# Network Computing - Notes - v0.1.0

260236

March 2025

# Preface

Every theory section in these notes has been taken from the sources:

- Course slides. [1]

About:

GitHub repository

These notes are an unofficial resource and shouldn't replace the course material or any other book on network computing. It is not made for commercial purposes. I've made the following notes to help me improve my knowledge and maybe it can be helpful for everyone.

As I have highlighted, a student should choose the teacher's material or a book on the topic. These notes can only be a helpful material.

# Contents


**1 Datacenters** **4**

1.1 What is a Datacenter? . . . . . . . . . . . . . . . . . . . . . . . . . 4

1.2 Datacenter Applications . . . . . . . . . . . . . . . . . . . . . . . . 9

**Index** **14**

# 1 Datacenters

## 1.1 What is a Datacenter?

A **Datacenter** is a specialized **facility that houses multiple computing resources**, including servers, networking equipment, and storage systems. These **resources are co-located** (placed together in the same physical location) to ensure efficient operations, leverage shared environmental controls (such as cooling and power), and maintain physical security.

So the main characteristics are:

- **Centralized Infrastructure**: Unlike traditional computing models where resources are scattered, datacenters consolidate thousands to millions of machines in a single administrative domain.
- **Full Control over Network and Endpoints**: Datacenters operate under a single administrative entity, allowing customized configurations beyond conventional network standards.
- **Traffic Management**: Unlike the open Internet, datacenter traffic is highly structured, and the organization can define routing, congestion control, and network security policies.

| Feature | Datacenter Networks | Traditional Networks |
|---|---|---|
| **Ownership** | Fully controlled by a single organization | Usually spans multiple independent ISPs |
| **Traffic** | High-speed internal communication (east-west traffic) | Lower-speed, external client-based traffic (north-south) |
| **Routing** | Customizable (non standard protocols) | Uses standard internet protocols (BGP, OSPF, etc.) |
| **Latency** | Optimized for ultra-low latency | Variable latency, dependent on ISPs |
| **Redundancy** | High redundancy to ensure failover and fault tolerance | Often limited by ISP policies |

Table 1: Difference between Datacenters and other networks (e.g., LANs).

### ❓ Why are datacenters important?

Datacenters are the backbone of modern cloud computing, large-scale data processing, and AI/ML workloads. They provide **high computational power and storage** for various applications, such as:

1. **Web Search & Content Delivery**. For example, when a user searches for "Albert Einstein" on Google, the request is processed in a datacenter where:

   (a) The query is parsed and sent to multiple servers.
   (b) Indexed data is retrieved.
   (c) A ranked list of results is generated and sent back to the user.

2. **Cloud Computing**. Services like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud offer computation, storage, and networking resources on-demand.
   - Infrastructure as a Service (IaaS): Virtual machines, storage, and networking.
   - Platform as a Service (PaaS): Databases, development tools, AI models.
   - Software as a Service (SaaS): Google Drive, Microsoft Office 365.
3. **AI and Big Data Processing**. Large-scale computations like MapReduce and deep learning training rely on distributed datacenter resources.
4. **Enterprise Applications**. Datacenters host internal IT infrastructure for businesses, including databases, ERP systems, and virtual desktops.

### Evolution of Datacenters

While the concept of centralized computing dates back to the 1960s, the modern datacenter model emerged with cloud computing in the 2000s. Notable developments include:

- 1970s: IBM mainframes operated in controlled environments similar to early datacenters.
- 1990s: Rise of client-server computing required dedicated server rooms.
- 2000s-Present: Hyperscale datacenters by Google, Microsoft, and Amazon revolutionized networking, storage, and scalability.

### What's new in Datacenters?

Datacenters have been around for decades, but modern datacenters have undergone significant changes in scale, architecture, and service models. The primary **factors driving these changes** include:

- ✔ The exponential **growth of internet services** (Google, Facebook, Amazon, etc.).
- ✔ The **shift to cloud computing** and on-demand services.
- ✔ The need for **better network scalability, fault tolerance, and efficiency**.

One of the most striking changes in modern data centers is their massive scale:

- Companies like Google, Microsoft, Amazon, and Facebook operate **datacenters with over a million servers at a single site**.
- **Microsoft alone has more than 100,000 switches and routers** in some of its datacenters.
- **Google processes billions of queries per day**, requiring vast computational resources.
- **Facebook and Instagram serve billions of active users**, with every interaction generating requests to datacenters.

Another major change is the **shift from owning dedicated computing infrastructure** to **renting scalable cloud resources**. Datacenters no longer just host enterprise applications, **they now offer computing, storage, and network infrastructure as a service**. The most common cloud computing models are:

- **Infrastructure as a Service (IaaS)**. User rent virtual machines (VMs), storage, and networking instead of maintaining their own physical servers (e.g., Amazon EC2).
- **Platform as a Service (PaaS)**. Provides a platform with pre-configured environments for software development (databases, frameworks, etc.).
- **Software as a Service (SaaS)**. Full software applications hosted in datacenters and delivered via the internet (e.g., Google Drive).

The move to cloud computing has fundamentally changed datacenters, shifting the focus to resource allocation, security, and performance guarantees. They are also moving from multi-tenancy to single-tenancy:

- **Single-Tenancy**. A client gets **dedicated infrastructure** for their services.
- **Multi-Tenancy**. **Resources are shared among multiple clients while ensuring isolation**.

✖ **Implications**. But this massive scale brings new challenges:

- **Scalability**: The need for **efficient network designs** to handle rapid growth.

  Traditional datacenter topologies, such as tree-based architectures, are inefficient at scale. New designs, like **Clos-based networks (Fat Tree)** and **Jellyfish (random graphs)**, are being developed to:

  - ✔ Ensure **high bisection bandwidth** (allow any-to-any communication efficiently).
  - ✔ Provide **scalable and fault-tolerant networking**.

- **Cost management**: More machines mean **higher power, cooling, and hardware costs**.

  Datacenters are **expensive to build and maintain**, requiring:

  - **Efficient resource utilization** (prevent idle servers from wasting power).
  - **Energy-efficient cooling solutions** (cooling accounts for a *huge* portion of operational costs).
  - **Automation to reduce human intervention** (e.g., AI-based network optimization).

- **Reliability**: Hardware failures become **common at scale**, requiring **automated fault-tolerant solutions**.

  At the scale of modern datacenters, **hardware and software failures are common**. A key principle is: "*In large-scale systems, failures are the norm rather than the exception.*" (Microsoft, ACM SIGCOMM 2015).

  Thus, new **automated failover mechanisms** are required to:

  - Detect failures **quickly**.
  - Redirect traffic **seamlessly**.
  - Ensure **minimal service disruption**.

- **Performance & Isolation Guarantees**: In modern datacenters, **customers expect strict performance guarantees** for applications like: low-latency financial transactions, high-bandwidth video streaming, machine learning model training.

  To meet these demands, datacenters implement:

  - ✔ **Performance Guarantees**: Allocating bandwidth and compute power dynamically.
  - ✔ **Isolation Guarantees**: Ensuring one user's workload does not interfere with another's.

  But this requires **advanced networking techniques**, such as:

  - **Traffic engineering** to avoid congestion.
  - **Load balancing** to distribute workloads efficiently.
  - **Software-defined networking (SDN)** for centralized control over traffic flows.

**Key Takeaways: What is a Datacenter?**

- **Datacenters centralize** computing resources for performance, security, and scalability.
- **They differ from traditional networks** by offering more control, lower latency, and higher redundancy.
- **Applications include cloud services, AI, and enterprise computing**.
- **Scalability is a key challenge**, with hyperscale datacenters hosting millions of machines.
- **Efficiency and cost containment are major concerns**, requiring innovative architectures.

## 1.2 Datacenter Applications

Modern datacenters host a variety of applications that range from web services to large-scale data processing. These **applications can be classified based on their traffic patterns and computational needs**.

### Customer-Facing Applications (North-South Traffic)

Customer-facing applications involve direct interaction with users. This type of traffic follows a **North-South communication model**, meaning that **data flows between external users and the datacenter**.

**Example 1: North-South Traffic**

Examples include:

- **Web Search** (e.g., Google, Bing)
  - A user submits a query (e.g., "Albert Einstein").
  - The request is routed through the datacenter's frontend servers.
  - Backend database and indexing servers fetch relevant results.
  - The response is assembled and sent back to the user.
- **Social Media Platforms** (e.g., Facebook, Instagram, X (ex Twitter))
  - Users interact with content hosted in the datacenter (e.g., loading a feed, liking posts).
  - Each interaction requires queries to databases and caching systems.
  - Content delivery is optimized using load balancers.
- **Cloud Services** (e.g., Google Drive, Dropbox, OneDrive)
  - Users upload, store, and retrieve files.
  - Requests must be efficiently distributed across storage nodes.

### Large-Scale Computation (East-West Traffic)

Unlike customer-facing applications, backend computations do not involve direct interaction with external users. Instead, they focus on **processing massive datasets within the datacenter**. This type of traffic is known as **East-West traffic** because it occurs **between servers inside the datacenter** *rather than between the datacenter and the external world.*

**Example 2: East-West Traffic**

Examples include:

- **Big Data Processing** (e.g., MapReduce, Hadoop, Spark)
  - Large datasets are distributed across multiple servers.
  - Each server processes a portion of the data in parallel.
  - Results are combined to generate insights (e.g., web indexing, analytics).
- **Machine Learning & AI Training** (e.g., Deep Learning Models)
  - AI models are trained on massive datasets using clusters of GPUs/TPUs.
  - The process requires high-bandwidth, low-latency communication.
  - Synchronization between nodes is critical (e.g., gradient updates in distributed training).
- **Distributed Storage & Backup Systems** (e.g., Google File System, Amazon S3)
  - Data is replicated across multiple locations for reliability.
  - Servers frequently exchange data to ensure consistency and fault tolerance.

### ⚖ Key differences between North-South and East-West traffic

| Feature | N-S traffic | E-W traffic |
|---|---|---|
| **Direction** | External users ↔ Datacenter | Within datacenter |
| **Examples** | File downloads | AI training |
| **Bandwidth Needs** | Moderate | Very High |
| **Latency Sensitivity** | High | Critical |
| **Traffic Type** | Query-response | Bulk data transfer |

Table 2: Differences between North-South and East-West traffic.

In terms of latency sensitivity, North-South traffic is high because user interactions must be fast. On the other hand, East-West traffic is critical because synchronization delays affect computation.

### Traffic Patterns and Their Impact on Networking

The way data moves within a datacenter **heavily influences network design**. The main **goal** is to **ensure high bandwidth, low latency, and efficient resource utilization**.

- **Any-to-Any Communication Model**
  - In **large-scale distributed applications**, **any server should be able to communicate with any other server at full bandwidth**.
  - **Network congestion can severely degrade performance**, especially for AI/ML workloads and big data processing.
- **High-Bandwidth Requirements**
  - Applications like **MapReduce** and **deep learning** require **high data transfer rates**.
  - If bandwidth is insufficient, **bottlenecks occur**, leading to delays.
- **Latency is a Critical Factor**
  - **Low-latency networking is essential** for interactive applications and distributed computing.
  - AI training, for example, requires nodes to synchronize frequently; a delay in one node slows down the entire process.
- **Worst-Case (Tail) Latency Matters**
  - It's not enough for **most requests** to be fast; **the slowest request** can delay the entire computation.
  - **Minimizing tail latency** is crucial for efficient AI model training and database queries.

### Challenges in Datacenter Traffic Management

The massive scale and complexity of modern datacenters introduce **several networking challenges**, including:

- **Network Congestion and Bottlenecks**. When multiple servers communicate simultaneously, **some network links become overloaded**, leading to congestion.

  For example, if many AI training jobs share the same network path, it can become a bottleneck, slowing down training.

  This can be a **critical issue for applications requiring real-time performance** (e.g., financial transactions, cloud gaming).

- **Load Balancing and Traffic Engineering**. How do we distribute traffic *efficiently* across network links? The solutions are: **Equal-Cost Multi-path Routing** (ECMP, spreads traffic across multiple paths); **Dynamic Traffic Engineering** (adjusts paths in real time based on congestion levels).

- **Avoiding Link Over-Subscription**. If too many servers send data over a single link, the available **bandwidth is divided**, leading to **slow performance**. Modern datacenters aim for **full-bisection bandwidth**, meaning **any server can talk to any other server at full capacity**.
- **Scaling Challenges**. Traditional datacenter network architectures do not scale well beyond a certain point. **New network topologies** (e.g., Fat Tree, Jellyfish) are being adopted to address these limitations.

**Key Takeaways: Datacenter Applications**

- Datacenters handle **two major types of applications**:
  1. **Customer-facing applications (North-South traffic)** involve external users.
  2. **Large-scale computations (East-West traffic)** occur within the datacenter.
- **Traffic patterns** affect **bandwidth, latency, and congestion control**.
- **Managing congestion and ensuring high bandwidth** is critical for performance.
- **New network topologies and routing techniques** help address scaling challenges.

# References

[1] Antichi Gianni. Network computing. Slides from the HPC-E master's degree course on Politecnico di Milano, 2024.

# Index


**D**

Datacenter 4

**E**

East-West traffic 9

**M**

Multi-Tenancy 6

**N**

North-South traffic 9

**S**

Single-Tenancy 6