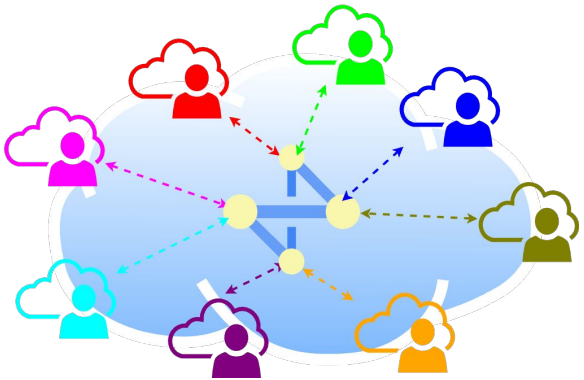
Performance Isolation in Multi-tenant Cloud Datacenters

ベノワ　ヌニャンケ



第93回 TechTrend Talk Series

vol.7

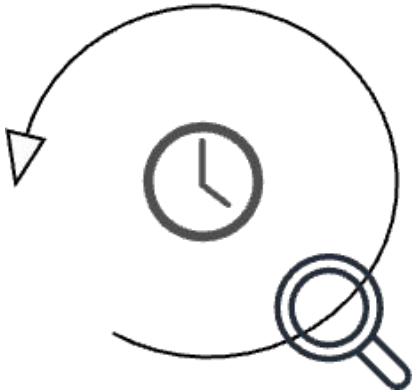
2023 / 11 / 21

Outline

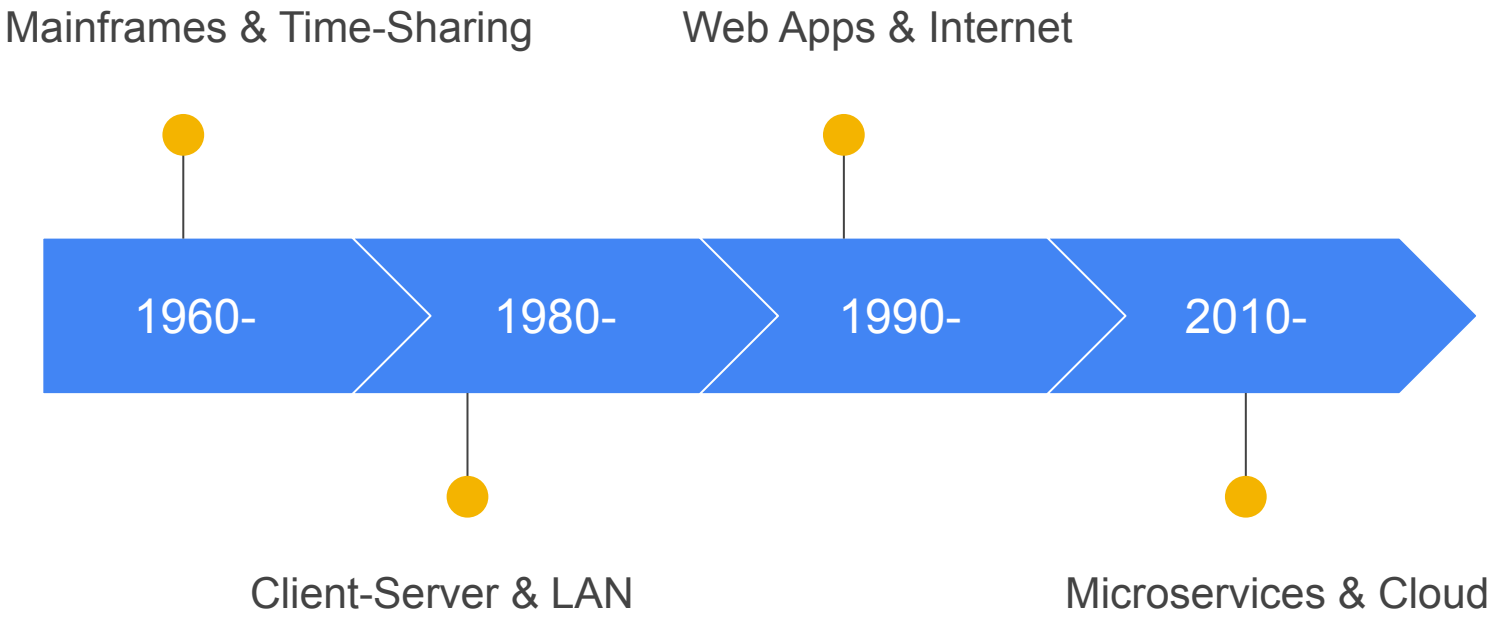
- Historical Perspective
- Multi-tenant Cloud Data Centers
- PerfIsol Initiative

Outline

- **Historical Perspective**
- Multi-tenant Cloud Data Centers
- PerfIsol Initiative

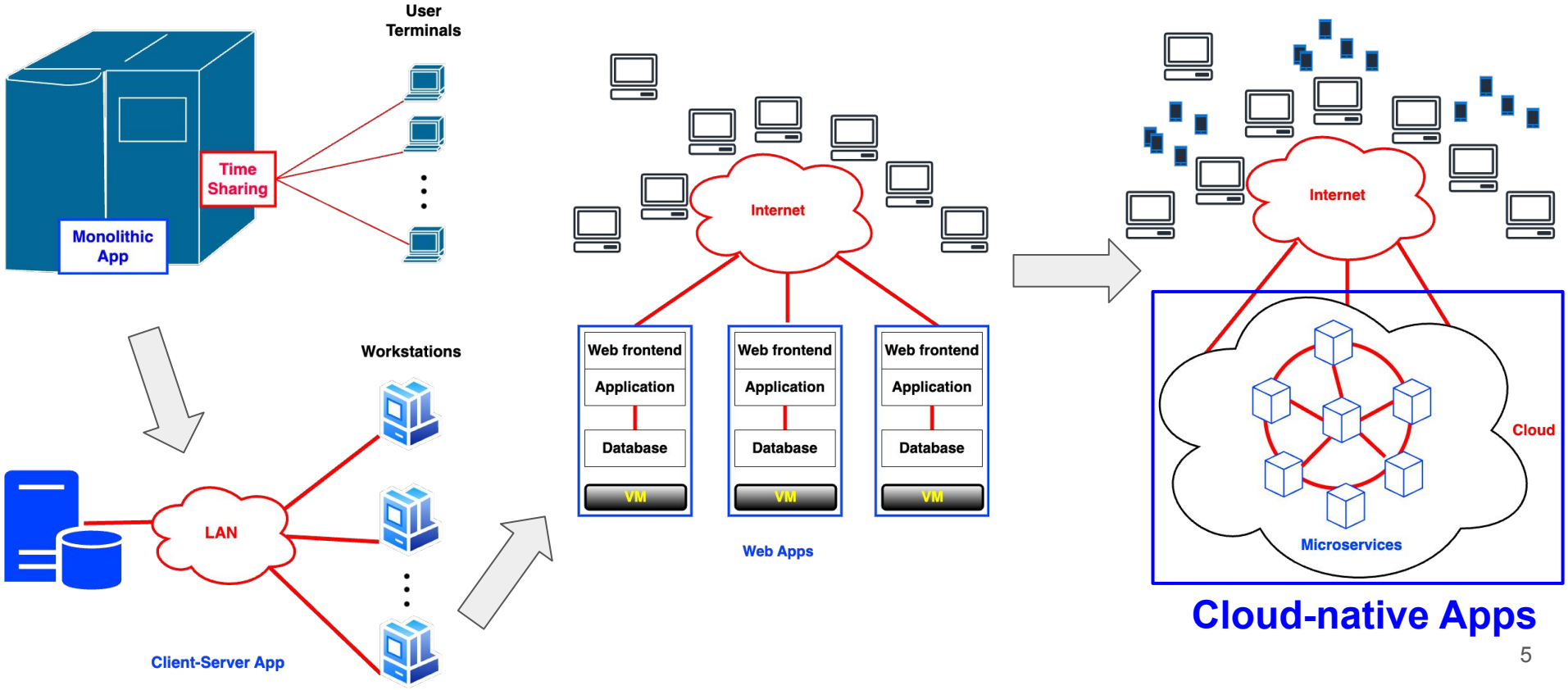


From Mainframes to Cloud: Application-Network Synergy



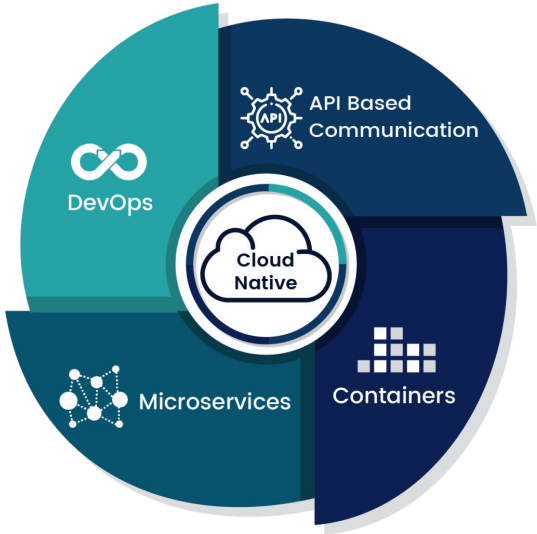
- Application evolution puts **different** and **increased challenges on the network**

Evolution of Application Architecture: Application-Network Journey



Modern Applications

- Cloud-native → **agility, velocity**
  - Microservices, using RPCs to communicate between services
  - Containers & orchestration → Kubernetes ⇐ **in the cloud**
  - Automation: DevOps, CI/CD



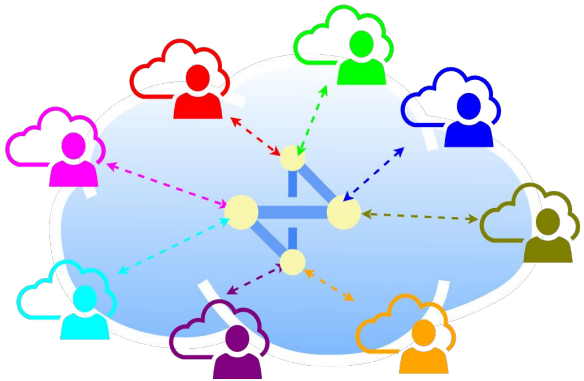
- **More services/tenants to the Cloud**

⇒ Multi-Tenant Cloud Data Centers (MTCDC) and its Networking challenges

Image source: https://blog.knoldus.com/cloud-native-future-of-software-architecture/

Outline

- Historical Perspective
- **Multi-tenant Cloud Data Centers**
- PerfIsol Initiative

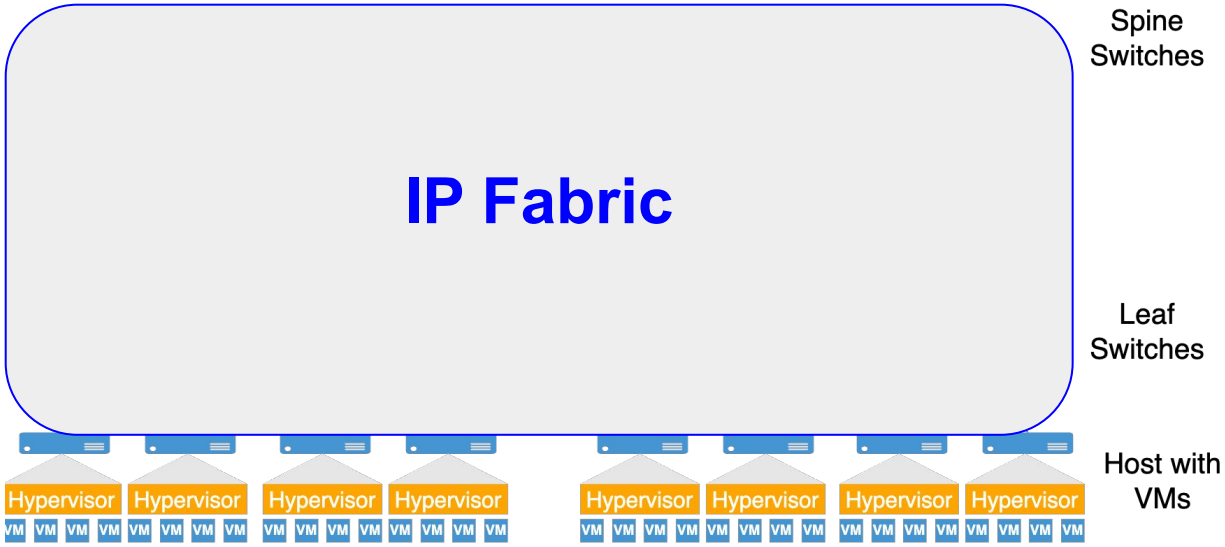


Challenges on Multi-tenant Cloud DC Networking

- **Management: flexible and efficient to operate**
- **Bandwidth: High server-to-server capacity**
  - Topology, Routing, Load Balancing
- **Network Virtualization at scale**
  - Accommodate different services/tenants simultaneously
- **Performance Guarantees**
  - **predictable** and reliable network ← efficient resource sharing

Multi-tenant Cloud Data Center Network Fabric

- Non-blocking fabric
  - High bandwidth



- Portland, VL2: Virtualization, scale [SIGCOMM 09]
- Andromeda: Network virtualization stack [NSDI 18]

The Rise of IP Fabrics and Corresponding Novel Routing Paradigms [https://www.comsoc.org/publications/ctn/rise-ip-fabrics-and-corresponding-novel-routing-paradigms-part-1] & RFC 7938

Multi-tenancy in Cloud Data Center Networks

- NVO3: Data-Center Network Virtualization over Layer 3
  - Programmatically create, provision, and manage networks completely within software
  - While leveraging the underlying physical network as the packet-forwarding backplane
- Overlay Network:
  - A virtual network
    - The separation of tenants is hidden from the underlying physical infrastructure
  - e.g., VXLAN, GENEVE ⇒ **tunnel** encapsulations

RFC 8014 “An Architecture for Data-Center Network Virtualization over Layer 3 (NVO3) ” 2016

Overlays and Underlay: Packet Journey

- UDP over IP encapsulation → L2 segment for a tenant

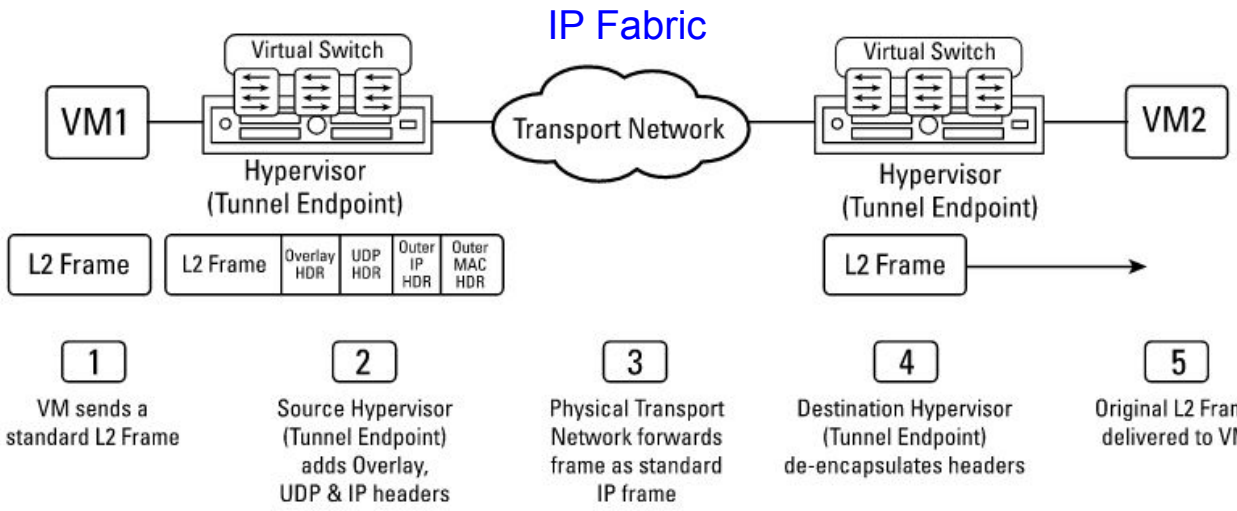


Image source: Network Virtualization For Dummies, VMware 3rd Special Edition

Logical Isolation vs. Performance Isolation

- **Overlays** offer **logical isolation only**:
  - Traffic Isolation
  - Address Isolation
- Ideally: **tenant virtual network (DC)** could support the ***illusion*** that each application is running on **its own isolated network**
  - Not satisfied by overlays only

Performance Isolation Matters

- Mechanisms ensuring tenants' resource usage doesn't impact other tenants.
- The **“illusion of feeling alone”:** each service/tenant operates as if connected to a separate physical switch
  - An app facing attack/bug does not adversely impact the performance of other apps/tenants
  - Taking into account the possibility of selfish or malicious behavior from tenants

- **Contention at shared resources in the MTCDC underlay fabric**
- **Tenants should get end-to-end guarantees → Isolate tenants across the network**

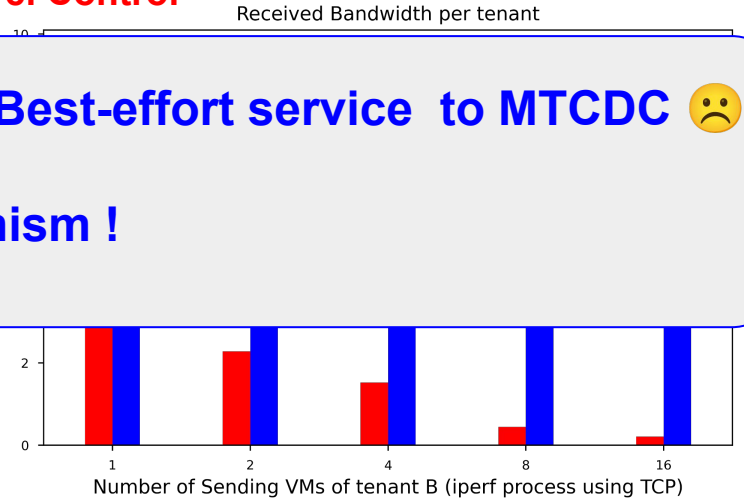
Noormohammadpour M., and Cauligi S. R. "Datacenter traffic control: Understanding techniques and tradeoffs." IEEE Communications Surveys & Tutorials 20.2 (2017)

How to share the cloud resources ?

Resources Sharing (1/2)

- Resources:
  - ~~Compute~~, ~~Memory,~~ Network ⇒ **links bandwidth** & **buffers** of switch/vswitch (queued packets)
- Traditional resource allocation on *packet-switched network*
  - **Queuing Disciplines**: e.g. FIFO → **Packet-level Control**

- **Packet-Level & Flow-Level Controls ⇒ Best-effort service to MTCDC** ☹
- **We need a Tenant-level Control mechanism !**



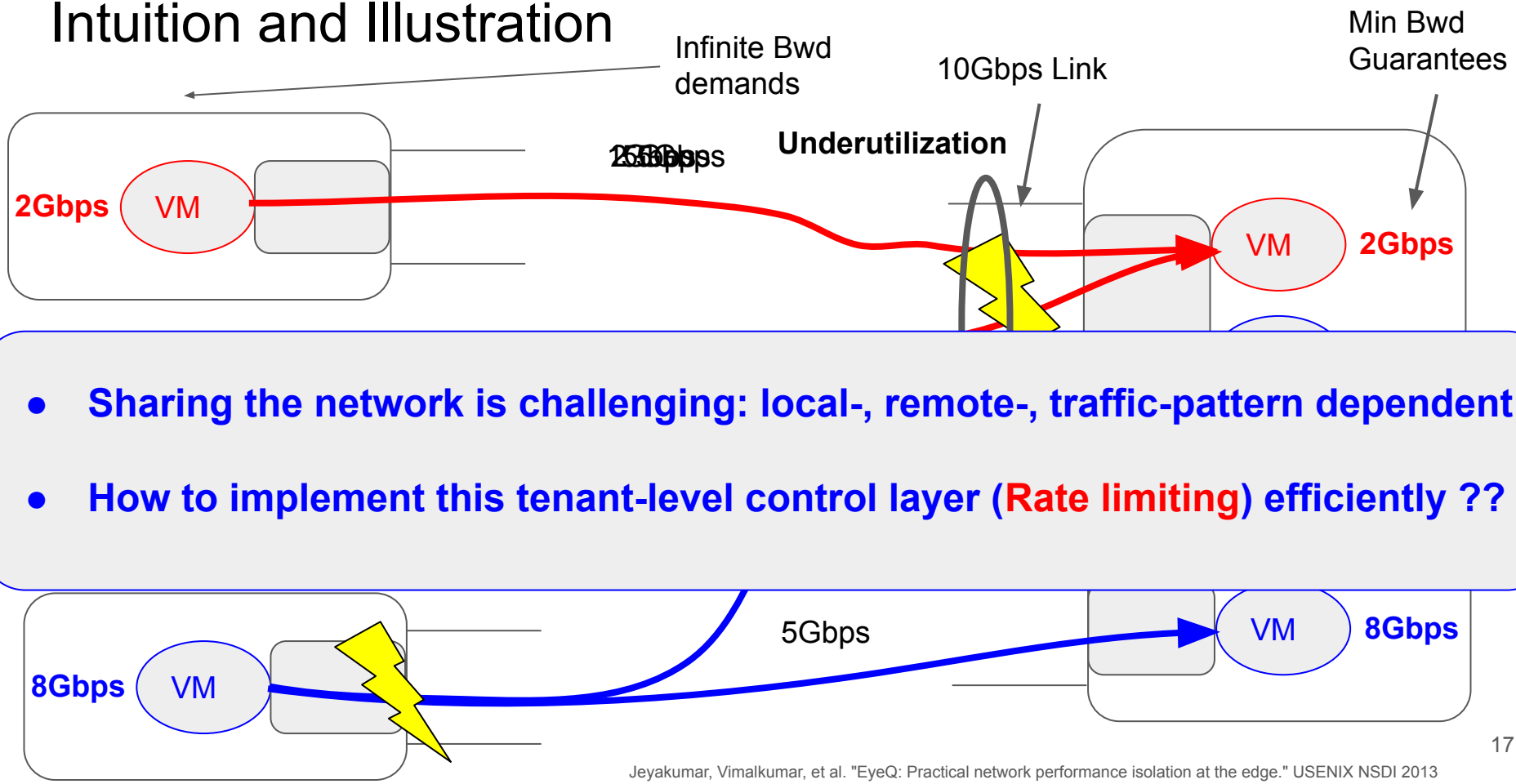
Computer Networks: A Systems Approach [https://book.systemsapproach.org/congestion/problem.html]

Resources Sharing (2/2)

- **Tenant-level Control Layer** to cope with the multi-tenancy specificities
  - With SLA/SLO: pricing, bandwidth guarantees
  - Scale: number of tenants, VMs, workloads, physical topology
  - Additional network layer:
    - Overlays → the virtual networks for tenants

* SLA = Service-Level Agreement
* SLO = service-Level Objective

Intuition and Illustration



- **Sharing the network is challenging: local-, remote-, traffic-pattern dependent**
- **How to implement this tenant-level control layer (Rate limiting) efficiently ??**

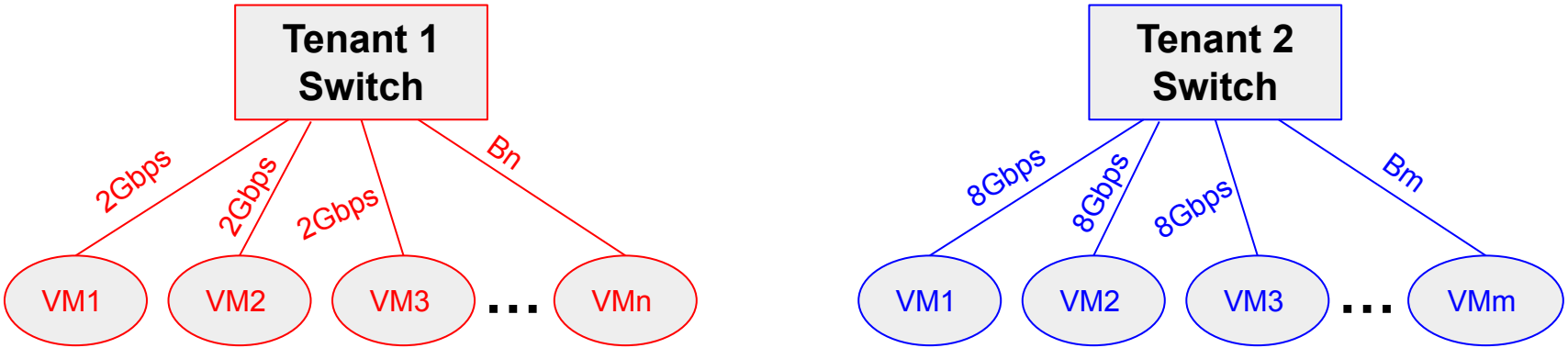
Jeyakumar, Vimalkumar, et al. "EyeQ: Practical network performance isolation at the edge." USENIX NSDI 2013

Tenant-level Control Layer Design

- **Service Model (Abstractions)**
  - How tenants express SLAs / SLOs ?
- **Well Defined Goals**
  - Establish measurable objectives for performance isolation
  - Navigating the objectives tradeoff space ⇒ resource sharing policies
- **Flexible and Efficient Implementation**

Service Model

- Tenant specifies performance requirements as **bandwidth capacity** of the vNIC



Duffield, Nick G., et al. "A flexible model for resource management in virtual private networks." ACM SIGCOMM 1999

Performance Isolation Goals

- Predictable performance
- SLOs guarantees in regard to SLAs
- Optimal resource utilization
- Fair QoS impact during overload

Tenant-Level Control Layer: Existing Works

- EyeQ
  - A Sender EyeQ Module (SEM) and Receiver EyeQ Module (REM) at every end-host → WFQ
  - Convergence time (~ 5-10ms), qdisc implemented as Linux kernel (v. 3.0.0) module

Tenant-level Control (Rate Limiting) Design Considerations

- **Guaranty Granularities:**
  - per-tenant, per-VM, per-flow ?
- **SLA and Pricing:**
  - flat-price vs. fixed + dynamic (for bandwidth allocation beyond the min.)
- **Nice vs disruptive tenants consideration**
- **Deployment/Implementation: Host/Switch capabilities**

Outline

- Historical Perspective
- Multi-tenant Cloud Data Centers
- **PerfIsol Initiative**

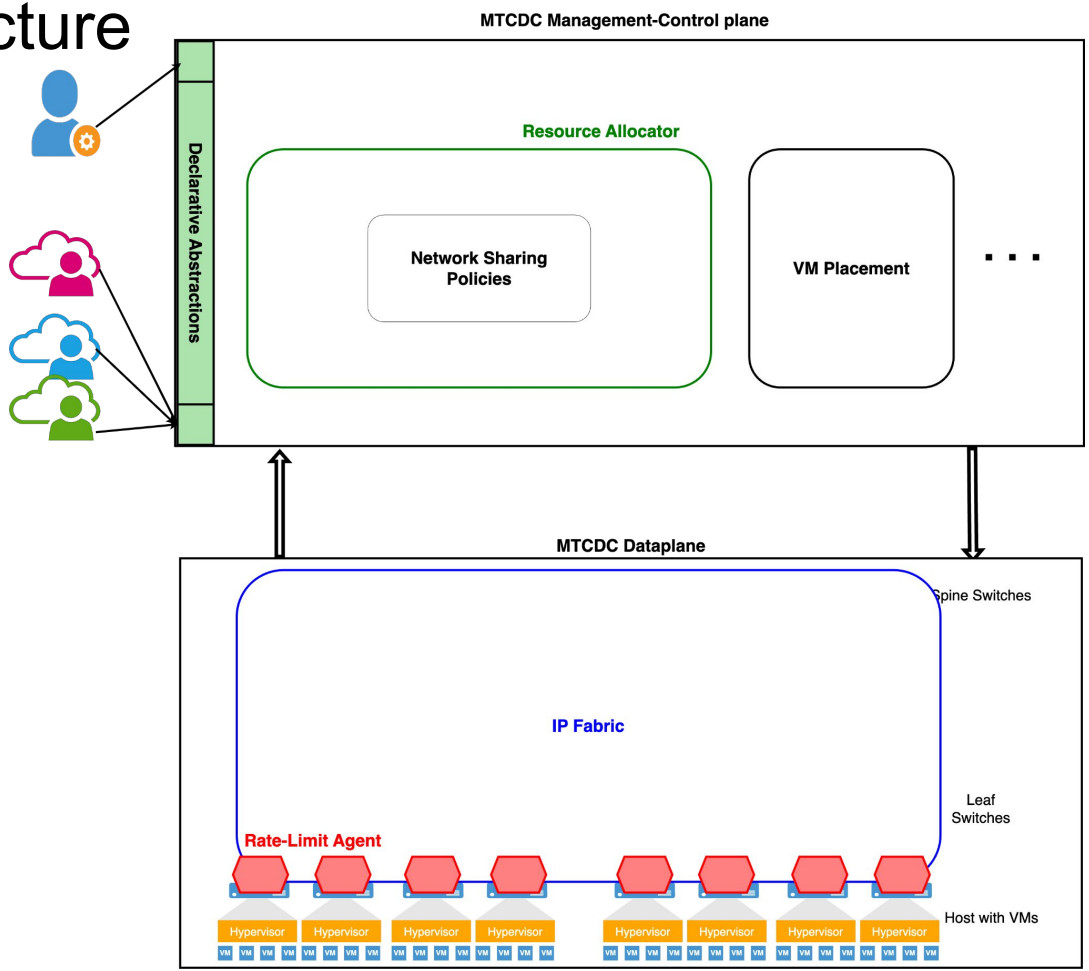
PerfIsol Research Goal

- **Complete system design comprising**:
  - Well defined service abstractions to tenants
  - Sharing models related to SLA/SLO
  - Low level operations in the multi-tenant data plane

As a unified framework

- **Without SLA/SLO (payment) Considerations**
- **Performance Isolation (fairness) ⇒ Abstract concept hard to justify**

PerflSol System Architecture



Rate-Limit Agent: Motivation

- Traffic Control (Rate limiting) at end-host
  - Typically implemented in software in the **kernel networking stack**
  - Some Limitations:
    - Non-precise rate limiting (for Performance isolation)
    - Configurable Qdisc (rate/ceil != minBwd/maxBwd)
      - Not directly programmable

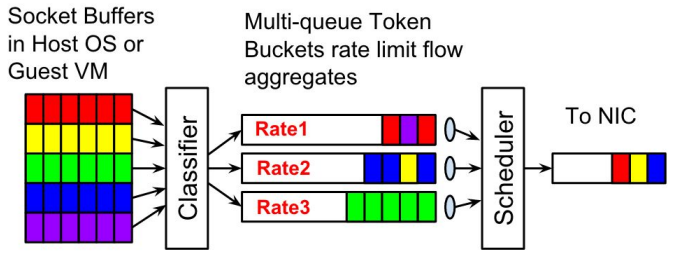
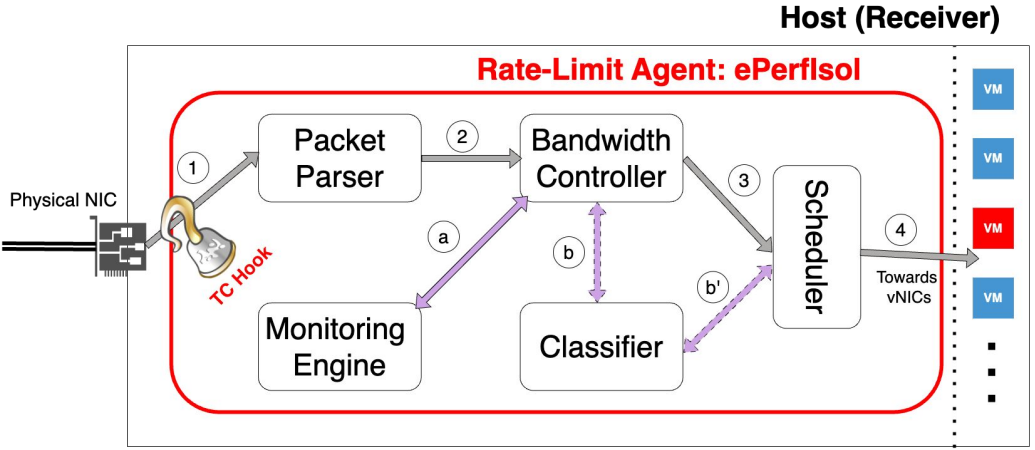


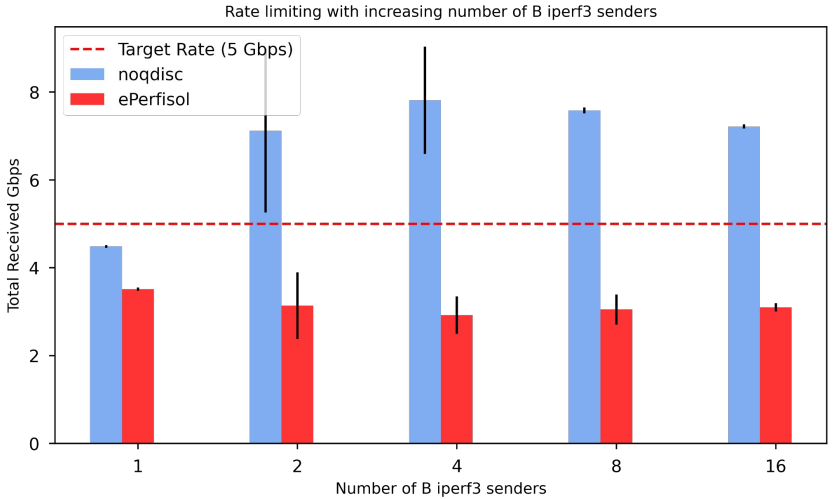
Figure 1: Token Bucket Architecture: pre-filtering with multiple token bucket queues.

- **Need for simplified and programmable Qdisc**
- **Leverage eBPF Kernel programmability capability**

ePerfIsol: eBPF-based Rate-Limit Agent (programmable qdisc)

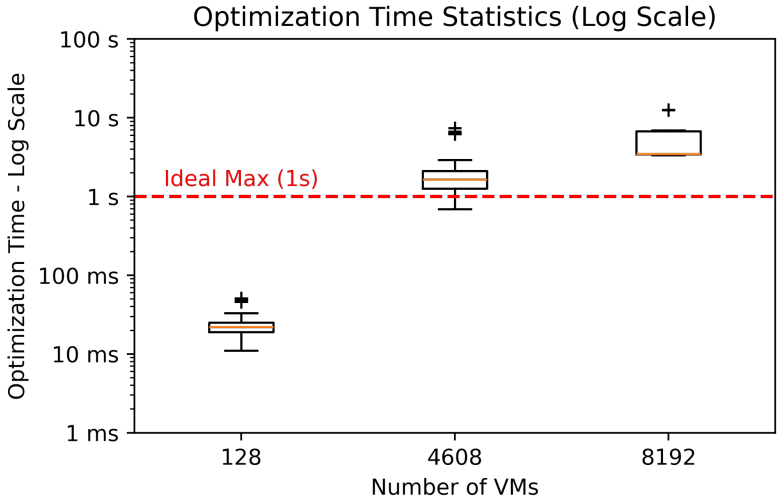
- Tenant-level mechanism
  - Counting
  - Thresholding → packet drops





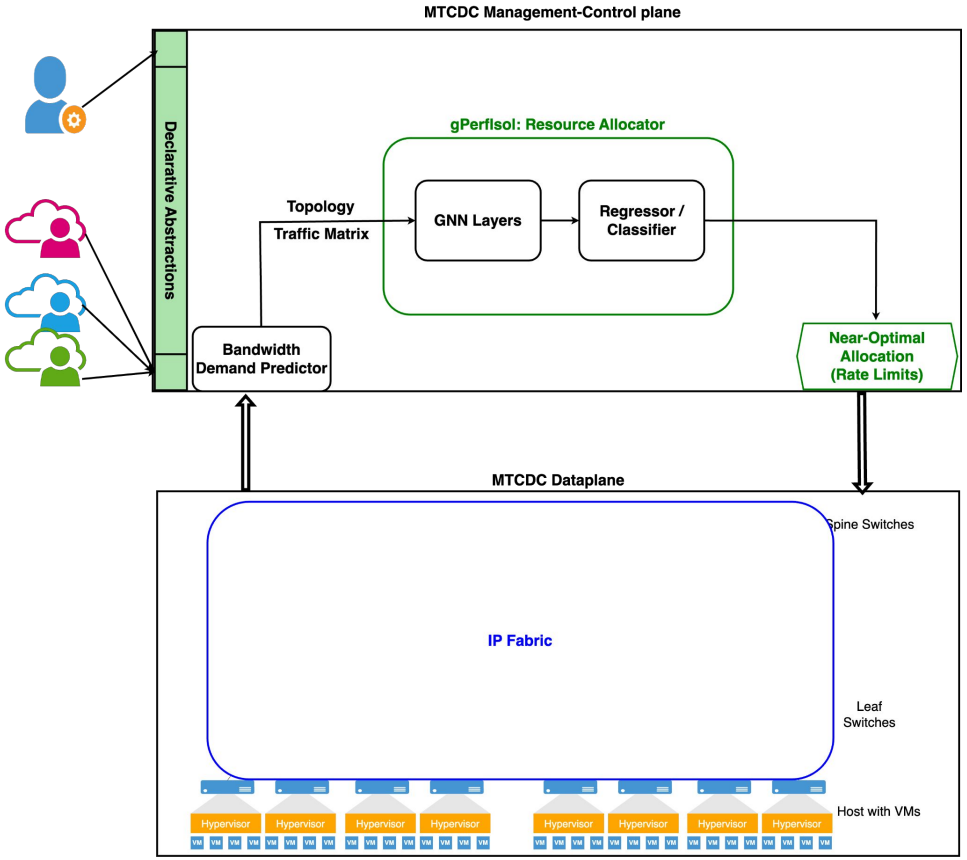
Resource Allocation: Motivation

- How to find the optimal rates consumed by the rate-limiters ?
  - Linear programming
  - Multi-commodity flow problem
    - MTCDC topology + VMs-pair traffic matrix
- Computing time Scalability issues
  - with huge number of nodes (**VMs** in MTCDCs)

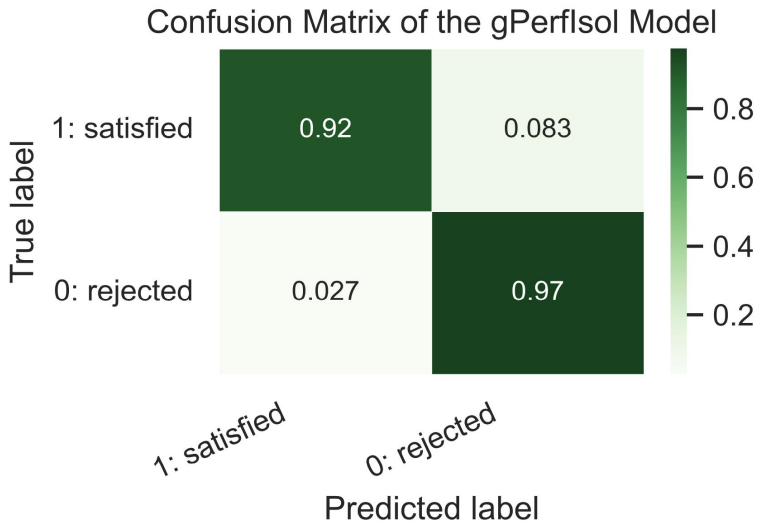


- **Need of accelerated rate-limits allocation mechanism**
- **Leverage Neural Network on graphs (GNNs) learning ability**

gPerfIsol: GNN-based Rate-Limits Allocation



- Admission Control
  - Accept or refuse a VM-pair demand



Summary (Takeaways)

- Application evolution puts different and **increased challenges on the network**
  - monoliths / mainframes → client-server / LAN → Web / Internet → microservices / MT-Cloud
- **Performance isolation** is needed to address **tenants interference** in MTCDCs
  - How tenant virtual networks **share** the underlying physical infrastructure ?
- PerfIsol Approach for performance isolation
  - Unified framework (management & data planes) for optimal resource sharing for multi-tenancy
  - **Rate Limiting**: allocation with gPerfIsol (GNN) & ePerfIsol (eBPF) for the execution

ご清聴ありがとうございました