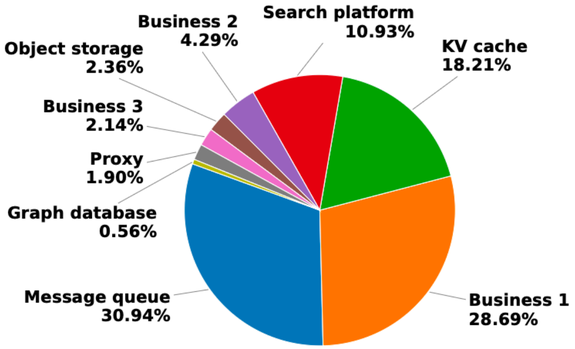
Net-P4ct: Enhanced WAN Bandwidth Fair Sharing Using P4 Programmable Switches

Haoran Chen, Mingwei Cui, Yihan Zou, Yihang Miao, Suhan Jiang, Damu Ding, Lirong Lai, Ming Gao, Rui Jiang, Shengyuan He, Anjian Chen, Jiaming Shi, Junjie Wan, Yandong Duan, Ruomin Fang, Hongyu Wu, Yongping Tang, Qiao Kang, Guangrui Wu, Xiyun Xu

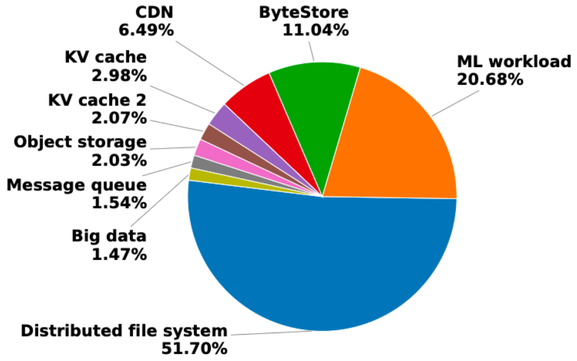
ByteDance

Challenge

- Hundreds of services with diverse business needs share the backbone
  - Cloud services/live-streaming/on-demand videos/internal support services
  - O(10) Tbps -> O(100) Tbps in the past few years
- Bandwidth demand grows fast, but expansion is expensive and slow
- Services have different SLOs



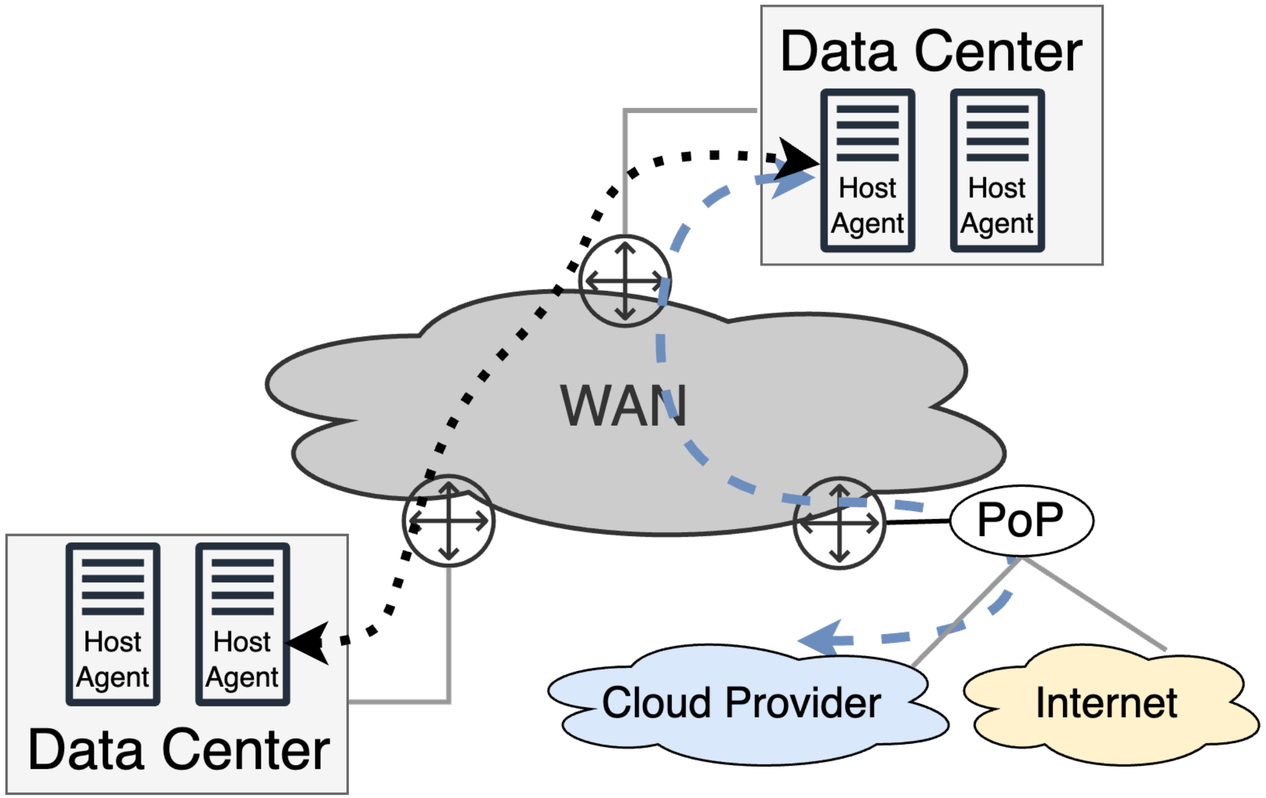
(a) High Priority

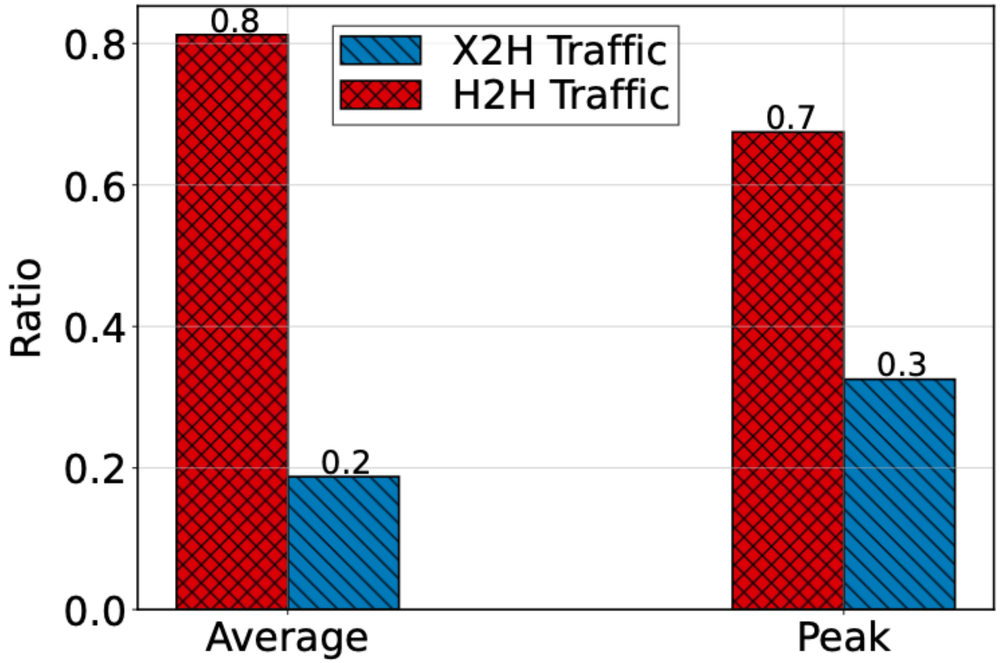


(b) Low Priority

- A few infrastructure services and main business applications account for majority of network traffic
- DFS contributes the majority of low priority traffic

Traffic from Multiple Sources



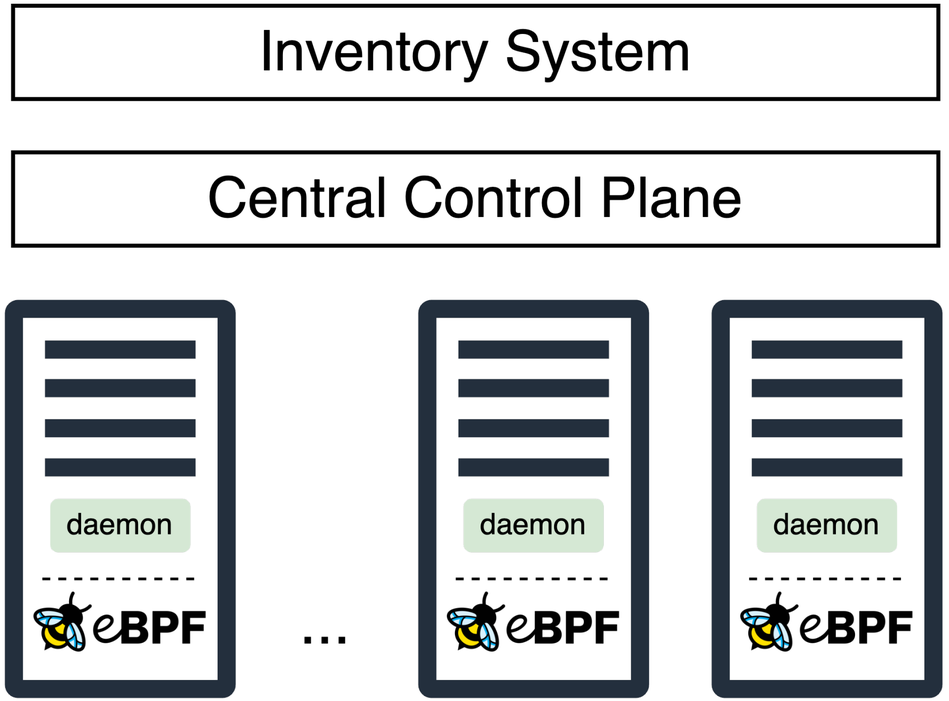


one of our backbone regions

- Host-to-host (H2H): traffic originates from services running in internal servers
- External-to-host (X2H): traffic originates from external cloud services or the Internet to internal hosts

Our Existing Solution: NetAgent (1/2)

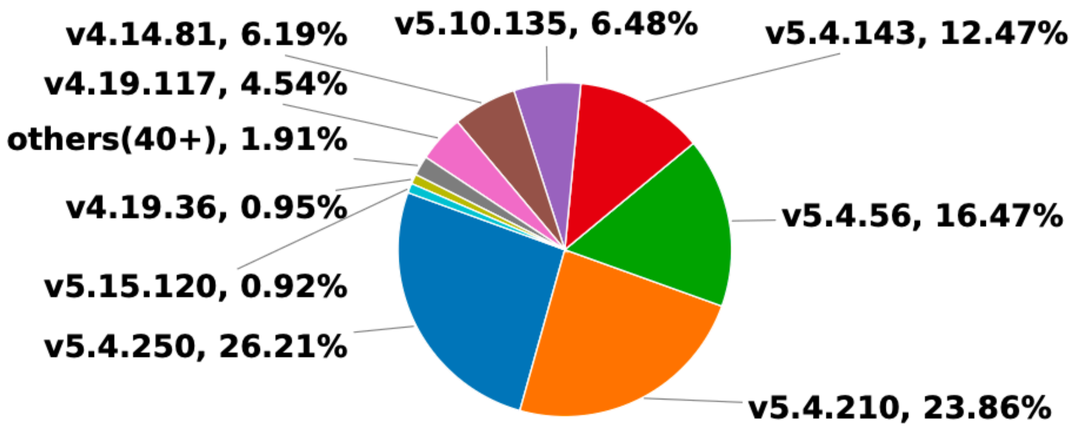
- eBPF program
  - Traffic monitoring
  - Packet header modification
  - Traffic throttling
- Central control plane + Inventory system
- Works with Kubernetes cluster
  - Integrate eBPF with cgroups



[BwE, SIGCOMM'2015] [Network Entitlement, SIGCOMM'2022]

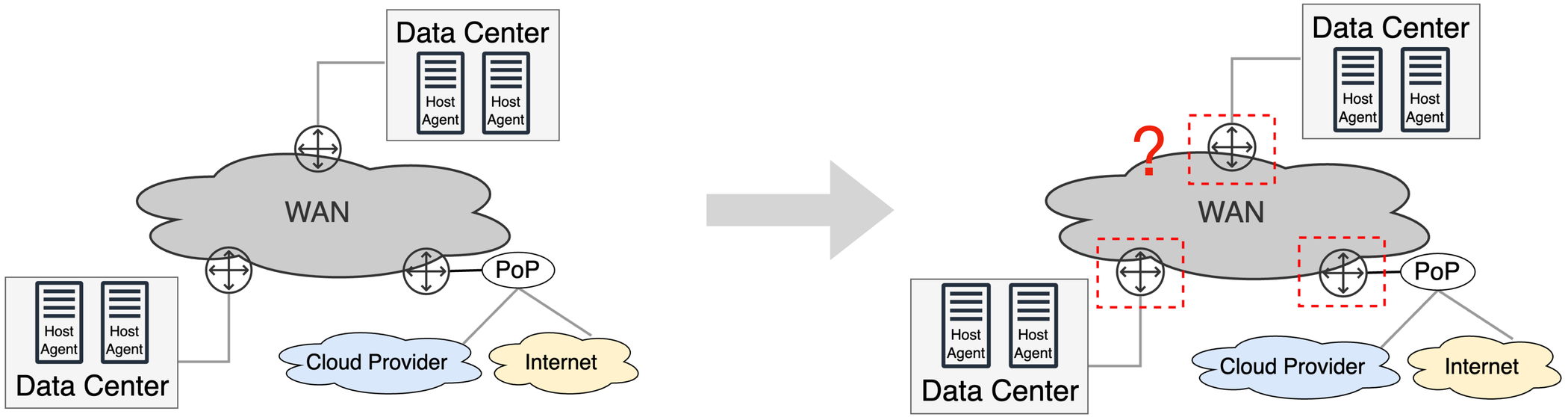
Our Existing Solution: NetAgent (2/2)

- Traffic throttling is compute-intensive:
  - 1million NetAgents use O(20k) CPU cores
- Diverse kernel versions:
  - 50+ different kernel versions in the production
  - Incidents caused by kernel bug (e.g., packets are mistakenly marked with a large timestamp)
- Traffic from “blind spots”:
  - X2H traffic cannot be captured by NetAgent



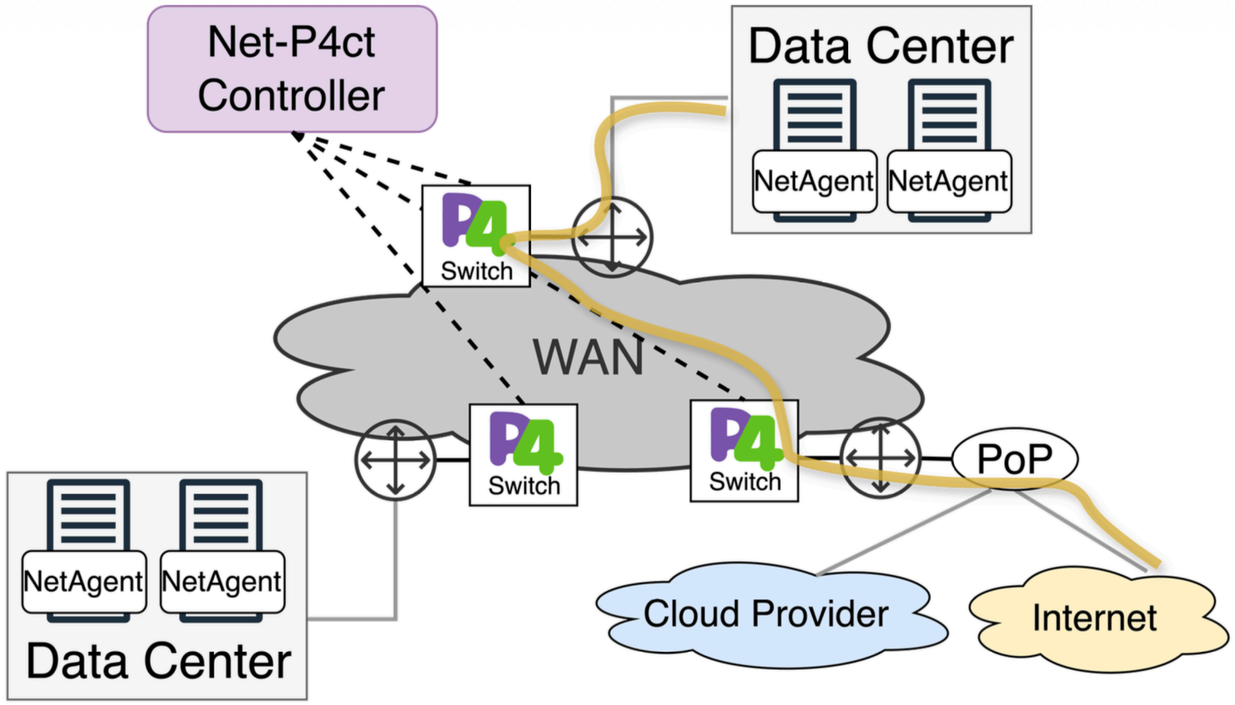
Need for Flexible In-Network Solution

- The components should locate on the critical paths of WAN
- What about conventional network devices (PE switches)?
  - They provide limited flexibility
  - Frequent updating configurations is not ideal for stability

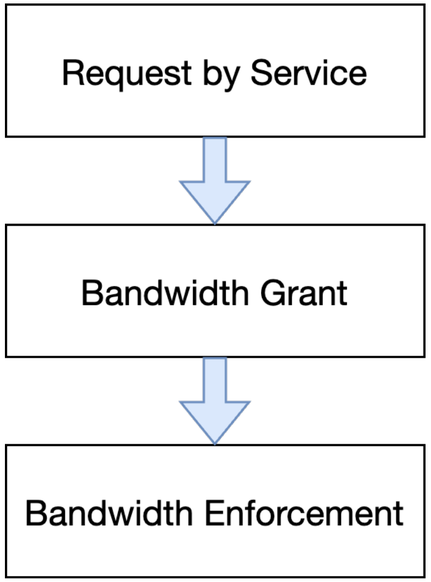


Net-P4ct Overview

- A WAN-wide bandwidth management and enforcement system across services in geographically distributed data centers



- P4 switch: in-network bandwidth enforcement
- NetAgent: tag packets with ID



Net-P4ct Workflow

Bandwidth Request Model Abstraction

- *Job* is the minimal unit of bandwidth request
  - { *ID, Service, Pattern, RegionA, RegionZ, GuaranteedBandwidth, Weight* }
- *Pattern* describes traffic characteristics of the job
  - Host-to-host pattern: Our containerized infrastructure distinguishes job with a *jobID*
  - External-to-host pattern: 3-tuple { *srcIPPrefix, dstIPPrefix, proto* }
- *GuaranteedBandwidth* represents a strict requirement.
- *Weight* reflects the priority

Bandwidth Request Grant (1/2): Model

- Link capacity model: each link's capacity is divided into
  - Reserved capacity: high-priority traffic and guaranteed bandwidth allocations
  - Non-reserved capacity: best-effort traffic and non-critical services

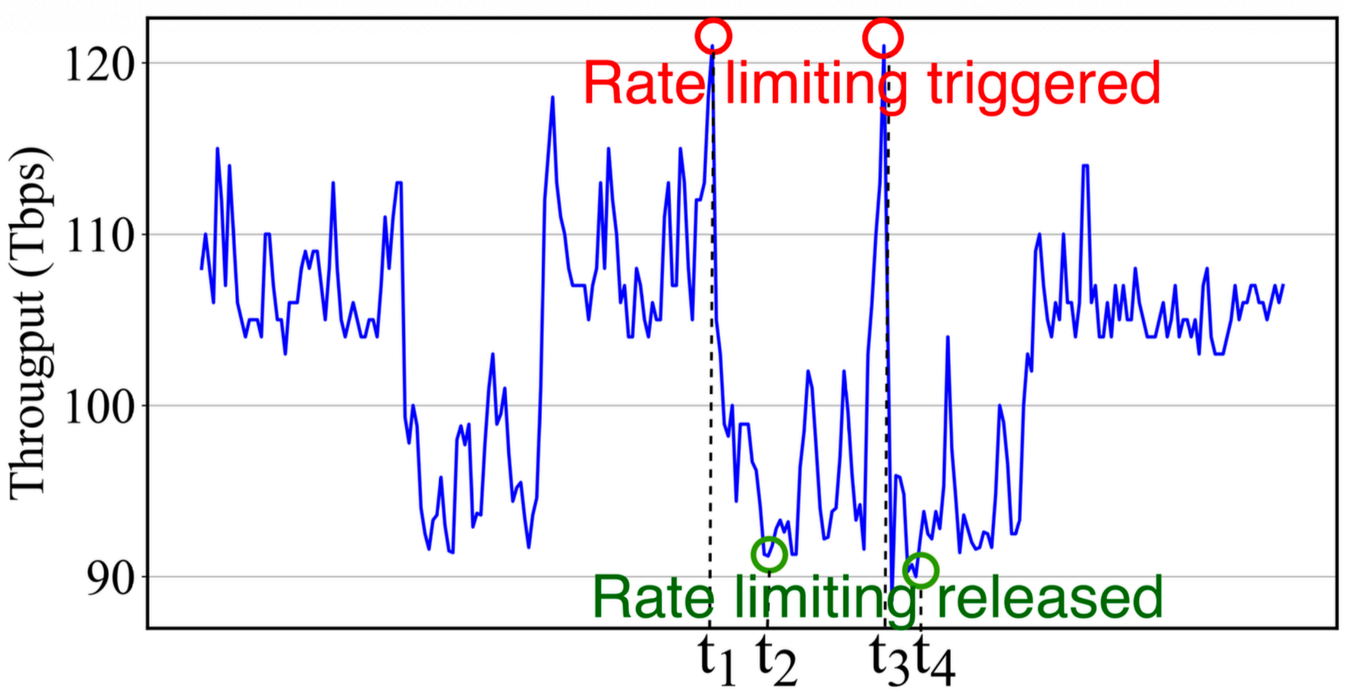
Bandwidth Request Grant (2/2): Calculation

- Object: max(*overall_network_throughput*)
- Constraints:
  - Work-conserving
  - Fair share bandwidth refers to that allocated from the non-reserved capacity (additional bandwidth)
  - Weighted max-min fairness model
- Algorithm: water-filling

Details in the paper

Bandwidth Enforcement

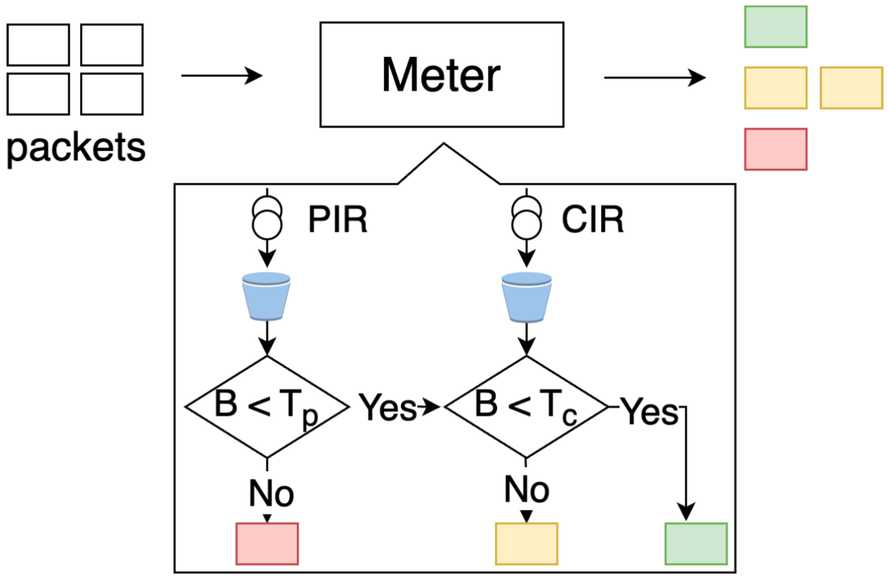
- Traffic throttling introduces throughput fluctuations due to threshold



- Use P4-switches to remark DSCP and downstream WAN switches enforce QoS policies with strict priority queuing

Meter-based Policy (MBP)

- A two-rate three-color marker (trTCM) for each service job
- Classify packets into three priority levels: green, yellow, or red

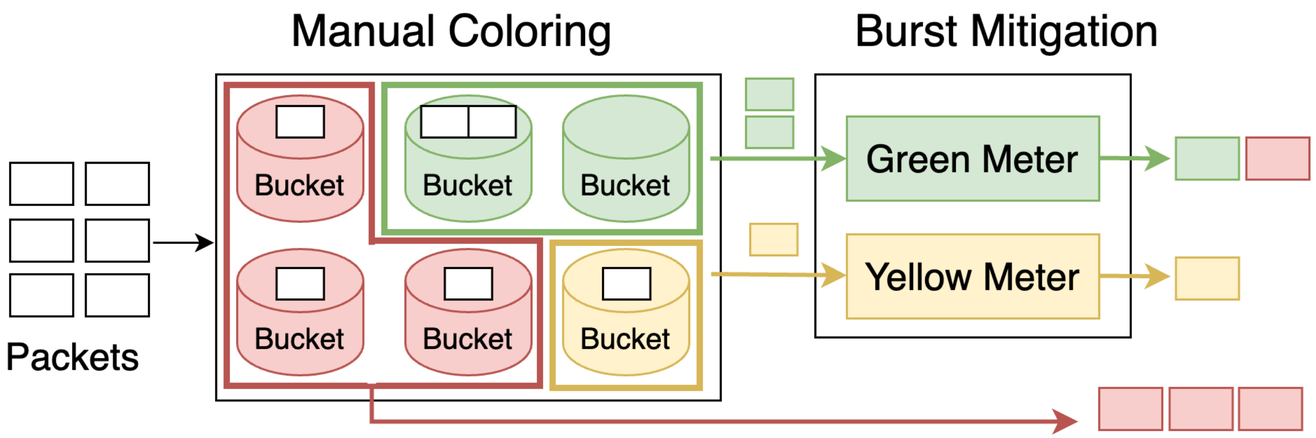


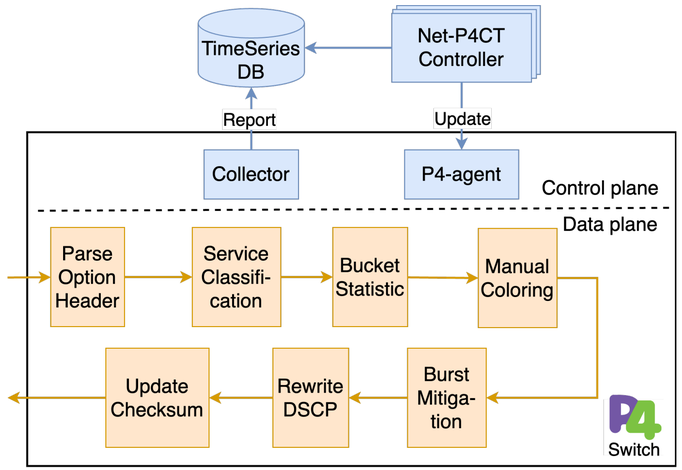
CIR = job's guaranteed bandwidth

PIR = guaranteed bandwidth + fair share

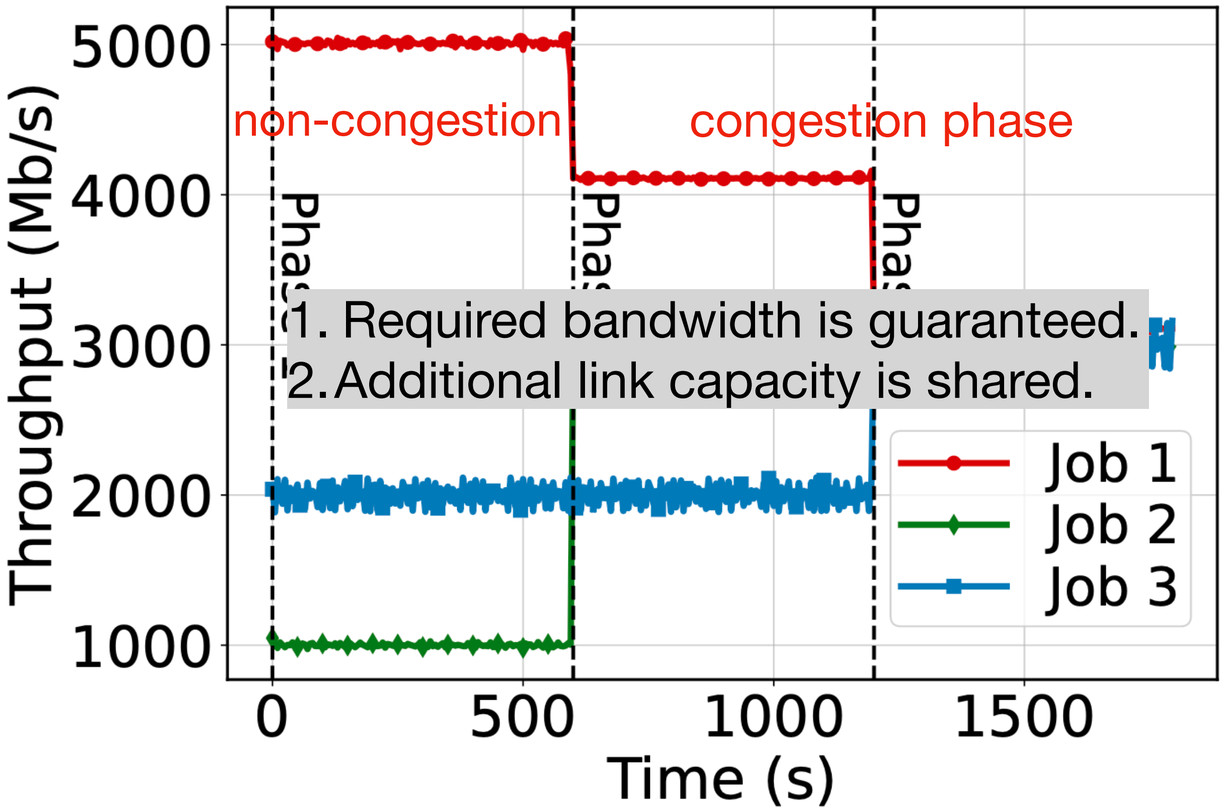
Statistics-based Policy

- Manual coloring: find the groups of flows matching the guaranteed bw and fair share
  - Hash bucket: a group of flows (5-tuple)
  - A variant of the knapsack problem based on feedback stats
- Burst mitigation: handle short-term traffic bursts
  - Green meter: traffic > guaranteed —> Red; Yellow meter: traffic > fair share —> Red

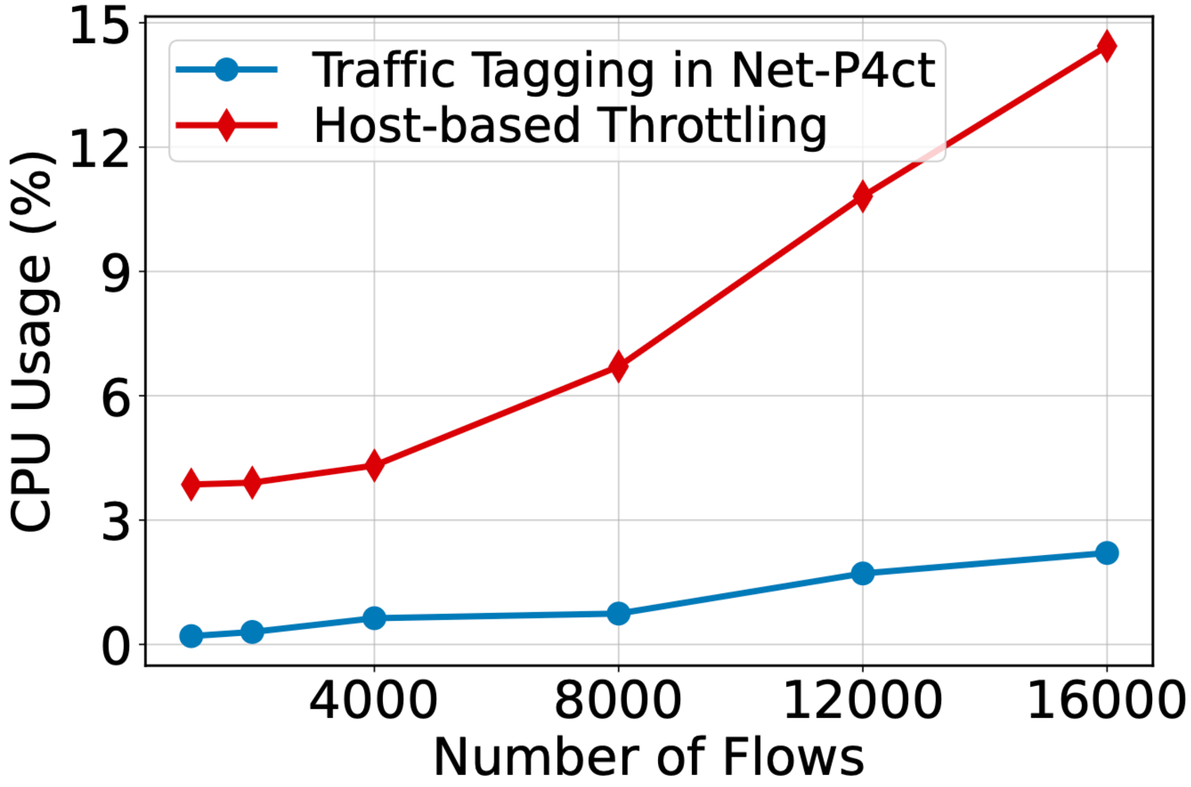




Evaluation



Bandwidth fair sharing



CPU overhead

Deployment Experience (1/2)

- Bandwidth request reviews are essential.
  - During early deployment, traffic patterns are unclear. It is crucial to review and adjust bandwidth requests.
  - If the usage is far less than the requested, we require service team to take a fixed portion of the cost.
- Cluster scaling-out
  - traffic < threshold (empirically set to 1 Gbps): redirect to 1 device for enforcement
  - traffic > threshold: distribute the policy enforcement across all P4-switches

Deployment Experience (2/2)

- Failure detection and fallback is critical
  - We deploy active probes that monitor both P4 and non-P4 paths simultaneously.
  - In the event of a P4 switch port failure or an entire P4 switch failure, the BGP routes are automatically withdrawn.
  - In the highly unlikely event of a P4 cluster failure, we use a fail-safe mechanism to ensure basic network connectivity.
- Deploying Net-P4ct outside ByteDance
  - The core concept of Net-P4ct is a general in-network bandwidth sharing mechanism
  - We are actively exploring the feasibility of porting Net-P4ct to other prominent networking silicon, such as Trident5

Conclusion

- Net-P4ct provides a generic in-network bandwidth management approach for broader coverage of WAN traffic.
- The system is work-conserving, guarantees per-service minimum bandwidth, and provides max-min fairness.
- We share our deployment experience in the real-world large-scale production network.

Contact: chenhaoran.51@bytedance.com