

## UNIT IV

# Networking for Cloud Computing

## 1. Introduction to Cloud Networking

Cloud networking is the infrastructure that enables connectivity within, into, and out of cloud environments. It serves as the fabric that links end-users to cloud applications, and connects thousands of physical servers inside data centers to form a unified computing pool.

Without highly optimized networks, cloud scalability, agility, and on-demand resource provisioning would be impossible. Cloud networks must handle massive data throughput with minimal delay while dynamically adapting to shifting virtual workloads.

## 2. Overview of the Data Center Network (DCN) Environment

A Data Center Network (DCN) is the structural arrangement of network devices (switches, routers, and cables) that connect all resources inside a data center.

### Traditional 3-Tier Architecture

Historically, data centers used a hierarchical three-layer model designed primarily for **North-South traffic** (data moving out of the data center to the internet/client).

- **Core Layer:** The backbone of the network. It switches packets as fast as possible between high-speed routing modules and connects the data center to external networks (ISP).
- **Aggregation (Distribution) Layer:** Acts as an intermediary boundary. It implements policies, security functions (firewalls, intrusion detection), and routes traffic between different subnets.
- **Access Layer:** The edge of the network. It physically connects the servers (racks) to the network via **Top-of-Rack (ToR)** switches.

### Modern Leaf-Spine Architecture (2-Tier Clos Network)

Modern cloud computing triggers massive amounts of **East-West traffic** (data moving between servers *inside* the data center, such as database replication or distributed MapReduce tasks). The 3-tier model created bottlenecks here, leading to the adoption of the Leaf-Spine design.

- **Spine Switches:** The core network backbone where every spine switch connects to *every* leaf switch. Spine switches do not connect to each other or directly to servers.
- **Leaf Switches:** Connect directly to the servers (racks) and to all spine switches.
- **Key Advantages:** \* Predictable, low latency because every server is always exactly two network hops away from any other server.
  - High bandwidth and redundancy via Equal-Cost Multi-Pathing (ECMP) routing across all available paths.

## 3. Core Networking Issues in Data Centers

Designing and running a cloud-scale data center introduces unique physical and logical challenges

- **Over-subscription:** Network capacity is deliberately designed to share bandwidth under the assumption that not all servers will transmit at maximum capacity simultaneously. If too many servers burst traffic at once, switches drop packets due to buffer exhaustion.
- **Bisection Bandwidth Limitations:** This is the maximum available speed if you split the data center servers into two equal halves. Traditional networks have low bisection bandwidth, meaning the core network cannot handle all servers communicating at full speed simultaneously.
- **Layer-2 Scalability and STP Bottlenecks:** Traditional Ethernet uses the **Spanning Tree Protocol (STP)** to prevent network loops by blocking redundant paths. In a cloud data center, blocking links wastes valuable bandwidth. Cloud environments instead use technologies like **VXLAN (Virtual Extensible LAN)** to create large, flexible virtual networks over a stable physical framework.
- **Energy Efficiency and Cooling:** Networking equipment (switches, optical transceivers) consumes substantial power and generates massive heat. Data centers must optimize routing protocols to sleep idle ports or use efficient hardware to reduce operating costs.

#### 4. Transport Layer Issues in Data Center Networks (DCNs)

Standard TCP (Transmission Control Protocol) was engineered for the wide-area internet, where latency is high and links are unstable. When applied to the high-speed, ultra-low-latency environment of a DCN, standard TCP encounters severe architectural limitations:

##### A. TCP Incast

- **The Problem:** This is a many-to-one communication bottleneck. It occurs during synchronized distributed requests (e.g., a web search or MapReduce job where a parent node queries hundreds of worker nodes simultaneously).
- **The Mechanism:** All worker nodes reply at the exact same fraction of a second. This flash flood of packets overwhelms the buffer of the switch directly connected to the parent node.
- **The Result:** The switch drops the overflowing packets. Standard TCP interprets this as congestion, waits for a Retransmission Timeout (RTO)—which lasts hundreds of milliseconds—and drastically slows down application performance.

##### B. TCP Outcast

- **The Problem:** Occurs when a high-bandwidth Elephant Flow (a long-lived, heavy data transfer like a VM migration) and a low-bandwidth Mouse Flow (a short, latency-sensitive query) share the same network bottleneck port.
- **The Mechanism:** Due to differences in how packets enter switch queues, the heavy port traffic can persistently lock out or drop packets from the smaller flow.
- **The Result:** The smaller flow suffers severe throughput drop and unfair network access, delaying time-critical web responses.

#### Solutions Developed for DCN Transport Layers

To solve these limitations, cloud networks use specialized transport protocols:

- **DCTCP (Data Center TCP):** Uses Explicit Congestion Notification (ECN) bits in packet headers to detect small buffer build-ups early. Instead of waiting for dropped packets, it smoothly scales back its transmission speed based on the exact amount of congestion.
- **RoCE (RDMA over Converged Ethernet):** Bypasses the operating system kernel and host CPU entirely, allowing servers to read/write directly to another server's remote memory at hardware speeds without standard TCP overhead.

## 5. Network Profiles of Major Cloud Service Providers (CSPs)

The world's leading hyper scale cloud providers manage their global network infrastructures differently to gain advantages in speed, availability, and global coverage.

### 1. Amazon Web Services (AWS)

- **Architecture Style:** Built primarily on highly isolated Regional concepts and Available Zones (AZs) connected by ultra-low latency fiber webs.
- **Key Innovations:** Developed custom **Nitro cards** to offload virtualization and networking tasks directly to dedicated microchips, leaving the host server CPU 100% free for user workloads.
- **Core Network Services:** Amazon VPC (Virtual Private Cloud), AWS Direct Connect (dedicated private line bypass to AWS datacenters), and AWS Route 53 (DNS).

### 2. Microsoft Azure

- **Architecture Style:** Highly interconnected globally, utilizing massive Software-Defined Networking (SDN) deployments to manage infrastructure changes on the fly.
- **Key Innovations:** Heavily integrated **FPGA (Field Programmable Gate Arrays)** inside their SmartNICs (Smart Network Interface Cards) to process network packets at hardware wire-speed without burdening host CPUs.
- **Core Network Services:** Azure Virtual Network (VNet), Azure ExpressRoute (private connection framework), and Azure Load Balancer.

### 3. Google Cloud Platform (GCP)

- **Architecture Style:** Operates one of the largest private global fiber-optic networks in existence. Google routes client traffic onto its private backbone as close to the user as possible (Cold-Potato Routing).
- **Key Innovations:** Pioneered **Andromeda**, a highly scalable software-defined network virtualization platform, and **B4**, a private global WAN built entirely on Software-Defined Networking (SDN) using OpenFlow.
- **Core Network Services:** VPC Network, Cloud Interconnect, and Google Cloud CDN

