CLOUD COMPUTING
Data Center Management

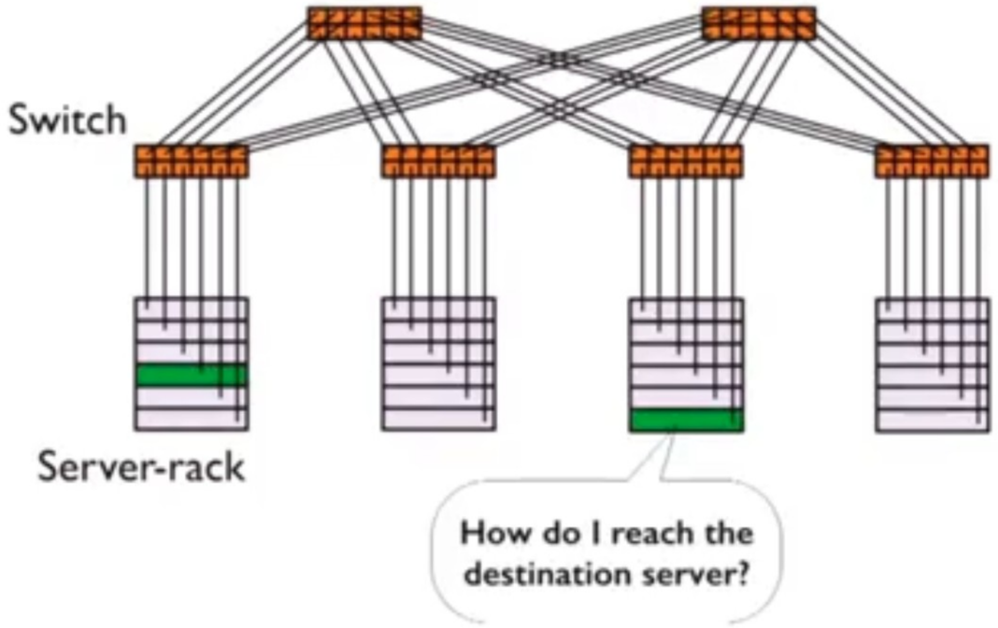
Zeinab Zali
Isfahan University Of Technology

References: -Cloud computing: Theory and practice, Chapter 5
-Cloud Networking course, Illinois university
-Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google’s Datacenter
-Network Virtualization in Multi-tenant Data Centers

Routing in Clouds

Finding Path

- For any source and destination server in a DC, the source wonders what path it can use to reach the destination
  - some routing information in switches are required to be able to find those paths



Traditional routing protocols

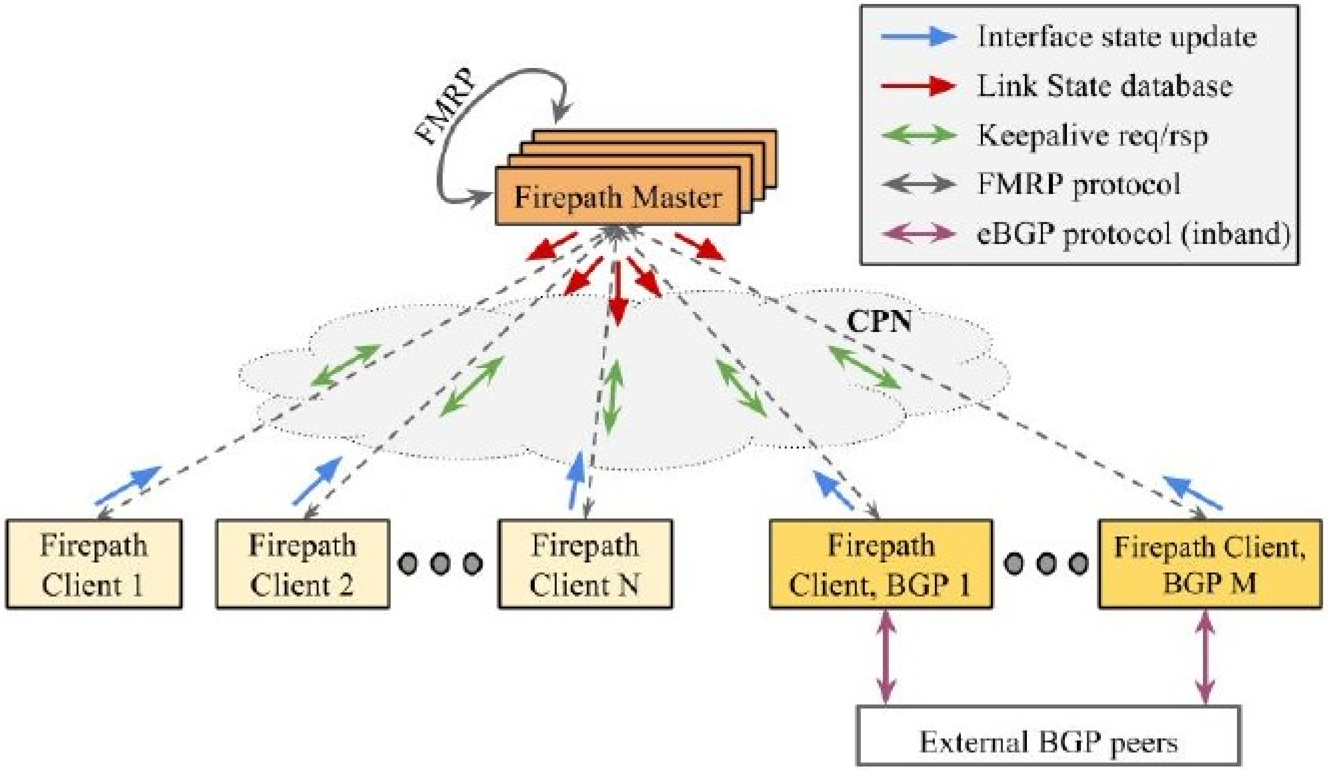
- Existing routing protocols did not at the time have good support for multipath, equal-cost forwarding (OSPF, BGP)
- The protocols overhead of running broadcast-based routing protocols across fabrics of the scale Data Centers is high
  - targeting with hundreds or even thousands of switching elements
  - Scaling techniques like OSPF Areas appeared hard to configure and to reason about

Google Datacenter Routing(I)

- Building a fabric with thousands of cables invariably leads to multiple cabling errors
- correctly cabled links may be re-connected incorrectly after maintenance
- Allowing traffic to use a miscabled link can lead to forwarding loops
- Links that fail unidirectionally or develop high packet error rates should also be avoided and scheduled for replacement.
- To address these issues, google developed Neighbor Discovery (ND), an online liveness and peer correctness checking protocol

Google Datacenter Routing(II)

- Google approach was driven by the need to route across a largely static topology with massive multipath



Centralized link state

- The switches learn actual configuration and local link state through pair-wise neighbor discovery
- Each switch exchange its local view of connectivity with a centralized master, which redistributes global link state to all switches
- Switches locally calculate forwarding tables based on this current view of network topology
  - On receiving an LSD update, each client computes shortest path forwarding with Equal-Cost MultiPath (ECMP) and programs the hardware forwarding tables local to its switch

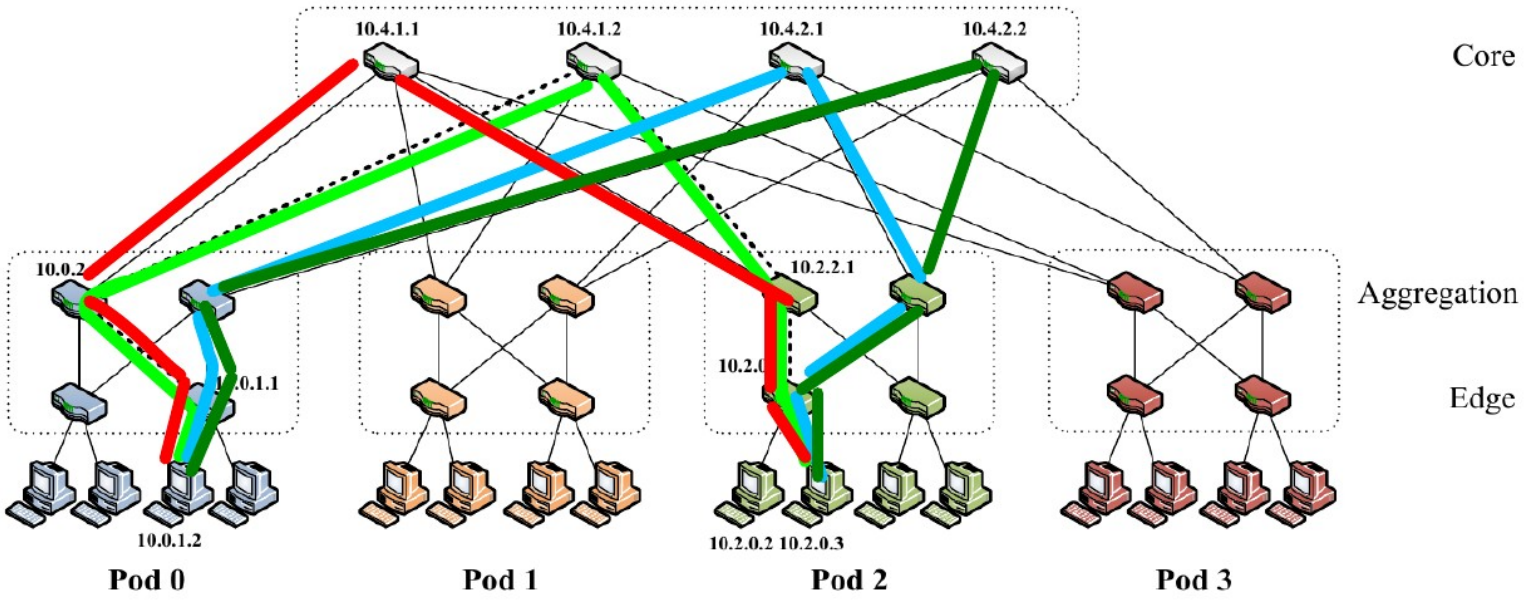
Neighbor Discovery

- Neighbor Discovery (ND) uses the configured view of cluster topology together with a switch’s local ID to determine the expected peer IDs of its local ports
- It regularly exchanges its local port ID, expected peer port ID, discovered peer port ID, and link error signal
  - Doing so allows ND on both ends of a link to verify correct cabling

Congestion Control

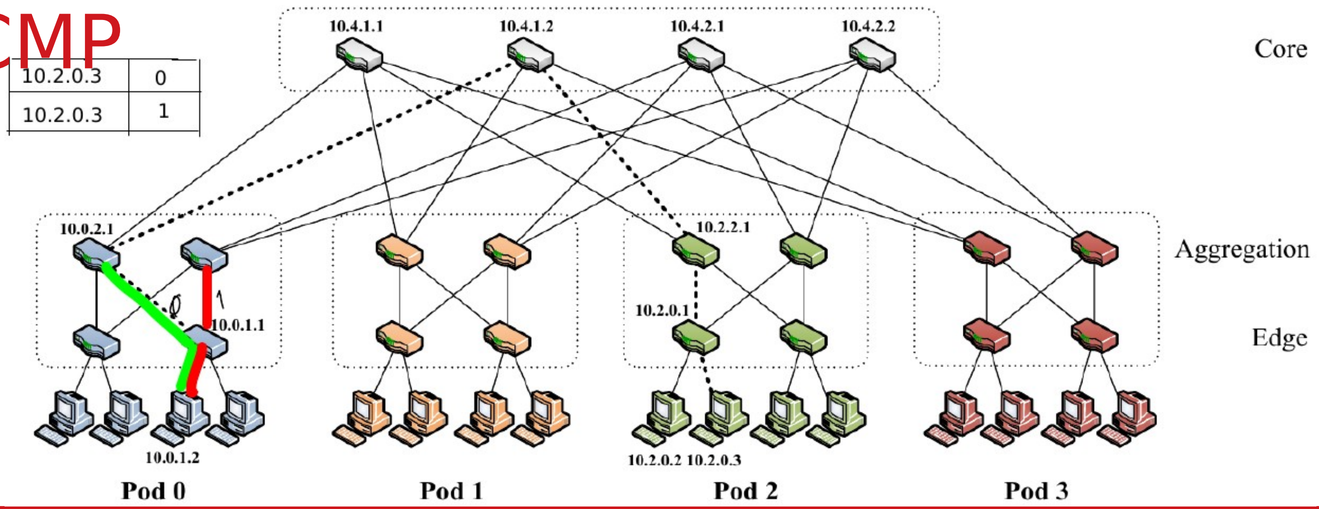
Multipath routing

- The goal of traffic engineering is to minimize network congestion.
- So we want to spread traffic throughout the network, while ensuring that none of the paths are congested, or at least most of the paths are not congested



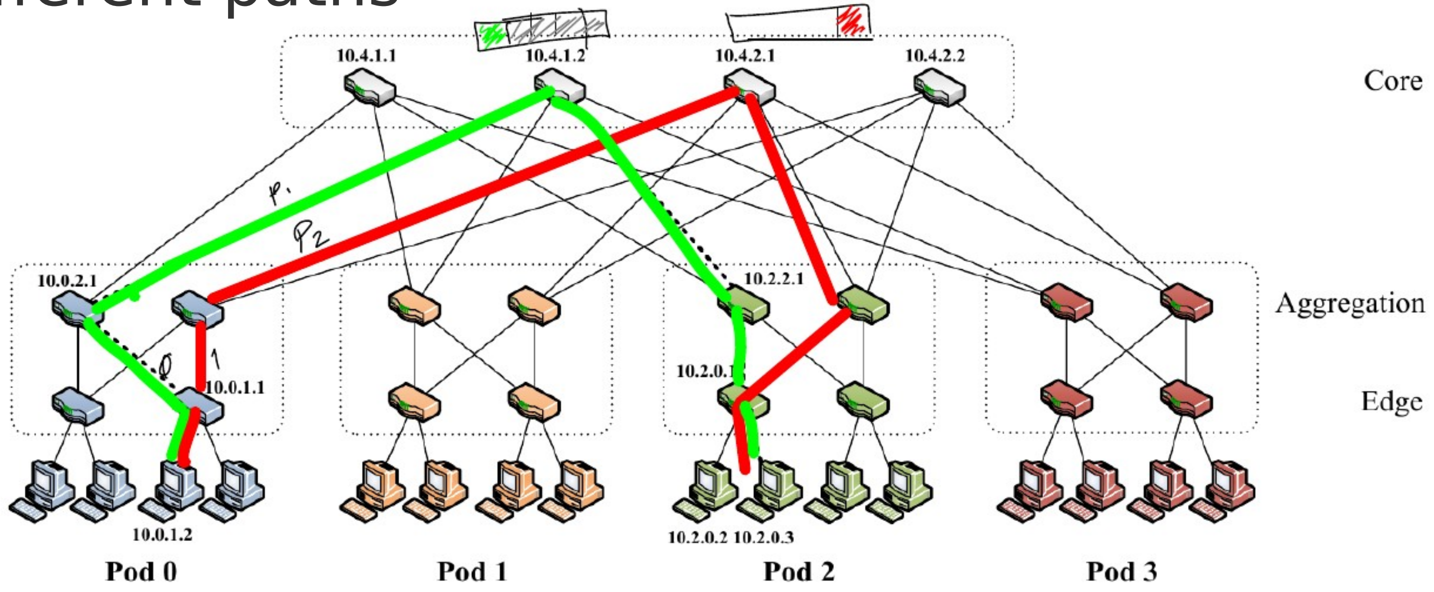
Equal Cost Multi Path

- In DC networks, there can be a large number of flows which are very short
  - So imagine that you are trying to gather information about the network's current state, the traffic's current state, and that state is changing quite quickly
- Base solution: Randomized spreading of traffic with ECMP



Randomized spreading

- making individual decisions for different packets, and those packets end up with this random selection on different paths, can be a problem for TCP
  - The packets may be reordered when arriving the destination because of different delays on different paths



Flow level hashing

- We can decide which of the random choices to make based on the hash of the packet header
  - flow-level hashing
- However, randomization will sometimes lead to imbalance in flow assignments.
  - This can cause long lasting congestion if multiple long lasting data flows are mapped onto the same path

flowlet

- If the gap between groups of packets is larger than the conservative estimate of the latency difference between the two paths, we can send the the packet groups on different paths without risk of re-ordering.
- A flowlet is a burst of packets that is separated in time from other bursts by a sufficient gap — called the "flowlet timeout"
- The idea of flowlet method is to pick a path at random for each flowlet

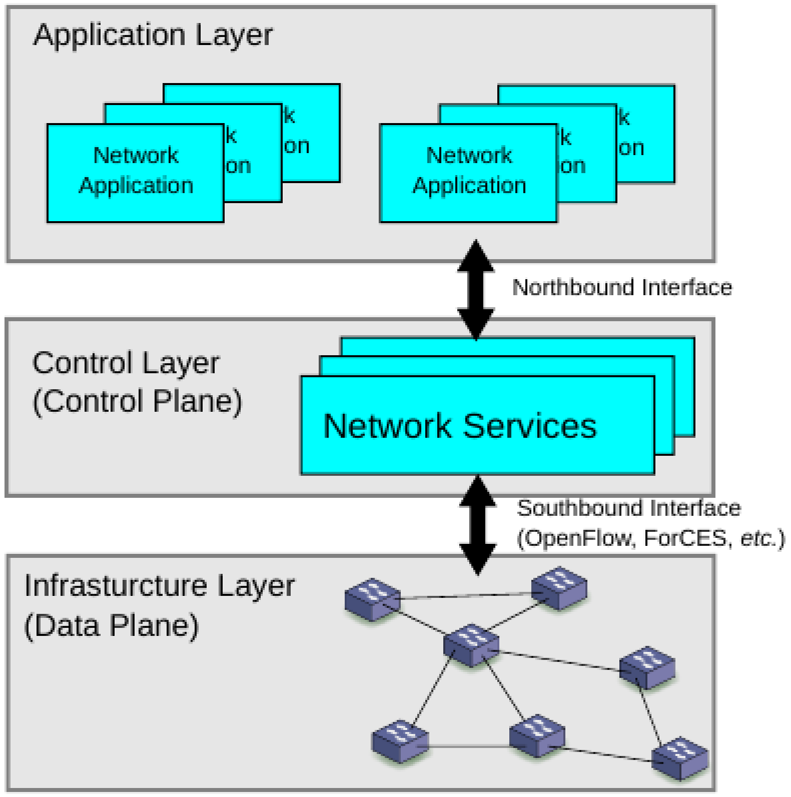
Real-time Congestion control

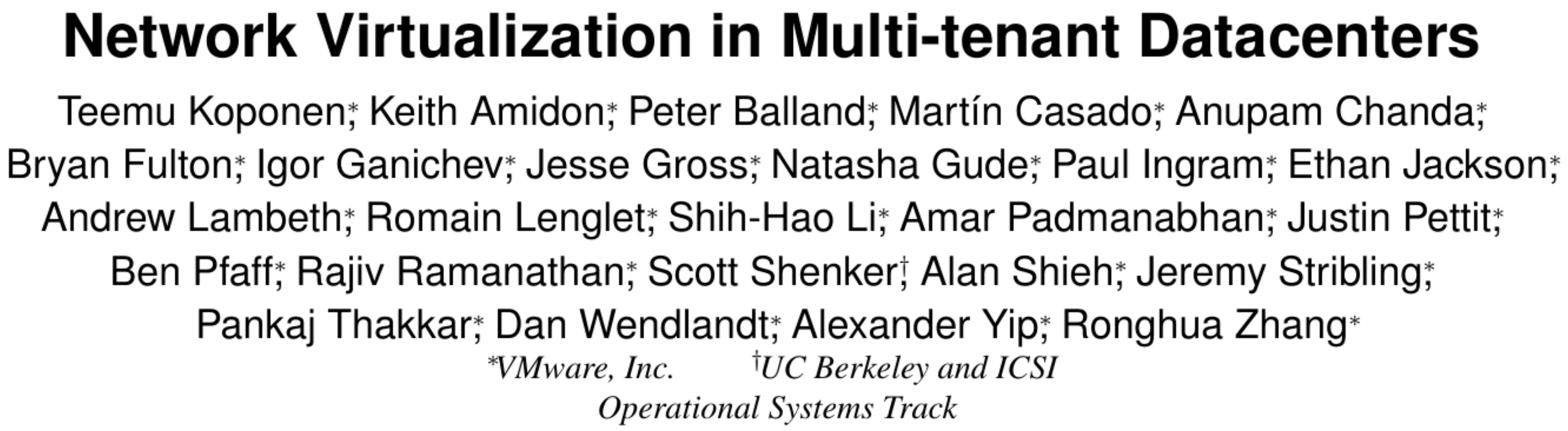
- The key idea is to monitor congestion on paths in real time, and pick the least congested path.
- This requires maintaining a table at every lead switch for every destination switch.
- There you keep track of the extent of congestion on each of the parts within these two switches.
  - One solution is to piggy-backed each packet between two switches with the delay information between the hops

SDN and Clouds

Software Defined Network

- Separating the control plane from data plane





Network Virtualization in Multi-tenant Datacenters

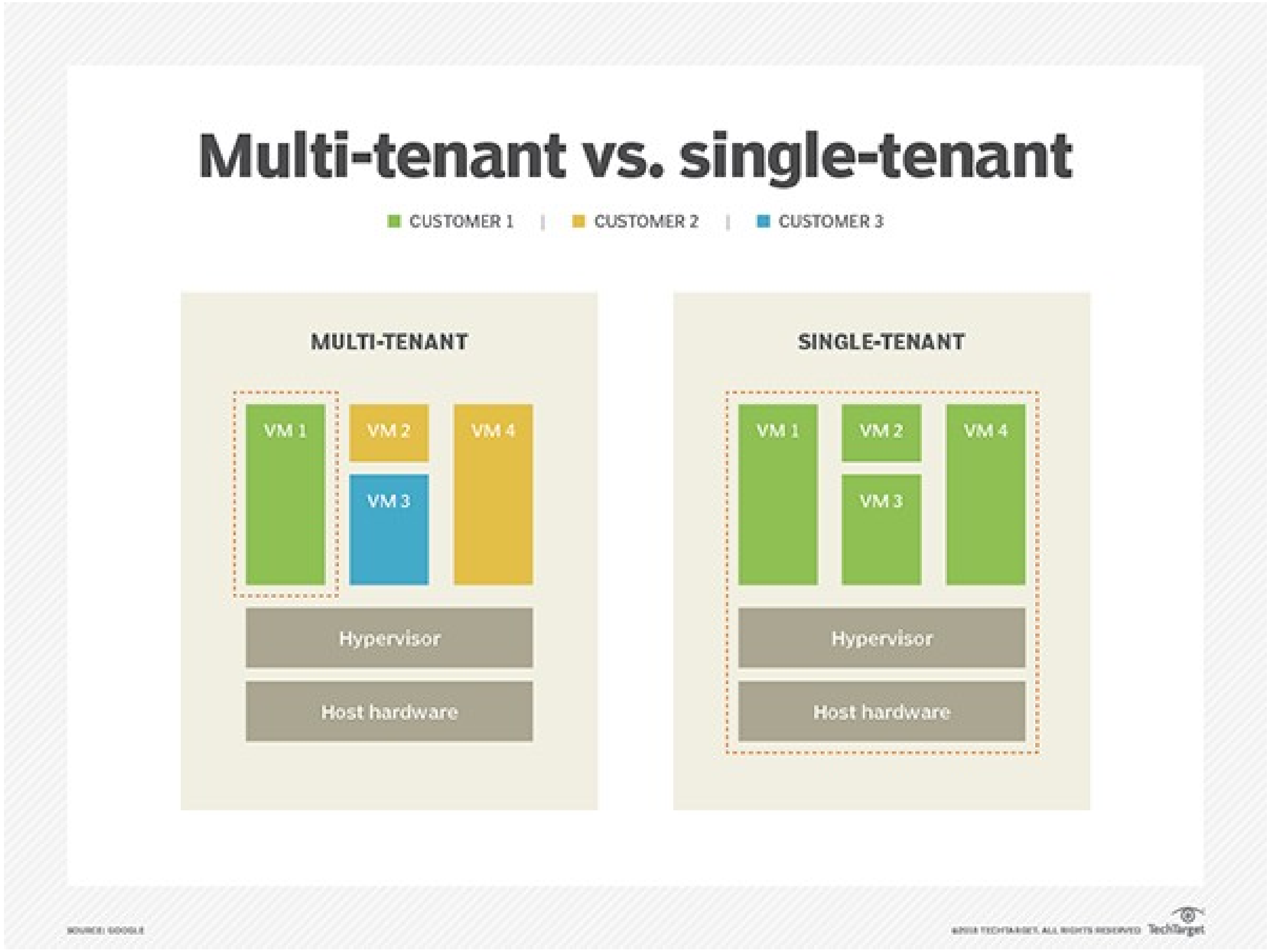
Teemu Koponen[*], Keith Amidon[*], Peter Balland[*], Martín Casado[*], Anupam Chanda[*], Bryan Fulton[*], Igor Ganichev[*], Jesse Gross[*], Natasha Gude[*], Paul Ingram[*], Ethan Jackson[*], Andrew Lambeth[*], Romain Lenglet[*], Shih-Hao Li[*], Amar Padmanabhan[*], Justin Pettit[*], Ben Pfaff[*], Rajiv Ramanathan[*], Scott Shenker[†], Alan Shieh[*], Jeremy Stribling[*], Pankaj Thakkar[*], Dan Wendlandt[*], Alexander Yip[*], Ronghua Zhang[*]

[*]VMware, Inc.   [†]UC Berkeley and ICSI

Operational Systems Track

- Product outcomes:
  - Nicira: a network virtualization company which is now a key component of VMware's Software Defined Data Center strategy
  - NSX Data Center: is the network virtualization and security platform that enables the virtual cloud network, a software-defined approach to networking that extends across data centers, clouds, and application frameworks

multi-tenant datacenter


Multi-tenant vs. single-tenant
🟩 CUSTOMER 1 | 🟨 CUSTOMER 2 | 🟦 CUSTOMER 3
MULTI-TENANT
VM 1
VM 2
VM 3
VM 4
Hypervisor
Host hardware
SINGLE-TENANT
VM 1
VM 2
VM 3
VM 4
Hypervisor
Host hardware
SOURCE: GOOGLE
©2018 TECHTARGET. ALL RIGHTS RESERVED TechTarget

MTD challenges (I)

- Different workloads require different network topologies and services
  - Traditional enterprise workloads using service discovery protocols often require flat L2
  - large analytics workloads require L3
  - web services often require multiple tiers
  - other applications depend on L4-L7 services

MTD challenges (II)

- It is difficult for a single physical topology to support the configuration requirements of all of the workloads of an organization
  - as a result, the organization must build multiple physical networks

Network Virtualization

- Virtualized workloads today operate in the same address space as the physical network
  - The network should be virtualized as well as the computations
- Different Virtual Networks traditionally configured on a box-by-box basis
  - configuration and management of these virtual networks is ideally should done through global abstractions rather than box-by-box configuration

NVP Architecture

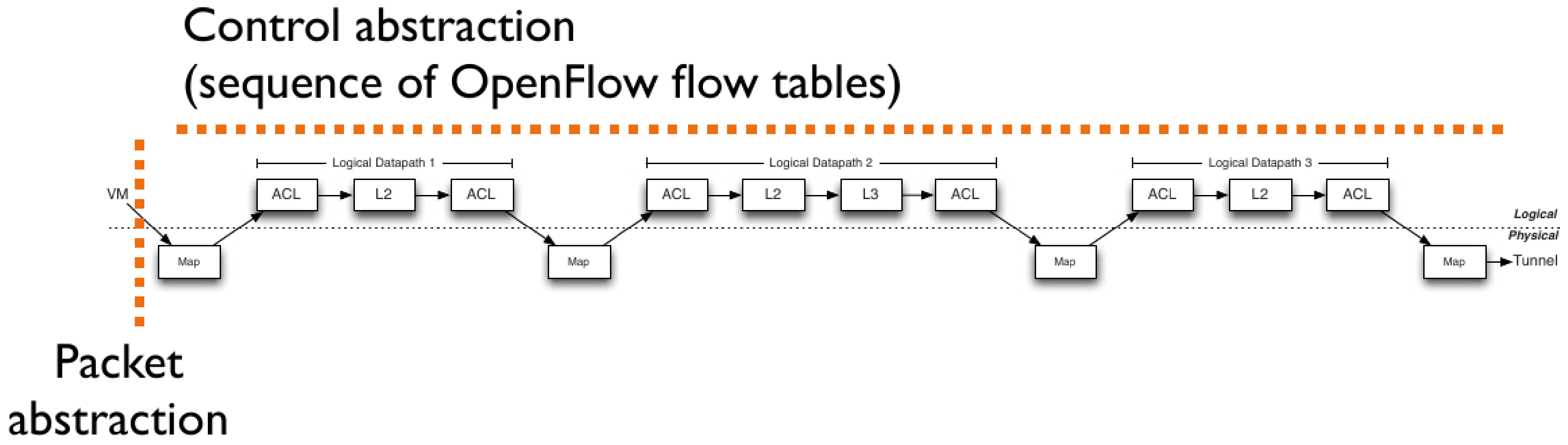
- MTDs have a set of hosts connected by a physical network
  - Each host has multiple VMs supported by the host’s hypervisor.
  - Each host hypervisor has an internal software virtual switch that accepts packets from these local Vms and forwards them to another local VM or another host hypervisor.
- NVP architecture is based on a network hypervisor that provides the right network virtualization abstractions

NVP

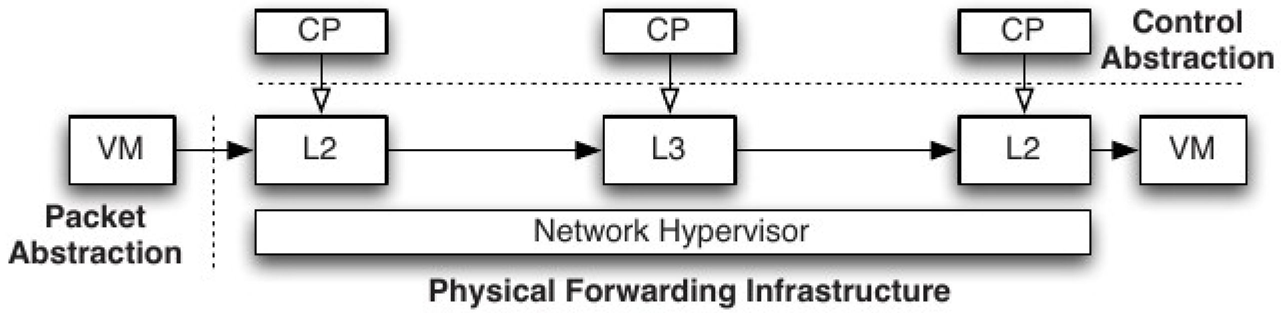
- NVP explicitly builds a network virtualization layer that separates an arbitrary network service from an arbitrary physical network
- A tenant interacts with a network in two ways:
  - tenant’s VMs send packets
  - tenant configures the network elements forwarding these packets.
- In configuring, tenants can access tenant-specific control planes that take switch, routing, and security configurations similar to modern switches and routers, translating them into low-level packet forwarding instructions.

NVP abstractions

- Packet abstraction: This abstraction must enable packets sent by endpoints in the MTD to be given the same switching, routing and filtering service they would have in the tenant’s home network
- Control abstraction: This abstraction must allow tenants to define a set of logical network elements that they can configure (through their control planes) as they would physical network elements.



Logical Data Path



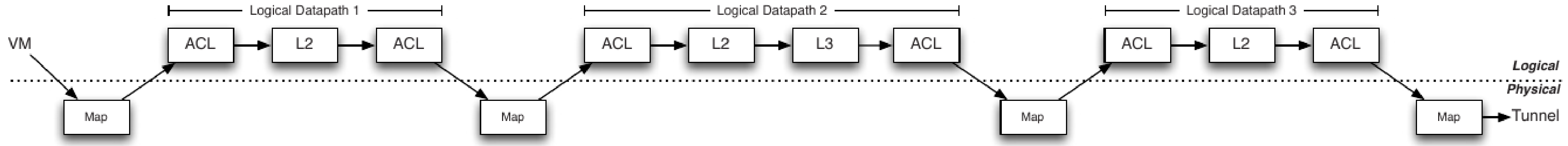
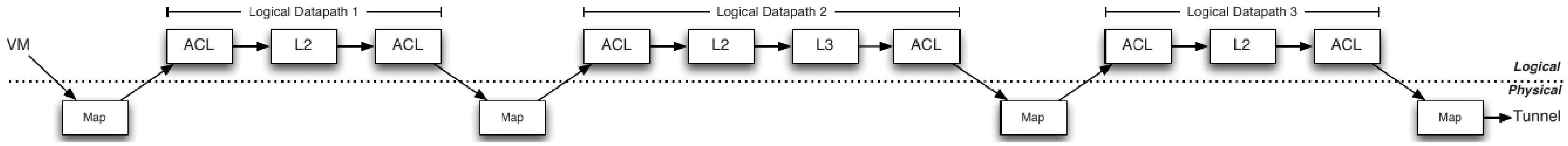


Figure 4: Processing steps of a packet traversing through two logical switches interconnected by a logical router (in the middle). Physical flows prepare for the logical traversal by loading metadata registers: first, the tunnel header or source VM identity is *mapped* to the first logical datapath. After each logical datapath, the logical forwarding decision is mapped to the next logical hop. The last logical decision is mapped to tunnel headers.

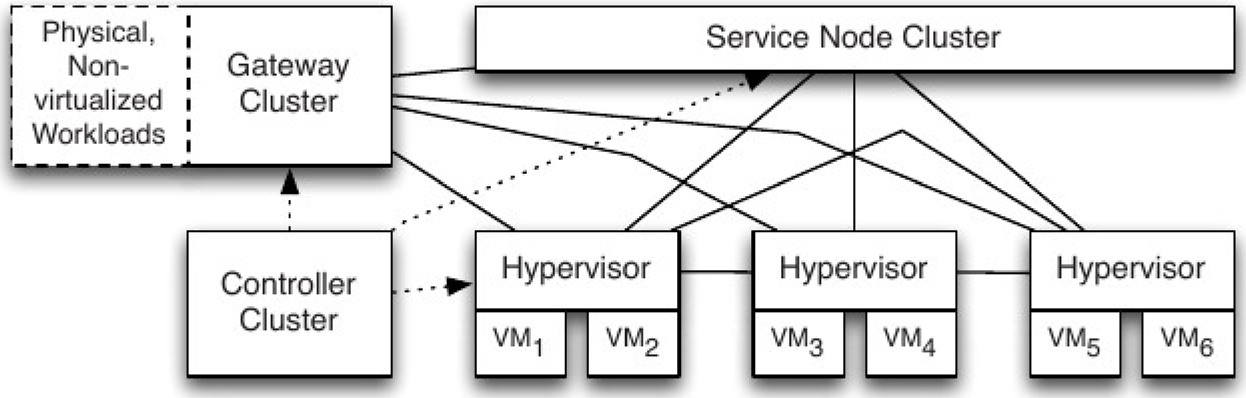
Logical Data Path

- The logical datapaths is implemented in the software virtual switches on each host, leveraging a set of tunnels between every pair of host-hypervisors
  - So the physical network sees nothing other than what appears to be ordinary IP traffic between the physical hosts
  - The receiving host hypervisor decapsulates the packet and sends it to the destination VM



Logical Data Path

- Some tenants want to interconnect their logical network with their existing physical one
- This is done via gateway appliances
  - all traffic from the physical network goes to the host hypervisor through this gateway appliance, and then can be controlled by NVP


Physical, Non-virtualized Workloads
Gateway Cluster
Service Node Cluster
Controller Cluster
Hypervisor
Hypervisor
Hypervisor
VM_1
VM_2
VM_3
VM_4
VM_5
VM_6