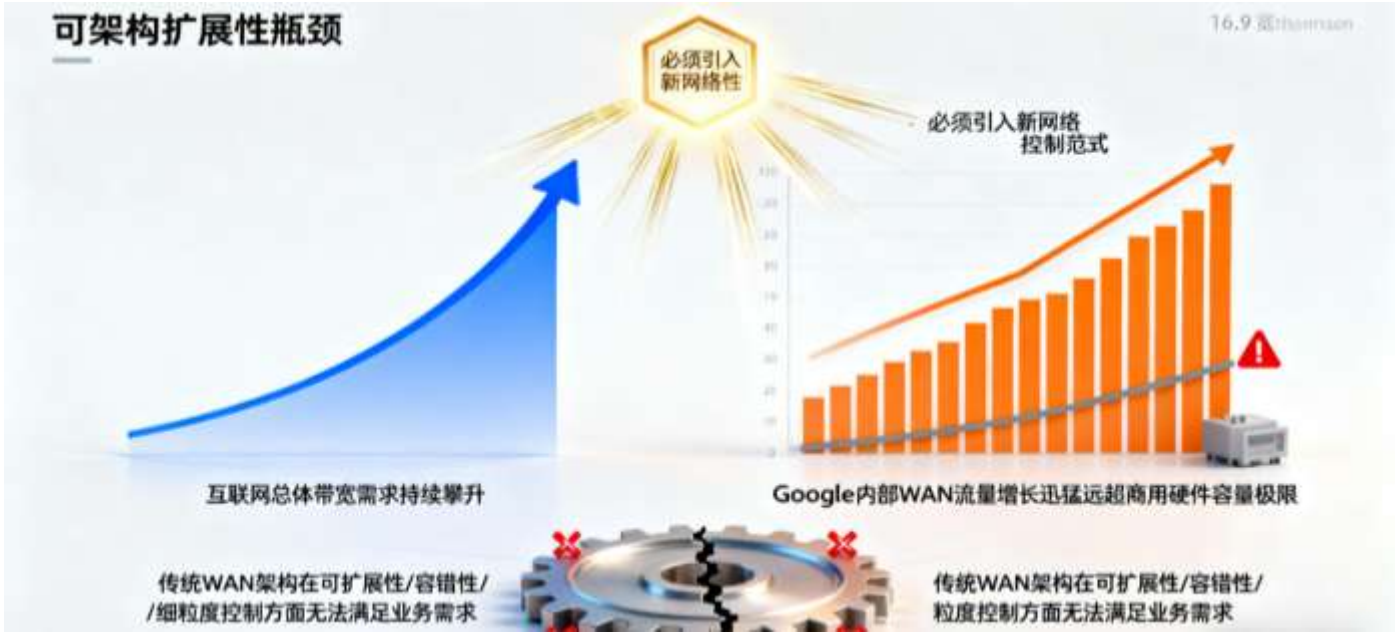
Service Flows Carried by B4

- B4主要支撑三种类型的应用流量：
  - 用户数据副本（如邮件、文档、视频）跨数据中心复制以确保可用性
  - 分布式存储访问，用于跨站点数据计算
  - 大规模状态同步操作
    - 例如，搜索索引同步（Google 通过将其更新后的索引数据批量推送到全球各个数据中心，来保持搜索服务的同步与一致性）

- 三类流量依次表现为：
  - 体量递增
  - 时延敏感度递减
  - 优先级递减
- 即高优先级业务通常流量小但时延敏感，而低优先级业务追求高带宽

Scalability Bottlenecks of Existing Architecture

- 互联网总体带宽需求持续攀升，Google的内部WAN流量增长迅猛，远超商用硬件的容量极限
- 传统WAN架构在可扩展性、容错性与细粒度控制方面无法满足Google的业务需求
- 必须引入新的网络控制范式



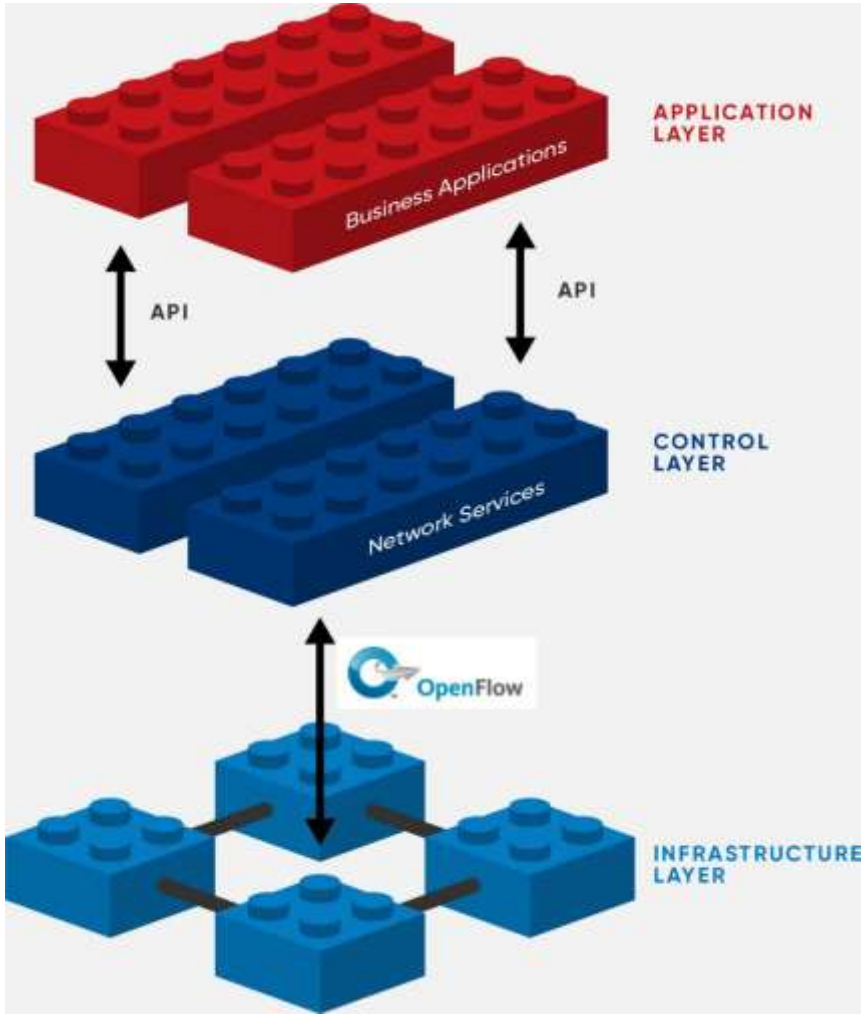
B4 Characteristics

- B4的架构设计基于四个核心观察
  - 弹性带宽需求：跨数据中心数据同步任务可自适应带宽变化
  - 站点数量有限：总规模仅几十个数据中心，便于集中控制
  - 端到端应用可控：可在边缘强制流量整形与优先级管理
  - 成本敏感性强：传统30–40%利用率的链路预留及昂贵端口成本不可持续

- SDN通过转发与控制分离，实现集中式控制，可以较好适配B4的网络流量特征

The Motivation for Adopting SDN

- SDN使Google得以建立独立的软件控制平面，优点：
  - 统一的全网可视化与协调
  - 控制平面可独立于硬件演化
  - 快速迭代与部署新协议
  - 模拟与仿真环境简化测试
- 附加优势：
  - 快速迭代新协议
  - 简化测试环境
  - 通过模拟确定性中心TE服务器的异步路由行为，改进容量规划
  - 通过以fabric为中心的WAN视图简化管理



Outline

I. Introduction

II. Background

III. Design

IV. Traffic Engineering

V. TE Protocol and OpenFlow

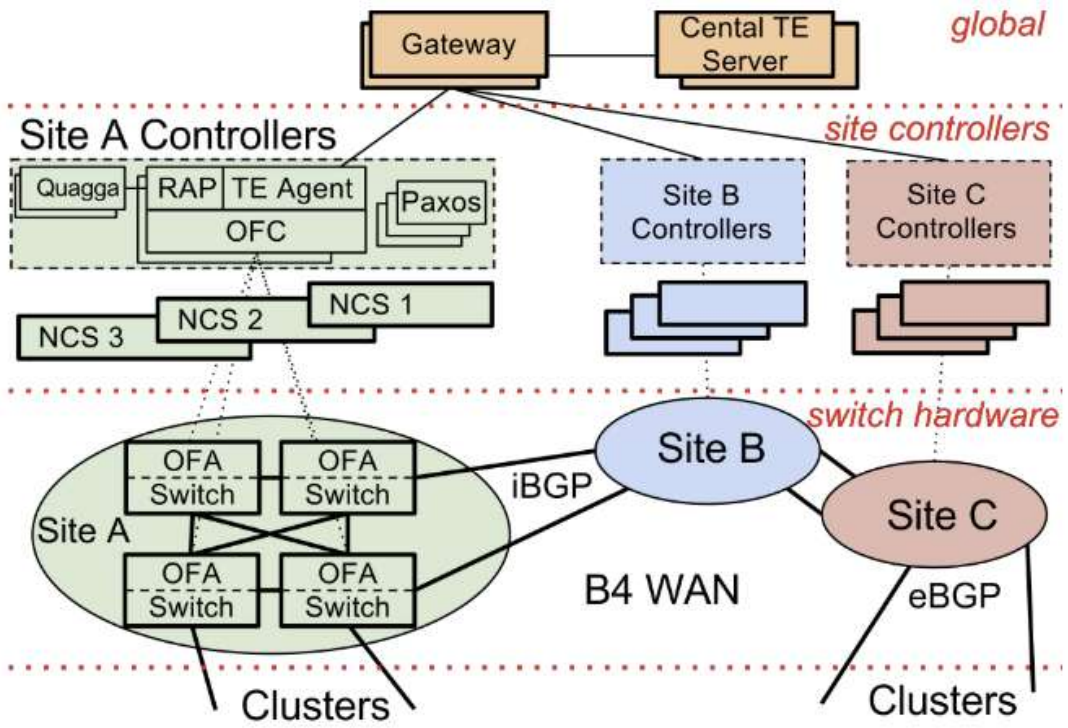
VI. Evaluation and Experience

VII. Review

Overview

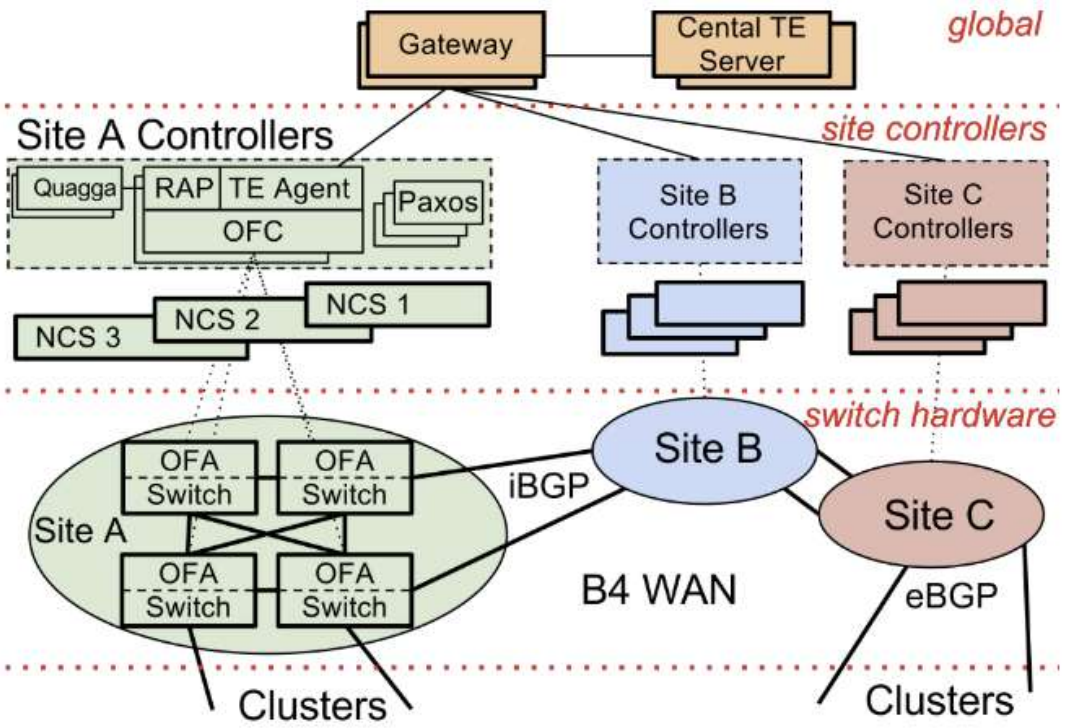
- B4的SDN架构逻辑上划分为三层：
  - 交换硬件层
  - 站点控制层
  - 全局层

- 交换机硬件层：包含支持 OpenFlow 的交换机



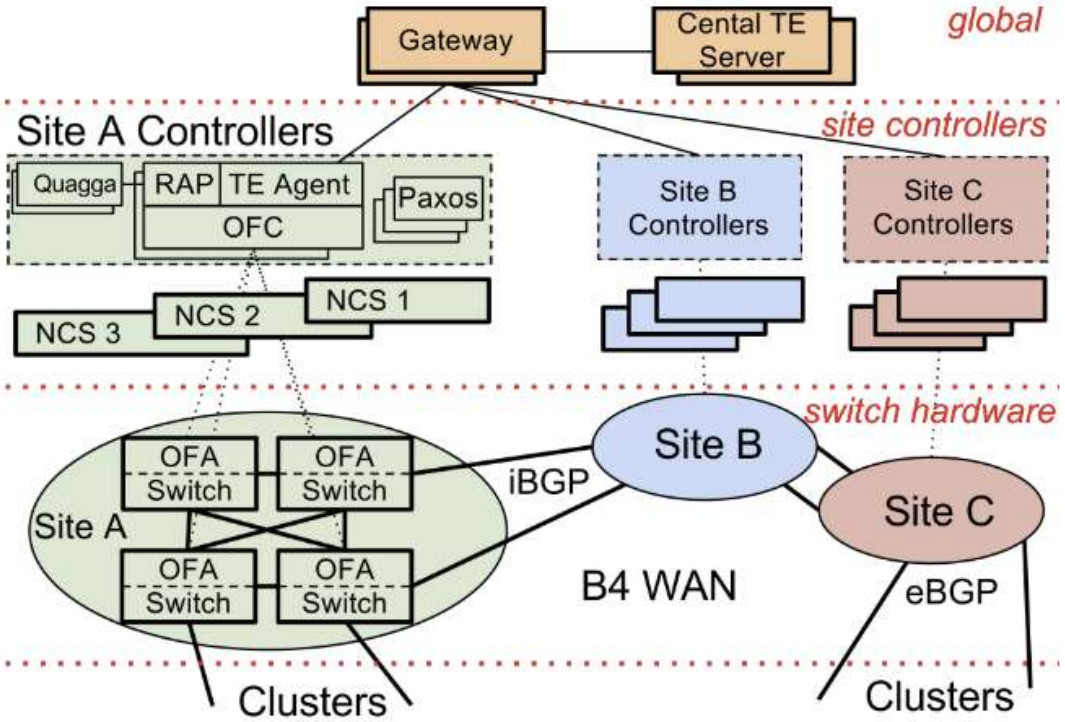
Overview

- 站点控制器层：由网络控制服务器（NCS）组成，同时托管 OpenFlow 控制器（OFC）和网络控制应用（NCA）
  - 示例：站点 A 控制器包含 NCS1、NCS2、NCS3，以及 Quagga（路由协议软件）、RAP（路由应用代理）、TE Agent（流量工程代理）



Overview

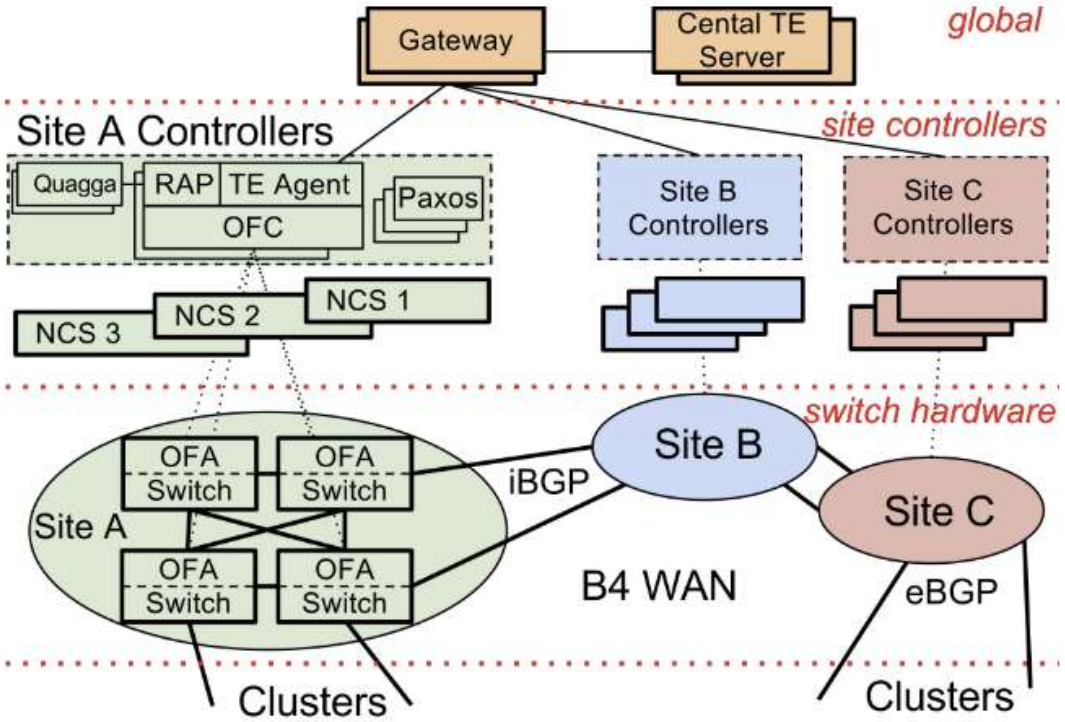
- 全局层（Global Layer）
  - 核心组件：SDN 网关（SDN Gateway）和中央流量工程服务器（Central TE Server）
- TE通过Gateway与各站点控制层交互，实现全局带宽调度与路由优化
- 为保证鲁棒性，所有全局应用均在多个站点冗余部署并采用独立的主选举机制



Overview

兼容传统协议的设计考量

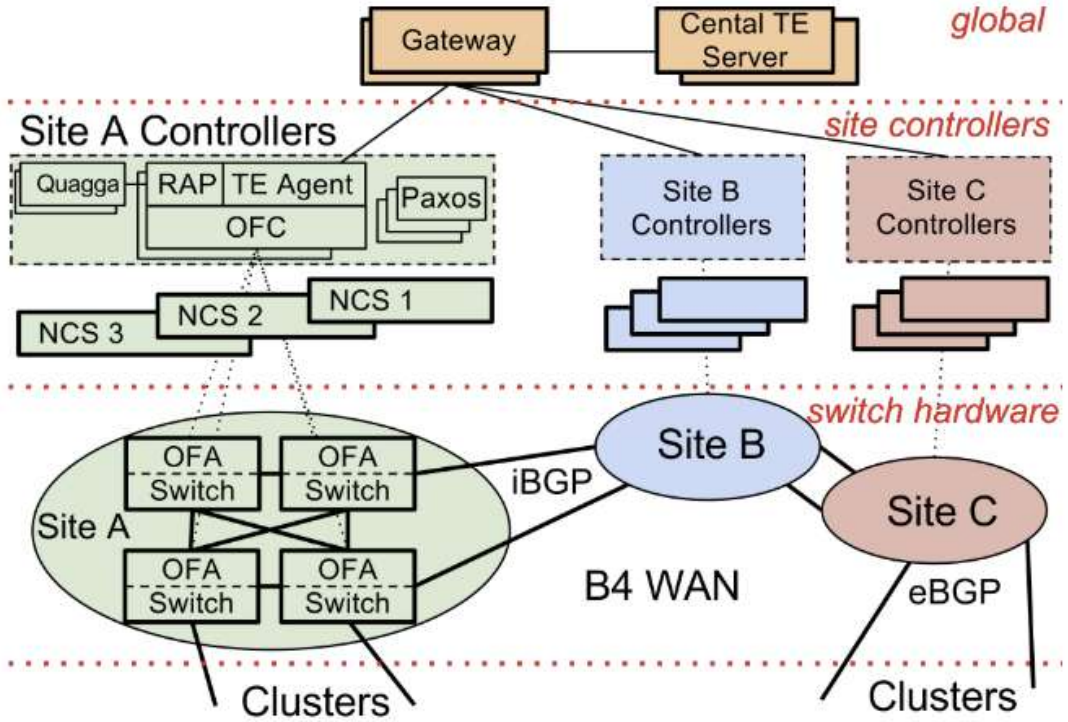
- B4的各数据中心集群逻辑上构成独立的自治系统（AS），通过BGP与B4交换机互联
- 为保证SDN平滑过渡，B4最初运行传统BGP/ISIS协议，并逐步叠加SDN与集中式TE
- 该策略既保留了对外兼容性，又降低了部署风险



Overview

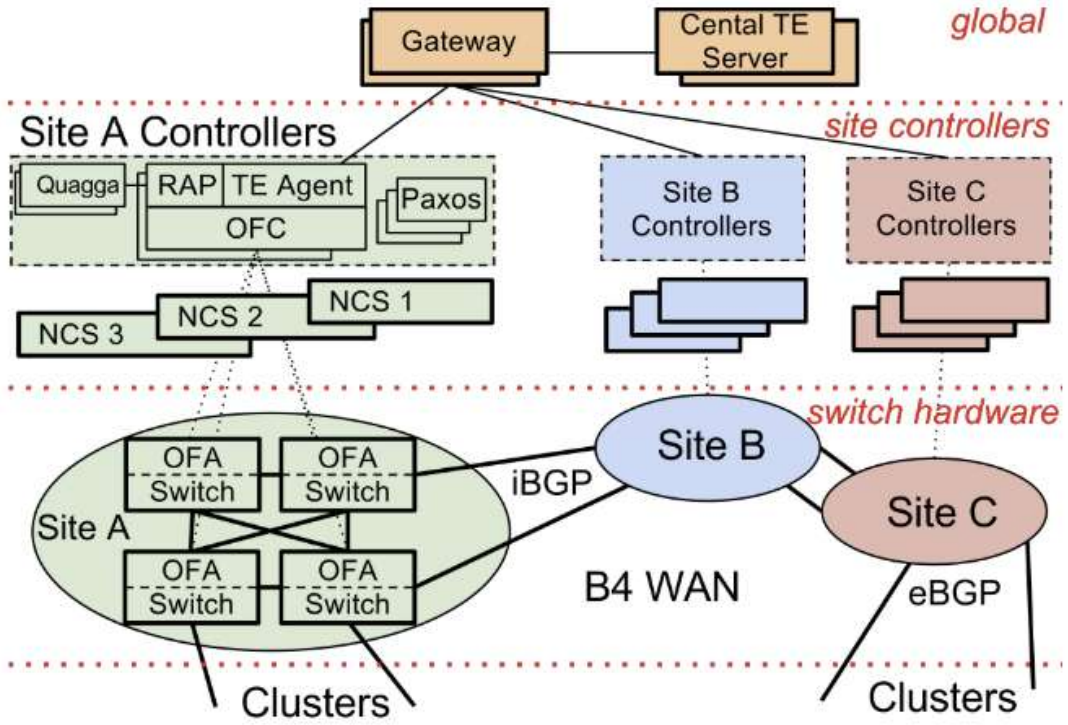
路由与TE服务的分层实现

- 在功能集成上，研究团队选择将传统路由与集中式流量工程（TE）分离实现
- 初期仅部署标准路由服务，随后以TE叠加方式逐步上线
- 此策略使得系统具备“回退机制”，可在TE出现异常时快速切换回最短路径转发，极大提升了系统可维护性



Overview

- 站点抽象：流量工程（TE）将每个 B4 站点抽象为 “单个节点”，站点间的连接抽象为 “单条指定容量的链路”
- 流量分发规则：跨站点链路的所有流量必须在该链路的所有组成物理链路间均匀分配
- 负载均衡实现：B4 采用自定义版本的等价多路径（ECMP）哈希算法，确保流量均匀分布

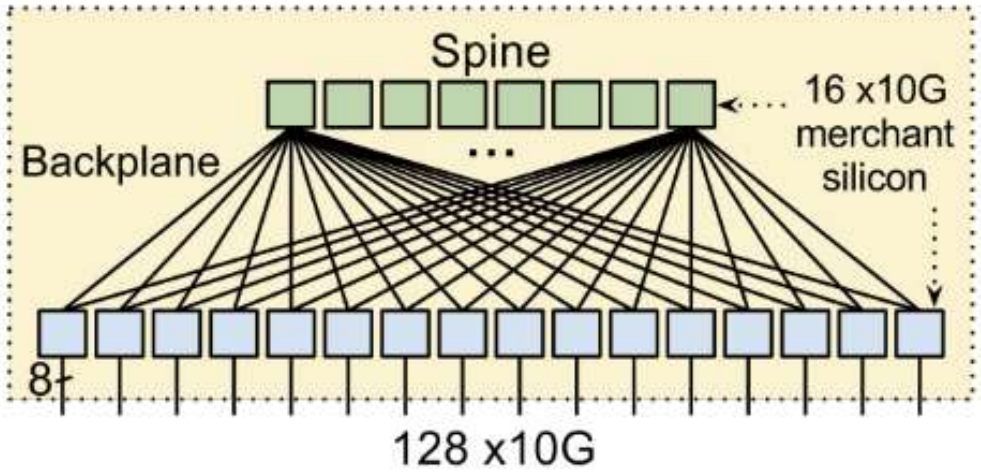


Switch Design

- 传统广域路由设备具有深缓冲与复杂高可用设计
- B4设计理念
  - **精细化端点管理**（不需要复杂的交换机功能）：调整发送速率来避免深度缓存的需求，从而减少代价高昂的数据包丢弃。交换机部署在数量相对较少的数据中心之间，无需大型转发表
  - **极简化硬件功能**：交换机故障通常不是硬件问题。通过将大部分软件功能从硬件中剥离，借助分布式系统容错机制管理软件可靠性
  - **自行设计硬件**（实现集中式TE）：市面上没有任何现成平台能够支持SDN部署，即无法提供对交换机转发行为的底层控制接口

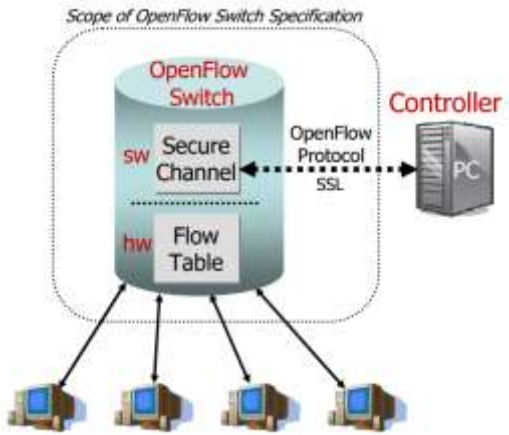
Switch Design

- 为突破单个交换芯片的容量限制，采用了基于商用硅芯片（merchant silicon）的多芯片两层Clos拓扑，通过铜背板互联
- 每台交换机提供128个10GbE端口，由24个商用16x10GbE芯片组成，实现高端口密度与低阻塞比
- 此设计兼顾可扩展性与成本效益，为高带宽跨数据中心通信提供基础



Switch Design

- B4交换机嵌入Linux系统，运行OpenFlow Agent（OFA）进程，用以桥接硬件驱动与远程控制器
- OFA扩展了OpenFlow协议以支持硬件流水线操作，将OpenFlow消息翻译为具体芯片指令，该代理层是B4实现软硬件解耦的关键
- OFA需在逻辑上抽象出“单一无阻塞交换机”，但底层实际由多个独立芯片构成，主要挑战在于：
  - 将OpenFlow的抽象转发表高效映射到具体芯片结构
  - 保持多个芯片的表项一致性与全局视图同步
- 这是B4硬件抽象层设计的核心技术之一



Network Control Functionality

网络控制功能与容错机制

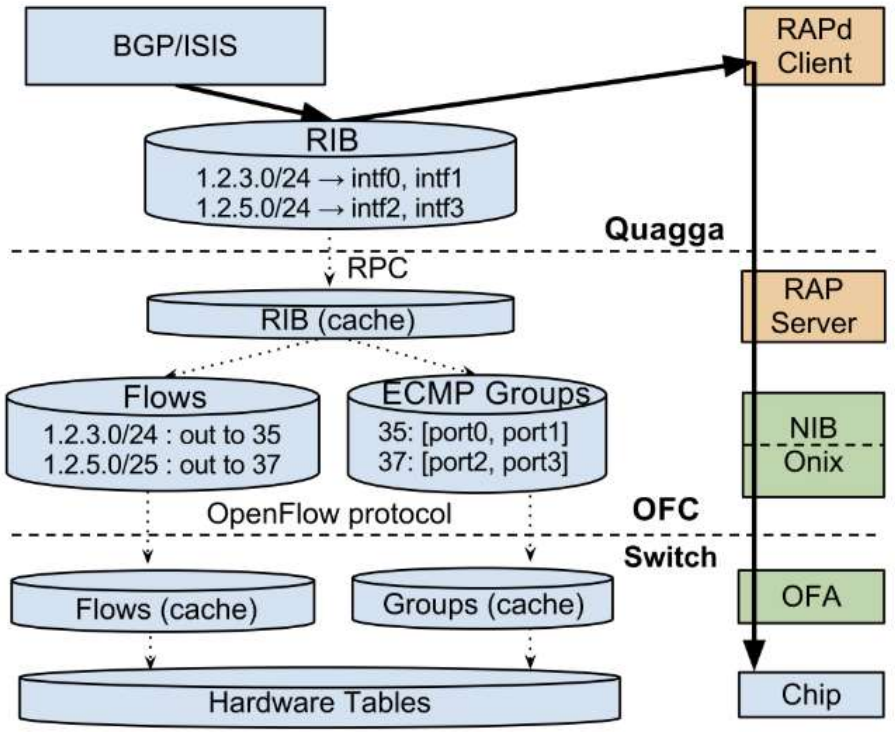
- 在控制层，NCS与交换机通过专用的控制网络通信
- 每站点运行多副本控制进程，通过Paxos算法进行主副本选举与故障检测
- 该机制确保任何控制进程故障都能自动切换主节点，从而实现高可用的分布式控制平面

控制器状态同步与Onix平台

- B4使用改进版Onix框架作为OpenFlow控制平台，核心组件为网络信息库（NIB），用于记录拓扑、链路状态与配置
- 多个OFC实例作为热备份维持一致视图，切换时同步静态配置与动态网络状态，保证控制平面快速收敛

Routing

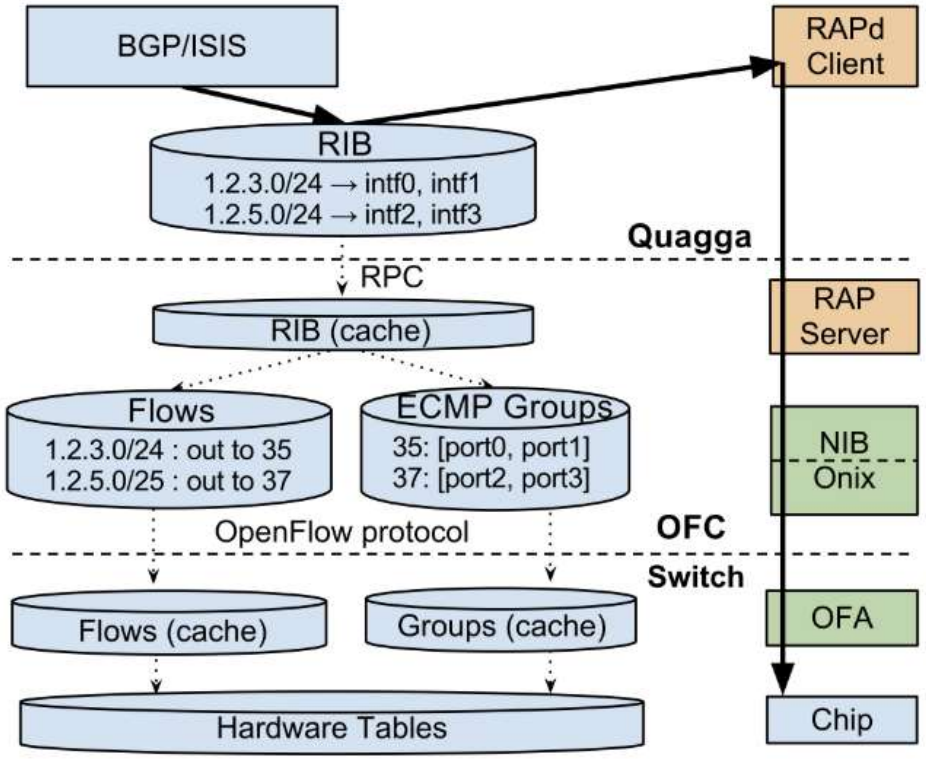
- 如何在混合网络部署中，将基于 OpenFlow 的交换机控制与现有路由协议集成，是B4 面临的主要挑战之一
- 为了实现SDN和传统路由的融合，采用了Quagga
- Quagga 软件：在 NCS 上运行开源的 Quagga，支持 BGP/ISIS 路由协议
- 路由应用代理（RAP）：作为 SDN 应用，实现 Quagga 与 OpenFlow 交换机的连通性，功能包括：
  - 路由更新转发
  - 协议数据包代理
  - 接口状态同步



RAP

路由表转换

- Quagga 为每个物理交换机端口创建 tun/tap 虚拟接口
- 将 Quagga 的路由信息库（RIB）条目转换为两张 OpenFlow 表：
  - 流表（Flow Table）：将前缀映射到 ECMP 组表条目
  - ECMP 组表（ECMP Group Table）：支持多个流共享同一组表条目
- RAPd：Routing Application Proxy daemon，RAPd进程订阅Quagga的路由信息库（RIB）更新，并通过RPC将任何变化代理到运行在OFC（OpenFlow控制器）中的RAP组件



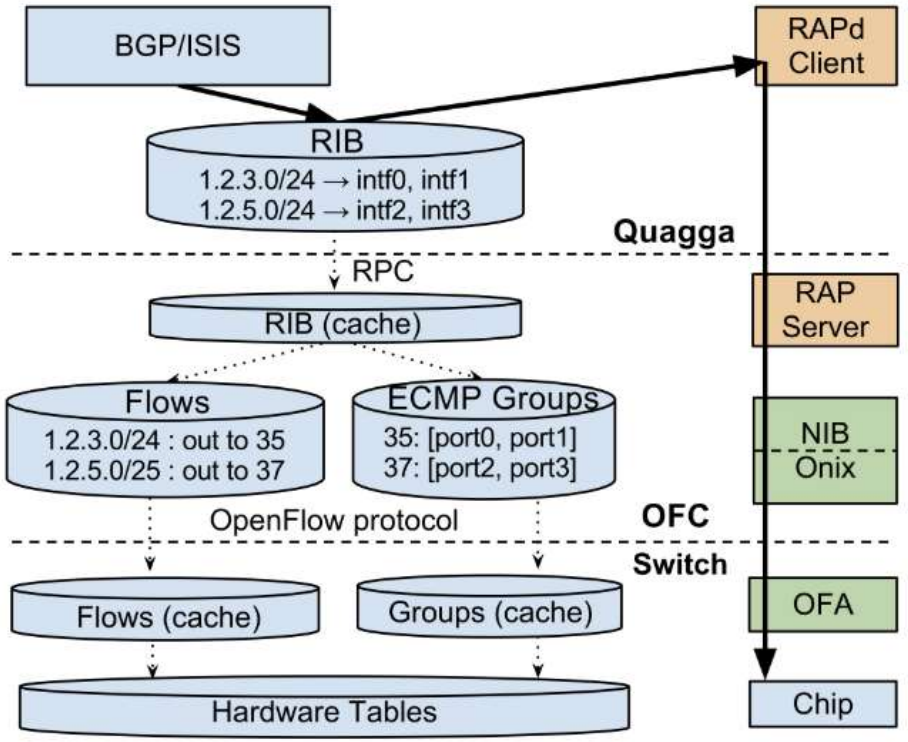
RAP

报文转发方向

- 入方向：OFA → OFC → RAPd → Quagga
- 出方向：Quagga → RAPd → OFC → OFA
- 从NCS内核开始，这些协议数据包通过RAPd、OFC和OFA转发，OFA最终将数据包放在数据平面上。我们对传入的数据包使用相反的路径。

端口状态变更流程

1. 交换机的 OFA 检测到端口状态变化，向 OFC 发送 OpenFlow 消息
2. OFC 更新本地 NIB，并将变更同步至 RAPd
3. RAPd 修改对应虚拟接口的 netdev 状态
4. 状态变更同步至 Quagga，触发路由协议更新



Outline

I. Introduction

II. Background

III. Design

IV. Traffic Engineering

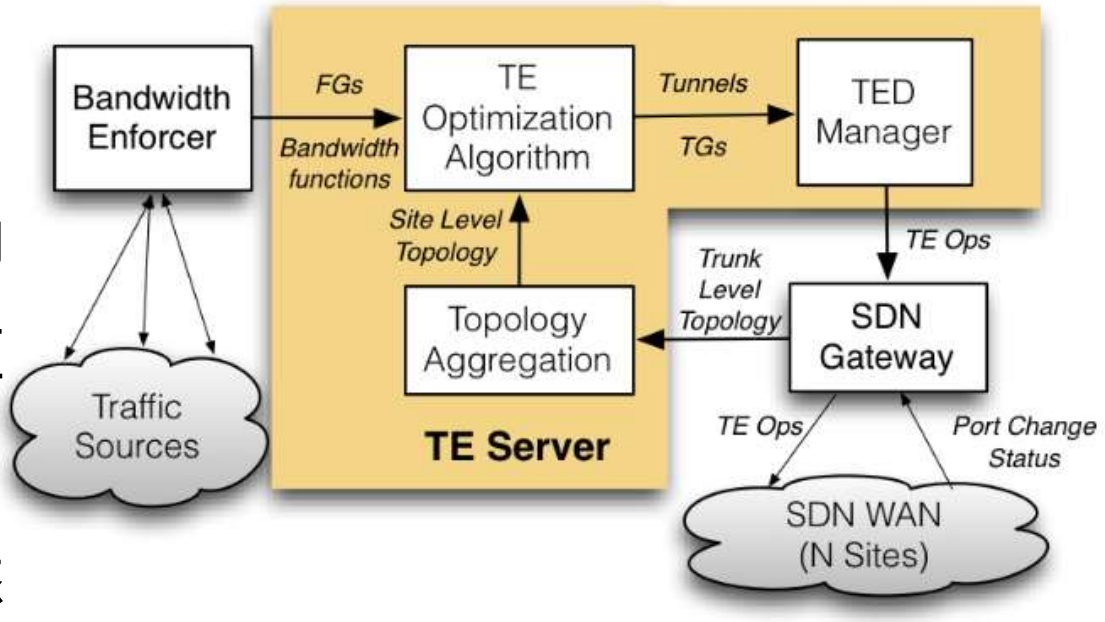
V. TE Protocol and OpenFlow

VI. Evaluation and Experience

VII. Review

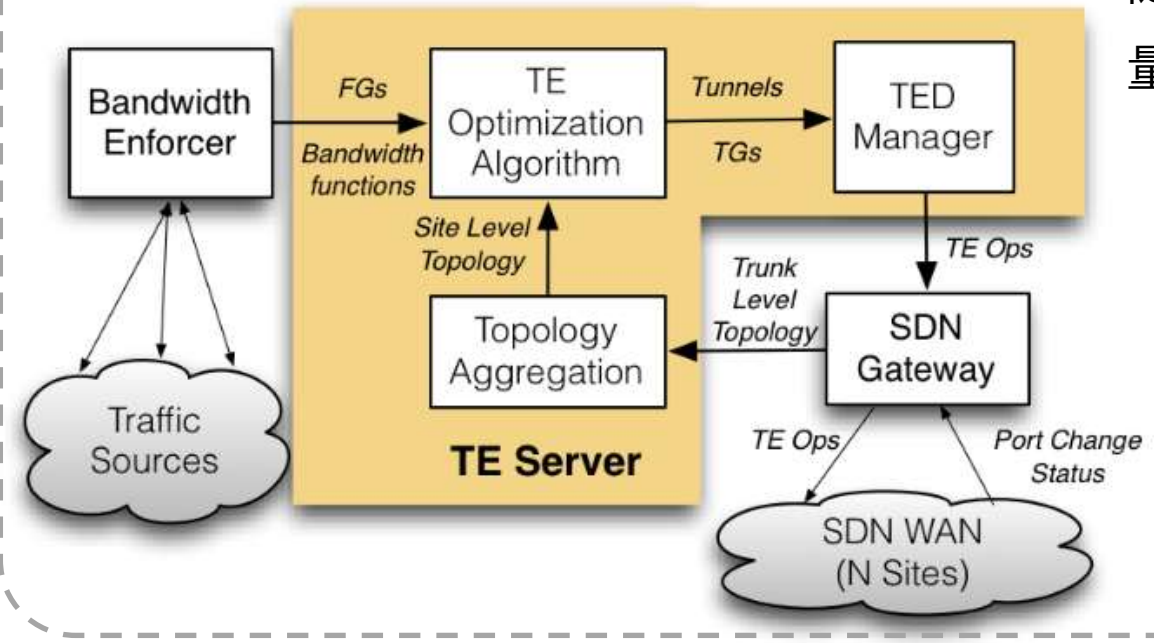
General Architecture

- B4的流量工程（TE）旨在在多应用竞争带宽的条件下，实现最大–最小公平（max-min fairness） 的带宽分配
- 该优化目标在保证资源高利用率的同时，确保任何应用的进一步收益不会以损害其他应用公平份额为代价
- 系统通过集中式调度与多路径传输实现动态带宽共享



Centralized TE Architecture

- B4的TE体系结构组成：
  - 中央TE服务器
  - SDN Gateway
  - 分布式控制层

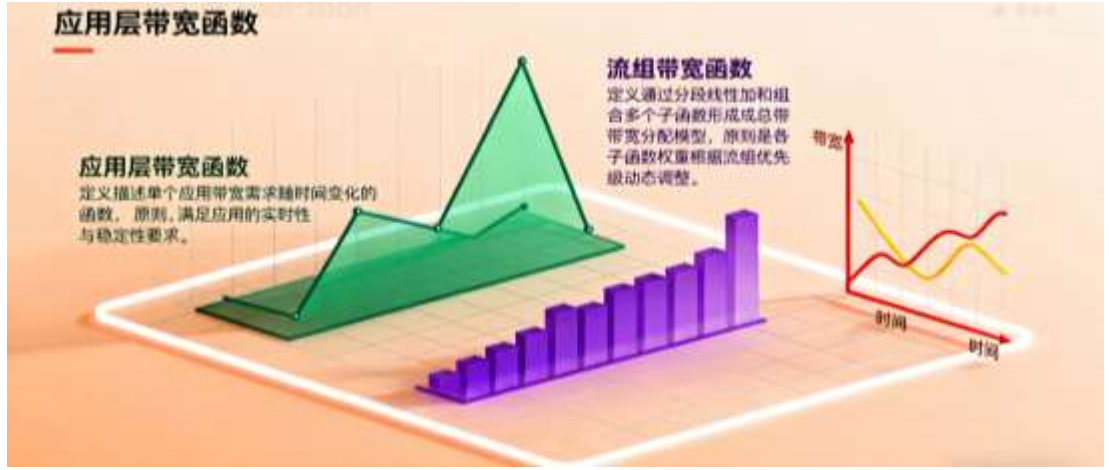


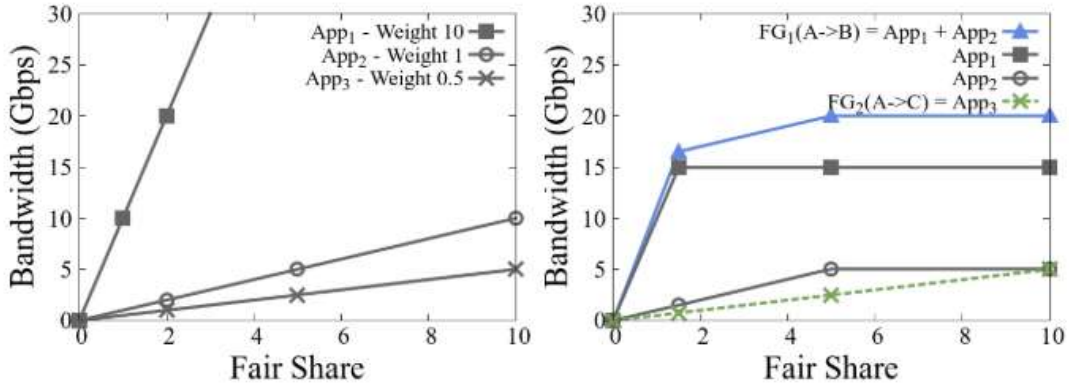
- TE服务器维护：
  - 网络拓扑图：抽象站点间连接与容量
  - 流组（Flow Group, FG）：按源站、目标站及QoS聚合应用
  - 隧道（Tunnel, T）：站点级路径序列
  - 隧道组（Tunnel Group, TG）：映射FG至多条隧道并定义流量分配比例

- TE服务器的输出通过Gateway下发至OFC，再由OpenFlow安装到交换机中

Bandwidth Function

- 应用层带宽函数
  - 定义：根据流的相对优先级（称为 “权重”），为应用分配带宽
  - 原则：每个流的带宽分配与权重成正比
  - 描述其在不同公平份额下可获得的带宽
  - 高优先级应用具有更陡的函数斜率

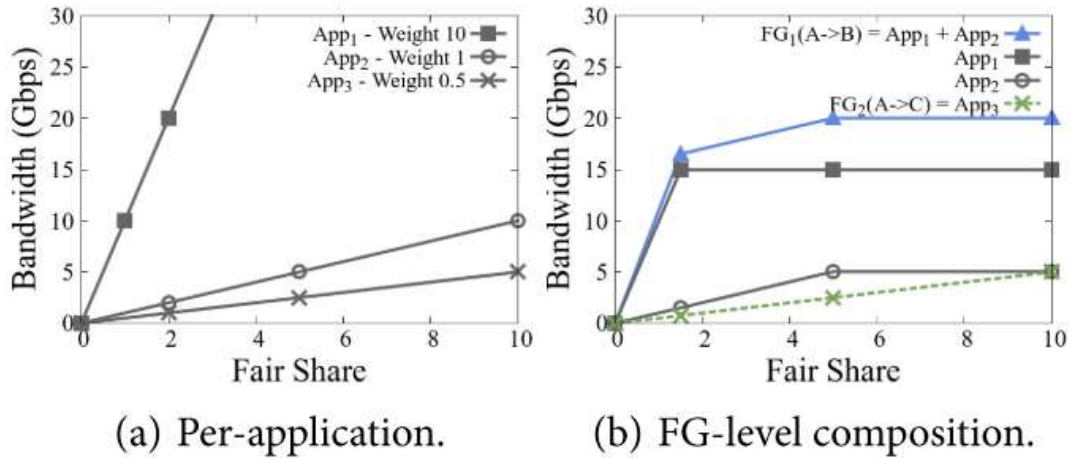




(a) Per-application. (b) FG-level composition.

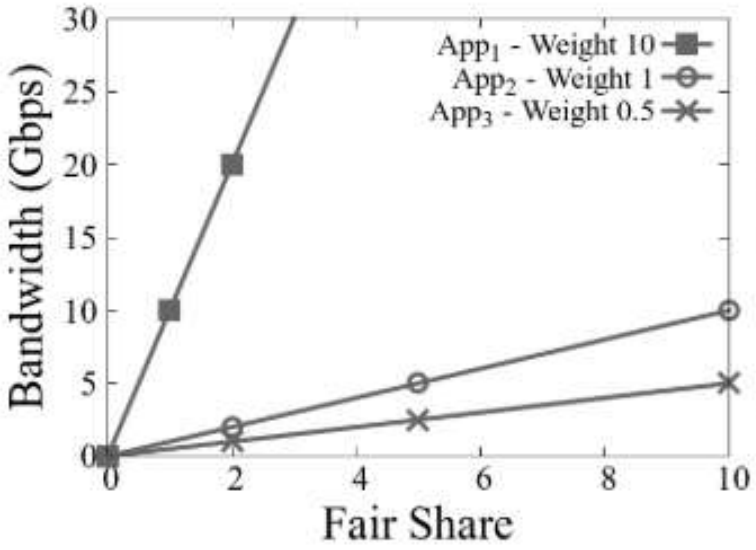
Bandwidth Function

- 流组（FG）带宽函数（Bandwidth functions）
  - 定义：流组的带宽函数是其包含的所有应用带宽函数的分段线性加和组合
- 带宽函数综合反映各应用的需求与优先级，是TE优化的基础目标函数
- 带宽强制器（ Bandwidth Enforcer ）负责在边缘实施速率限制，确保FG的实际流量不超出网络约束。

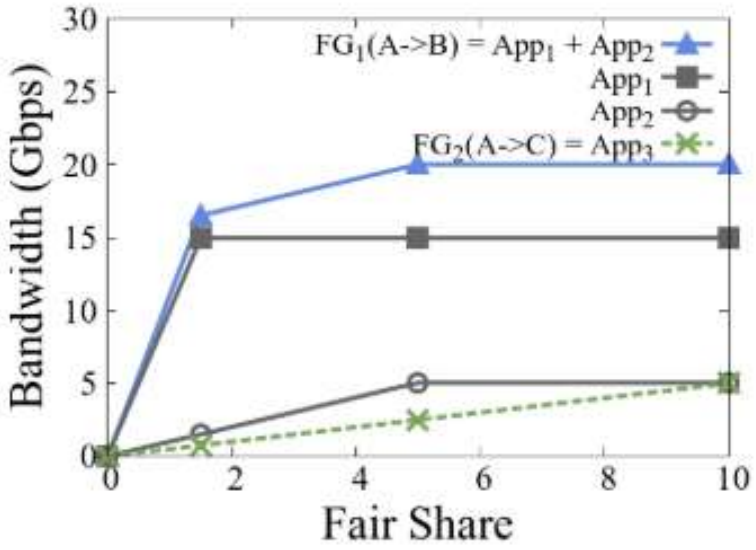


Example

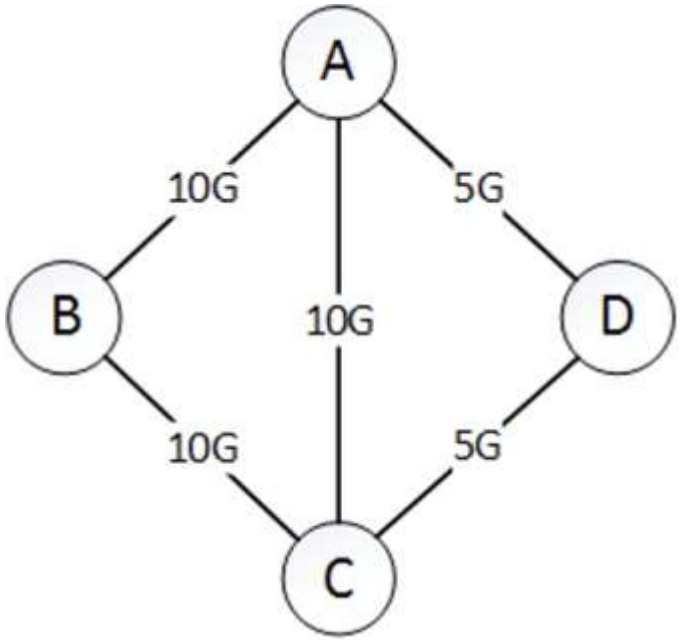
- FG1=App1+App2
  - App1 需求 15Gbps， A→B
  - App2 需求 5Gbps， A→B
- FG2=App3
  - App3 需求 10Gbps， A→C



(a) Per-application.

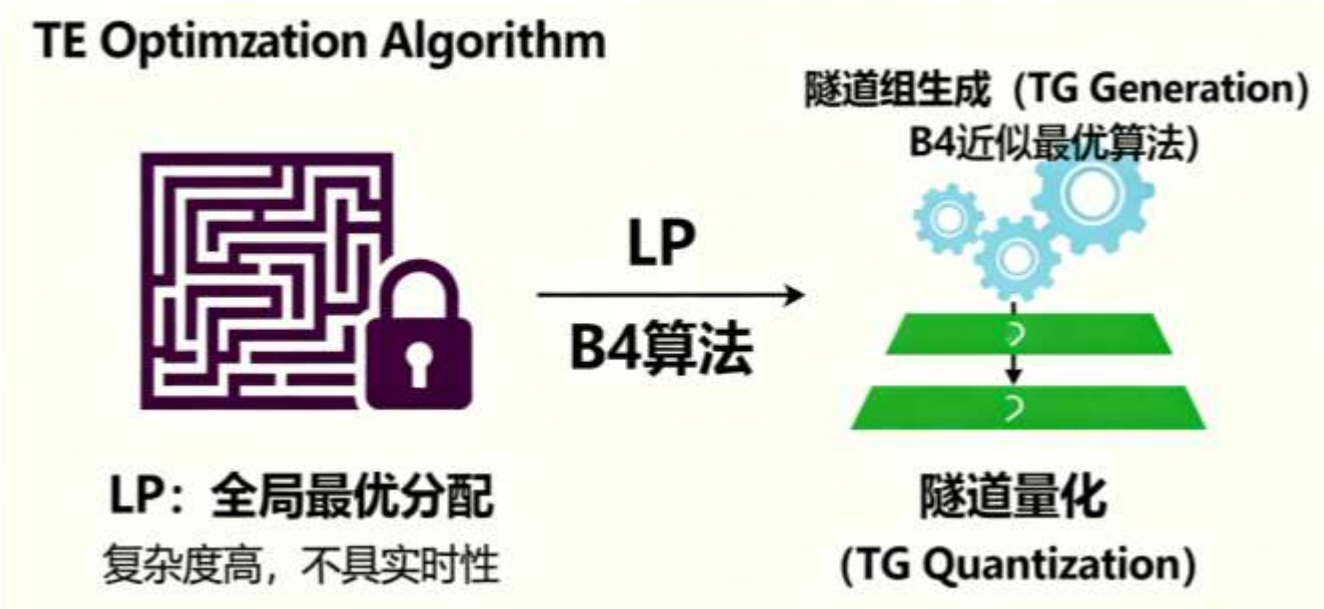


(b) FG-level composition.



TE Optimization Algorithm

- 理论上，线性规划（LP）可求得全局最优分配，但复杂度高且不具实时性
- B4设计了一种近似最优算法：
  - 隧道组生成（TG Generation）
  - 隧道量化（TG Quantization）
- 该算法实现了类似的公平性和至少99%的带宽利用率，性能相对于LP快25倍



Tunnel Group Generation

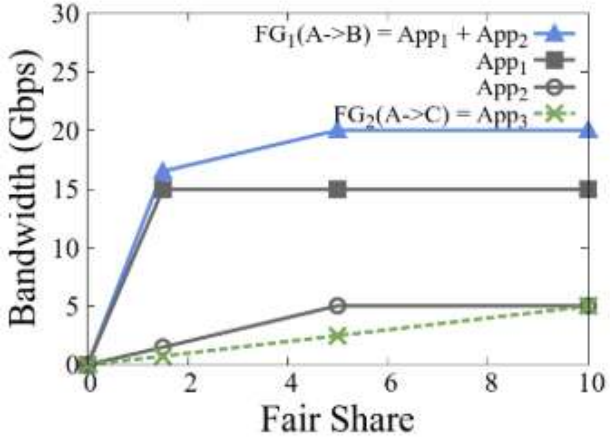
隧道组生成：

- 按流组（FG）的带宽函数，在链路间分配容量，确保同一条链路上的所有竞争流组要么获得相等的公平份额，要么完全满足自身需求
- 迭代过程：
  - 为所有流组在首选隧道上逐步增加公平份额，找到瓶颈链路
  - 冻结穿越瓶颈链路的所有隧道
  - 切换到下一条首选隧道，重复上述步骤

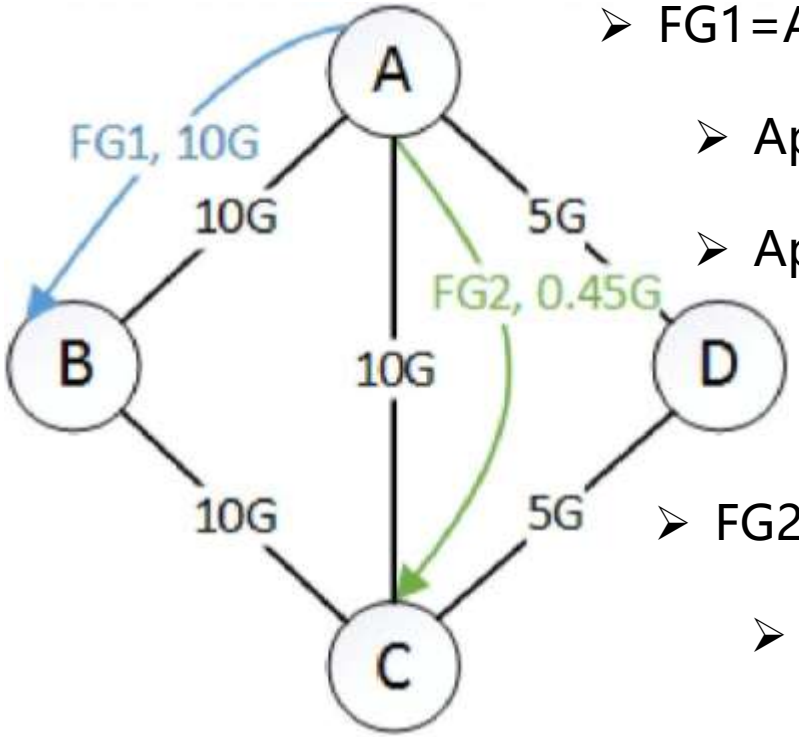
Example

步骤 1：初始分配（首选隧道）

- 隧道配置：T_{11}=A→B（FG_1 的首选隧道），T_{21}=A→C（FG_2的首选隧道）
- 公平份额 = 0.9 时，A→B 成为瓶颈链路
  - FG_1 获得带宽：10Gbps
  - FG_2 获得带宽：0.45Gbps



(b) FG-level composition.

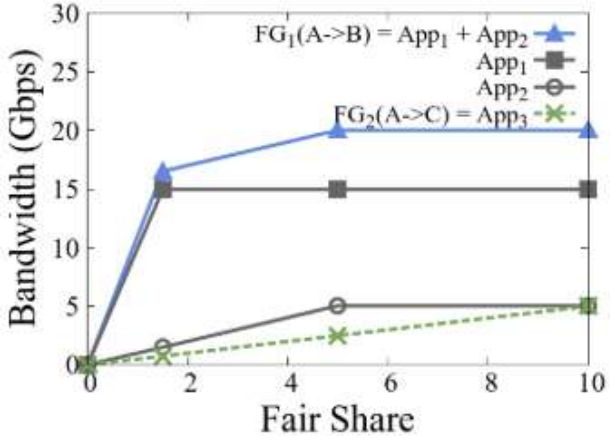


- FG1=App1+App2
  - App1 需求 15Gbps， A→B
  - App2 需求 5Gbps， A→B
- FG2=App3
  - App3 需求 10Gbps， A→C

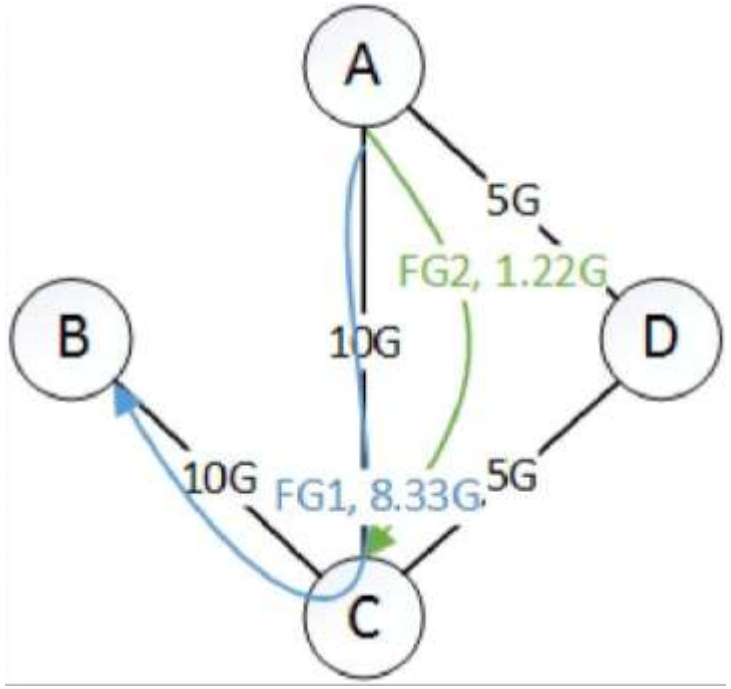
Example

步骤 2：次选隧道分配

- 启用次选隧道 T_{12}=A→C→B（FG_1 的次选隧道）
- 公平份额 = 3.33 时，A→C 成为瓶颈链路
  - FG_1累计带宽：10Gbps + 8.33Gbps=18.33Gbps
  - FG_2 累计带宽：0.45Gbps + 1.22Gbps=1.67Gbps



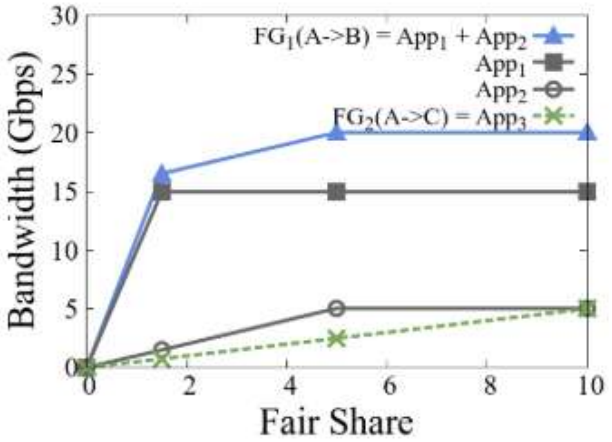
(b) FG-level composition.



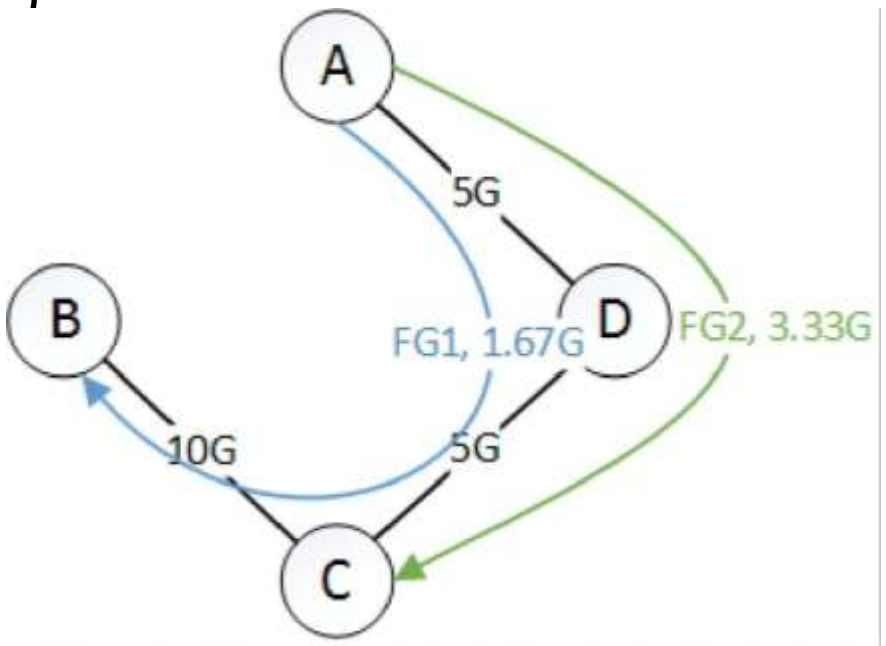
Example

步骤 3：第三隧道分配

- 启用隧道 T_{13}=A→D→C→B（FG_1）、T_{22}=A→D→C（FG_2）
- 公平份额 = 10 时，A→D 和 D→C 成为瓶颈链路
  - FG_1 需求完全满足（公平份额视为 “无限”）
  - FG_2 获得带宽：5Gbps（满足部分需求）



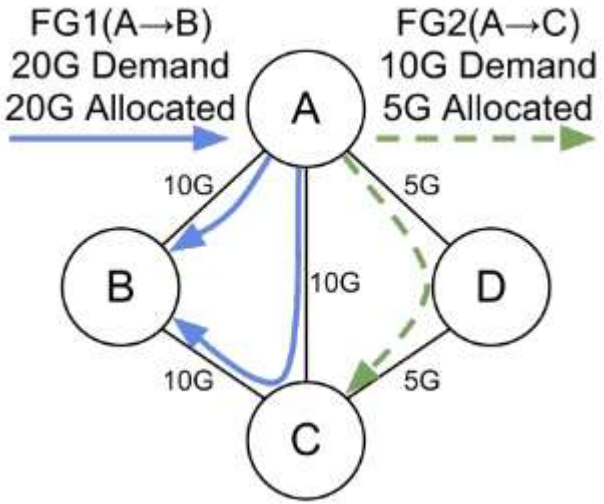
(b) FG-level composition.



Tunnel Group Quantization

- 量化目的
  - 将隧道组的带宽分配结果转换为0.5 的整数倍，简化硬件配置与流量控制
- 示例量化结果

流组（FG)	量化前分配比例	量化后分配比例
FG2	(0.3, 0.7)	(0.0, 1.0)
FG1	(0.5, 0.4, 0.1)	(0.5, 0.5, 0.0)



- 最终带宽分配
  - FG_1 （A→B，需求 20Gbps）：实际分配 20Gbps（通过两条隧道各承担 10Gbps）
  - FG_2 （A→C，需求 10Gbps）：实际分配 5Gbps（通过一条隧道承担）

Outline

I. Introduction

II. Background

III. Design

IV. Traffic Engineering

V. TE Protocol and OpenFlow

VI. Evaluation and Experience

VII. Review

TE State and OpenFlow

- B4的交换机在TE体系中承担三种角色：
- 封装交换机（Encap）负责建立隧道并分流流量
- 中继交换机（Transit）依据外层头部转发
- 解封装交换机（Decap）识别隧道并恢复内层包头。

TE Construct	Switch	OpenFlow Message	Hardware Table
Tunnel	Transit	FLOW_MOD	LPM Table
Tunnel	Transit	GROUP_MOD	Multipath Table
Tunnel	Decap	FLOW_MOD	Decap Tunnel Table
Tunnel Group	Encap	GROUP_MOD	Multipath table, Encap Tunnel table
Flow Group	Encap	FLOW_MOD	ACL Table

Table 2: Mapping TE constructs to hardware via OpenFlow.

TE State and OpenFlow

- 流组（FG）匹配：交换机通过数据包的目的 IP 地址与 FG 关联的前缀匹配，将数据包映射到对应的 FG
- 隧道哈希选择：入站数据包通过哈希算法，按期望比例分配到隧道组（TG）中的某一条隧道
- 隧道封装：源站点的交换机为数据包添加外层 IP 头部，外层目的 IP 地址作为隧道 ID（非实际目的地址）

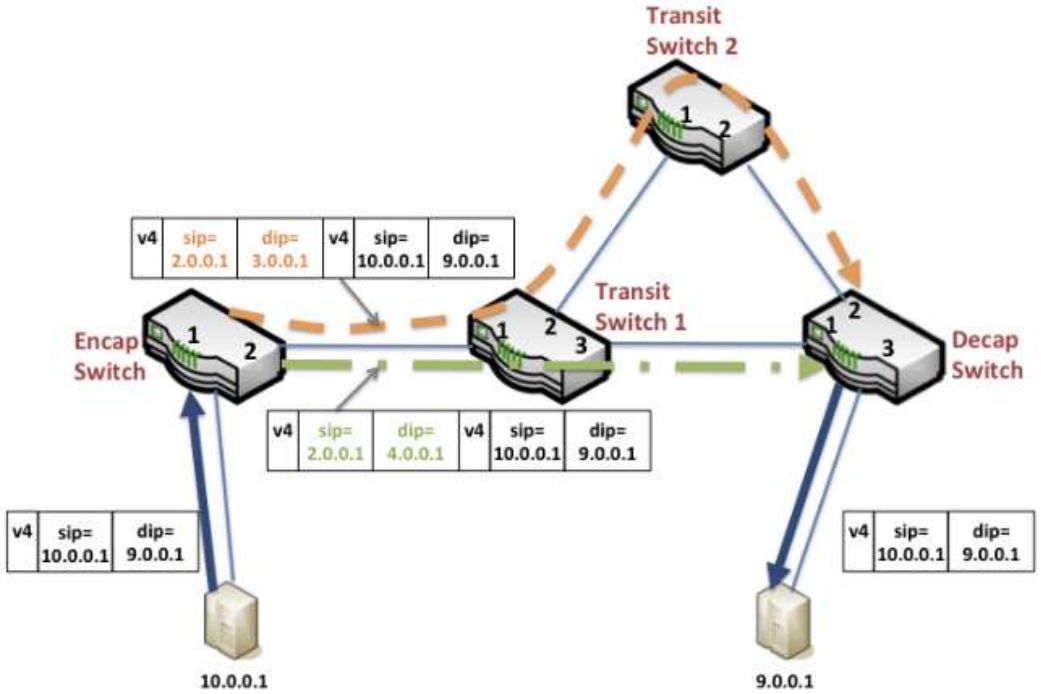
TE Construct	Switch	OpenFlow Message	Hardware Table
Tunnel	Transit	FLOW_MOD	LPM Table
Tunnel	Transit	GROUP_MOD	Multipath Table
Tunnel	Decap	FLOW_MOD	Decap Tunnel Table
Tunnel Group	Encap	GROUP_MOD	Multipath table, Encap Tunnel table
Flow Group	Encap	FLOW_MOD	ACL Table

Table 2: Mapping TE constructs to hardware via OpenFlow.

Example

封装阶段

- 50% 的流：外层 IP 地址为 2.0.0.1（源）、4.0.0.1（目的），沿最短路径转发
- 50% 的流：外层 IP 地址为 2.0.0.1（源）、3.0.0.1（目的），通过中转站点转发



解封装阶段

- 移除外层 IP 头部
- 根据内层数据包头部的最长前缀匹配（LPM）条目（由 BGP 生成），将数据包转发至目的地址

Composing Routing with TE

- B4 同时支持最短路径路由（shortest-path routing）和流量工程（Traffic Engineering, TE），从而保证即使 TE 被禁用，系统也能继续运行。为了实现这两种路由服务的共存，B4 利用了商用交换芯片中对多转发表（multiple forwarding tables）的支持。

表项协作机制

- 路由（BGP）的作用：在 LPM 表（最长前缀匹配表）中填充对应的转发条目，定义基础转发路径
- TE 的作用：通过 ACL 表（访问控制列表）设置自定义转发行为，实现流量工程需求
- 优先级规则：ACL 表规则的优先级严格高于LPM 表规则，确保 TE 策略优先执行

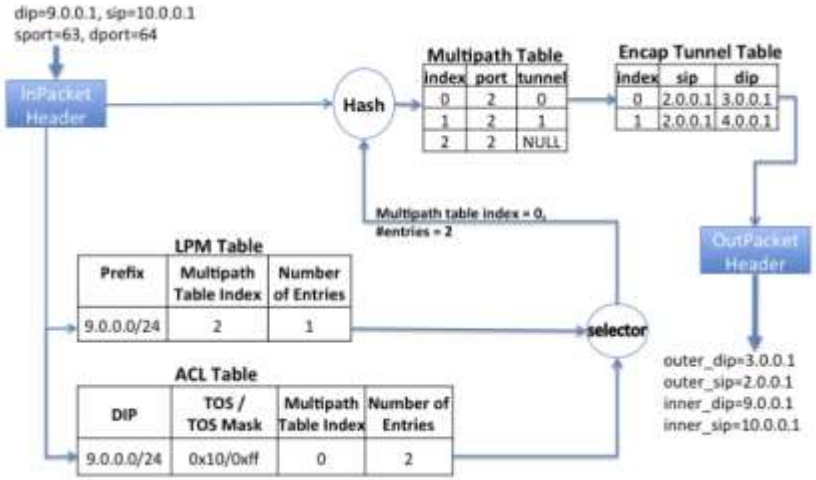
Example

输入数据包头部

- 源 IP（sip）=10.0.0.1 目的 IP（dip）=9.0.0.1
- 源端口（sport）=63 目的端口（dport）=64

表项匹配顺序

- 步骤 1：数据包同时匹配 LPM 表和 ACL 表（并行匹配）
  - LPM 表：9.0.0.0/24 前缀对应输出端口 2，无隧道封装
  - ACL 表：匹配内层 IP（sip=10.0.0.1，dip=9.0.0.1），指向多路径表索引 0（含 2 个条目）
- 步骤 2：因 ACL 优先级更高，执行多路径表逻辑
  - 交换机对数据包头部哈希，按多路径表条目数量（2）取模（ECMP 哈希）
  - 结果：将目的为 9.0.0.0/24 的流均匀分配到两条隧道



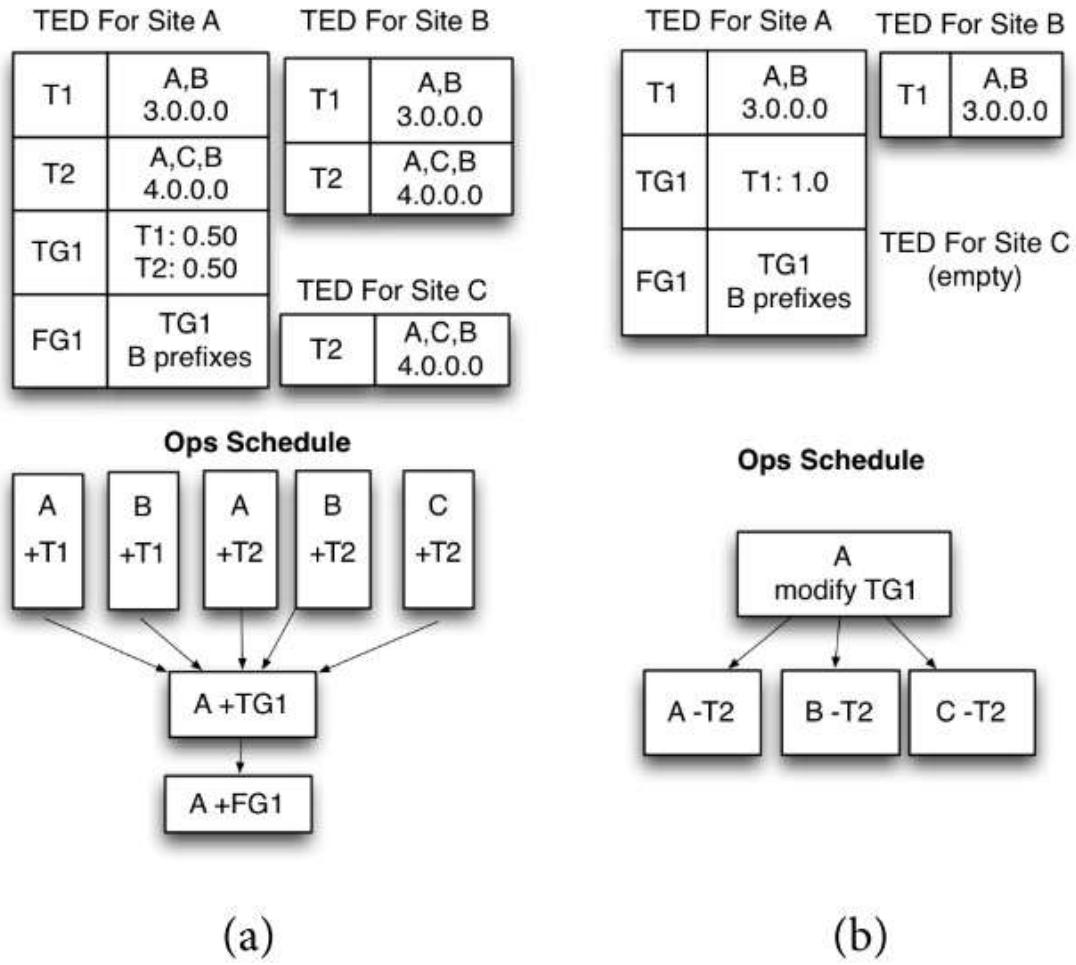
Coordinating TE States Across Sites

- TE服务器需在多个控制器间同步Tunnels、Tunnel Groups与Flow Groups状态
- 核心组件：流量工程数据库（TED）
  - 定位：全局键值存储，用于管理所有站点的隧道（T）、隧道组（TG）、流组（FG）
  - 生成逻辑：基于 TE 算法输出的 TG、FG、T，计算每个站点的 TED
- TE 操作（TE Op）规则
  - 原子性：单个 TE 操作仅能添加、删除或修改一个TED 条目
  - 下发流程：OFC 将 TE 操作转换为该站点所有设备的流编程指令

Dependencies

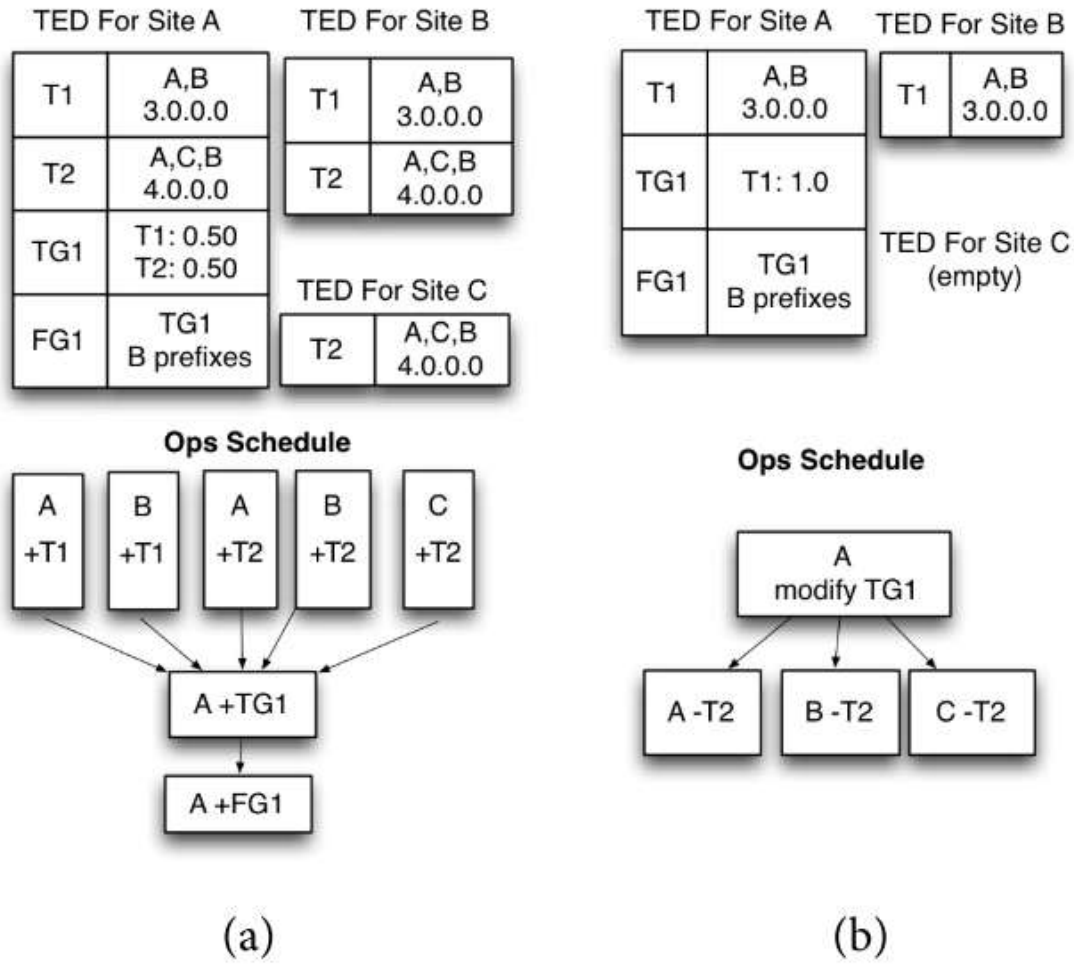
依赖关系：

- 隧道删除：必须先删除依赖该隧道的TG/FG 条目，再删除隧道
- 隧道创建：需先在所有相关站点创建隧道，再创建引用该隧道的 TG/FG
- 为防止乱序，TE通过站点级序列号与会话ID强制操作顺序一致性，保证配置的幂等性与可恢复性



Synchronizing TED between TE and OFC

- 在分布式环境下，TE服务器与OFC之间通过TE Session保持同步视图
- 每个会话具有唯一标识，启动时双端同步TED状态并周期性持久化
- 若控制平面通信中断或OFC失联，系统使用“强制失败（force fail）”策略阻止新操作下发，避免状态错乱或链路中断



TE Op Failures

- 当TE操作执行失败（如RPC错误、设备编程失败）时，TED将对应项标记为“Dirty”，系统在超时后自动重试，直到收到确认
- 由于TE操作设计为幂等性，重复执行不会导致不一致
- 该机制显著增强了系统的健壮性，使B4能够在部分组件失效时仍维持全网状态一致。

Outline

I. Introduction

II. Background

III. Design

IV. Traffic Engineering

V. TE Protocol and OpenFlow

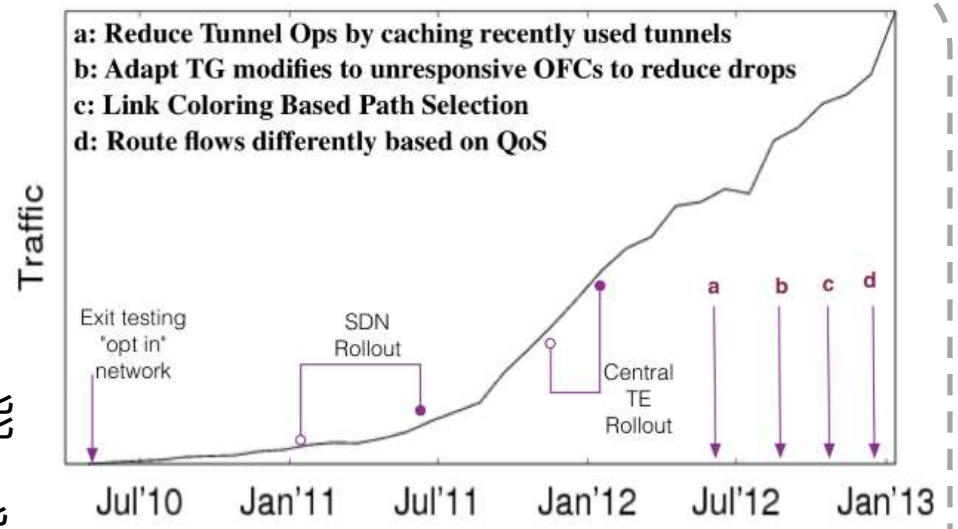
VI. Evaluation and Experience

VII. Review

Deployment and Evolution

- 基础架构阶段（2010-2011）：部署OpenFlow交换机和控制平面基础
- 路由集成阶段（2011）：集成BGP/ISIS传统路由协议
- 流量工程阶段（2012）：部署集中式TE系统，实现核心优化功能
- 优化完善阶段（2012后期）：添加缓存、自适应机制等增强功能

- 2012年9月至11月B4关键属性
- 链路容易频繁发生端口抖动，而动态集中式管理可对此起到改善作用



(a) TE Algorithm

Avg. Daily Runs	540
Avg. Runtime	0.3s
Max Runtime	0.8s

(b) Topology

Sites	16
Edges (Unidirectional)	46

(c) Flows

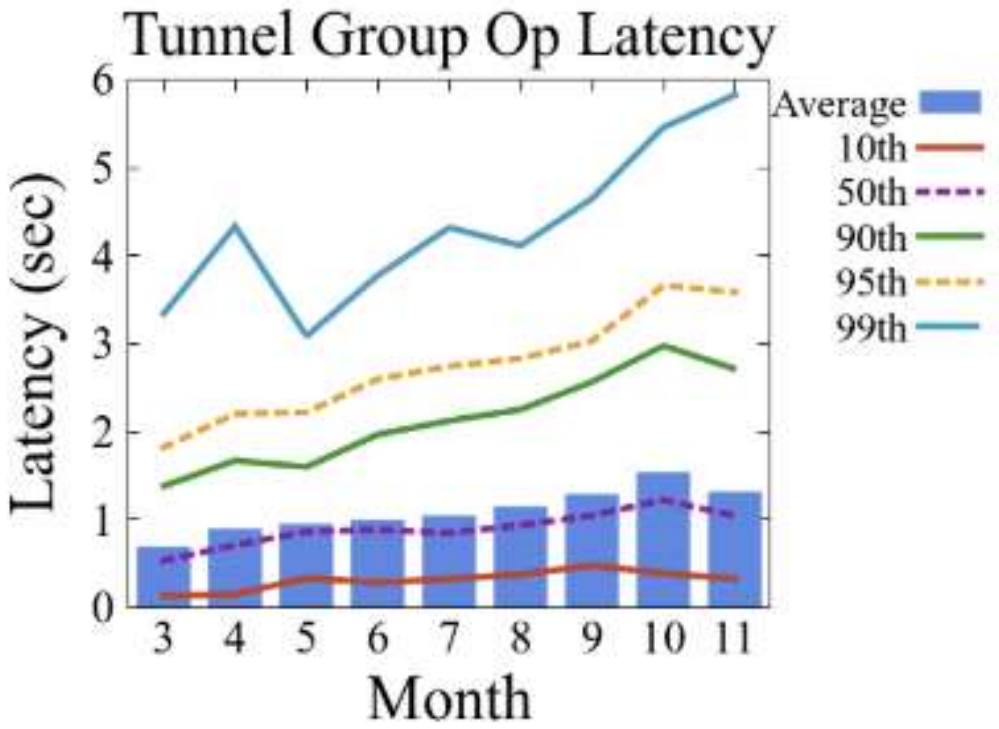
Tunnel Groups	240
Flow Groups	2700
Tunnels in Use	350
Tunnels Cached	1150

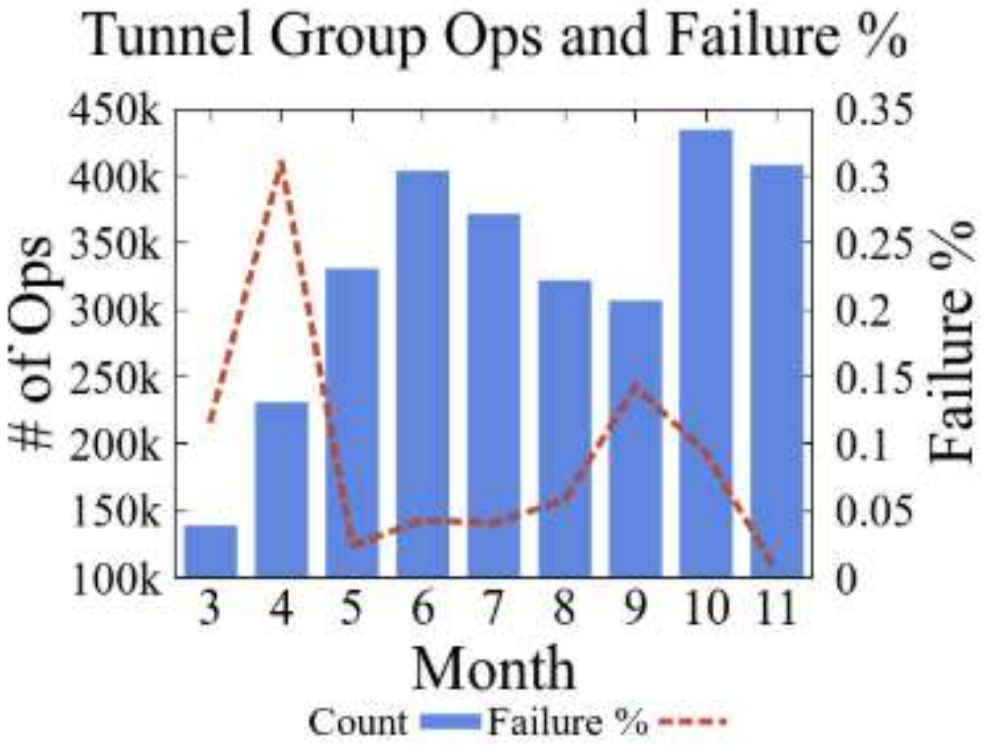
(d) Topology Changes

Change Events	286/day
Edge Add/Delete	7/day

TE Ops Performance

- 隧道组操作

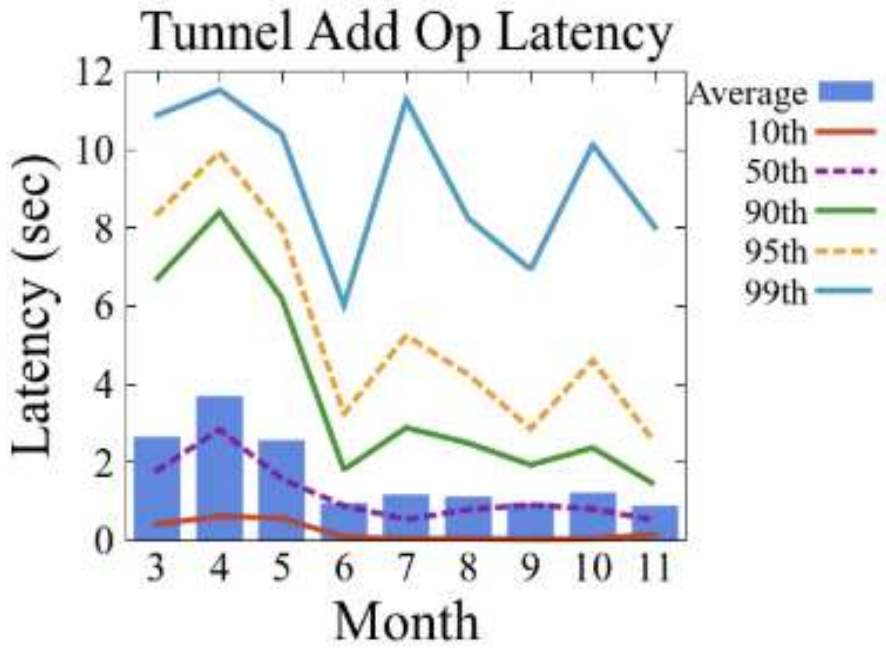


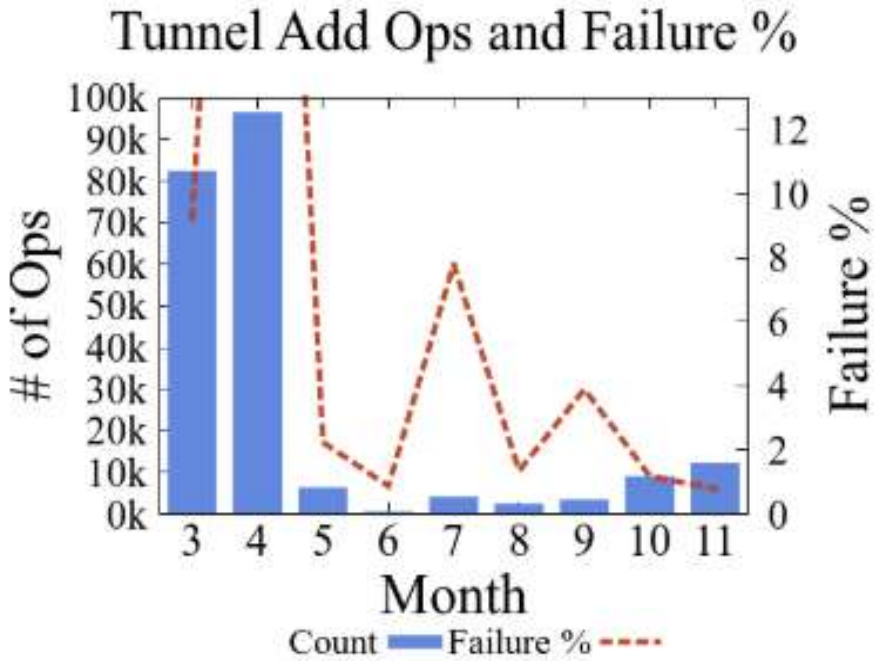


TE Ops Performance

隧道操作

- 通过优化缓存最近使用的隧道，隧道操作减少了近100倍，这也会减少导致失败操作





TE Ops Performance

- 操作延迟与交换机瓶颈

Op Latency Range (s)	Avg Daily Op Count	Avg STF	10th-perc STF
0-1	4835	0.40	0.02
1-3	6813	0.55	0.11
3-5	802	0.71	0.35
5-∞	164	0.77	0.37

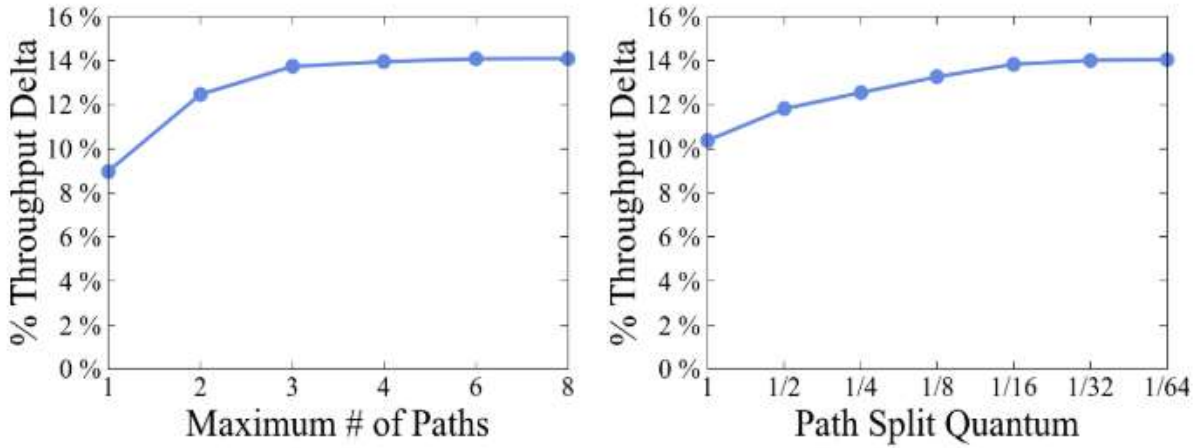
Table 4: Fraction of TG latency from switch.

Impact of Failures

Failure Type	Packet Loss (ms)
Single link	4
Encap switch	10
Transit switch neighboring an encap switch	3300
OFC	0
TE Server	0
TE Disable/Enable	0

- 单链路故障仅导致毫秒级流量中断（约几毫秒）
- 与封装路由器相邻的中转路由器发生故障时，需要长得多的收敛时间
  - 对于所有穿越故障路由器的隧道，需更新多路径表中所有受影响的条目
- OFC和TE服务器故障/重启都是无故障的
- 当流量工程（TE）故障时，回退到边界网关协议（BGP）

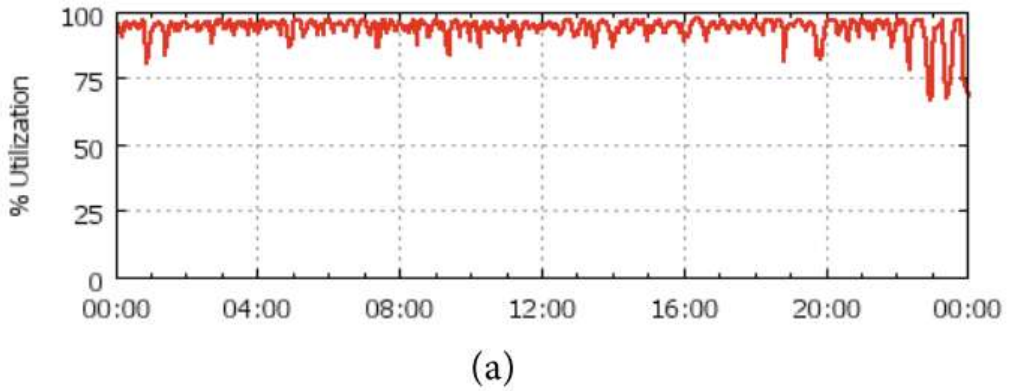
TE Algorithm Evaluation



- (a) 假设路径分割粒度为 1/64，以此分析（算法）对可用路径数量的敏感度
- (b) 将最大路径数量固定为 4，以此展示路径分割粒度带来的影响
- 当允许使用多条路径时，TE能显著提升全局吞吐量；随着路径数增加至4条，改善趋于平稳
- 路径分割粒度越细吞吐量越高
- 在实际部署中，B4采用4路径和1/4分割量子，平均吞吐量提升14%，尤其在故障或高需求时期效益更明显

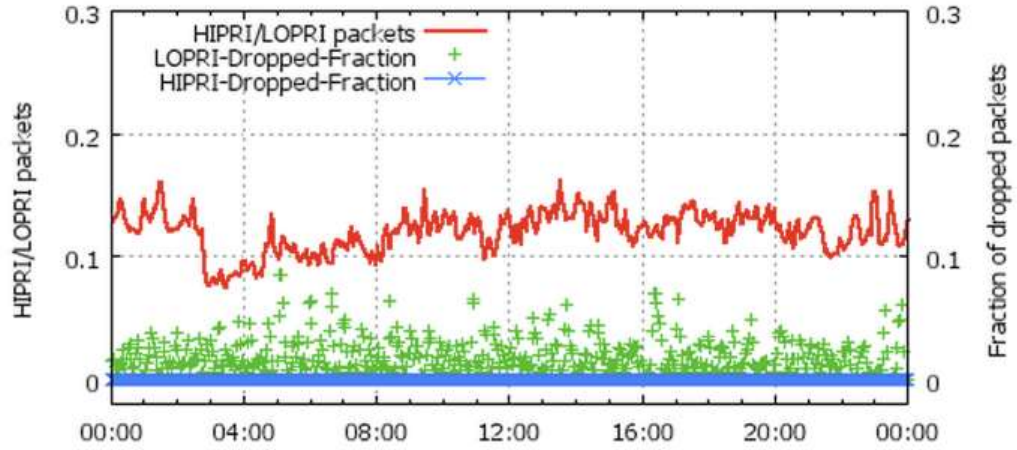
Link Utilization and Hashing

- 100%利用率



(a)

- 高优先级数据包与低优先级数据包的比例，以及各优先级数据包的丢包率

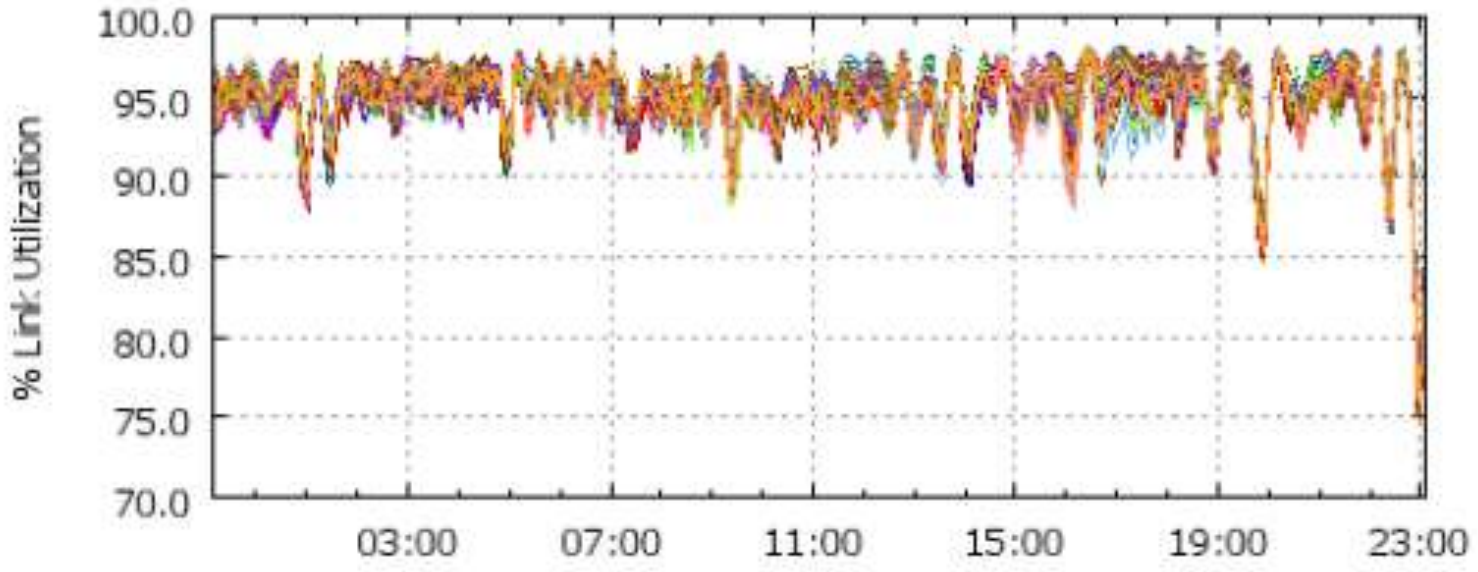


(b)

Link Utilization and Hashing

trunk中的每个链路的利用率

- 对于至少75%的站点到站点边缘，跨组成链接的链接利用率的最大:最小比率为1.05，无故障(即，从最优状态的5%)，2.0有故障。



Outline

I. Introduction

II. Background

III. Design

IV. Traffic Engineering

V. TE Protocol and OpenFlow

VI. Evaluation and Experience

VII. Review

Review

- 谷歌构建 SDN WAN（B4）的核心原因
- B4 的系统架构
  - 站点层
  - 全局层
- B4 中 BGP 的实现方式
- TE 算法核心

Review

- B4 优势总结：
- **成本效率与硬件创新：**采用商用硬件和自定义交换机，大幅降低资本支出和运营成本
- **高链路利用率与流量优化：**集中式流量工程驱动链路利用率近100%，远超传统WAN
- **应用感知与优先级管理：**基于带宽函数实现应用级优先级调度，保障关键业务

Review

- B4 优势总结：
- **可靠性与故障恢复：**多层故障恢复机制，实现毫秒级故障切换和高可用性
- **灵活性与可扩展性：**SDN架构支持快速功能迭代和网络规模扩展
- **运维简化与管理效率：**集中视图和自动化运维，大幅降低管理复杂度