Chapter 9: Distributed Systems

Chapter 9: Distributed Systems

- Advantages of Distributed Systems
- Network Structure
- Communication Protocols
- Network and Distributed Operating Systems
- Design Issues of Distributed Systems

Chapter Objectives

- Explain the advantages of networked and distributed systems
- Define the roles and types of distributed systems in use today

Overview

- A **distributed system** is a collection of loosely coupled nodes interconnected by a communications network

- Nodes variously called *processors, computers, machines, hosts*

  - *Site* is location of the machine, *node* refers to specific system

  - Generally a *server* has a resource a *client* node at a different site wants to use
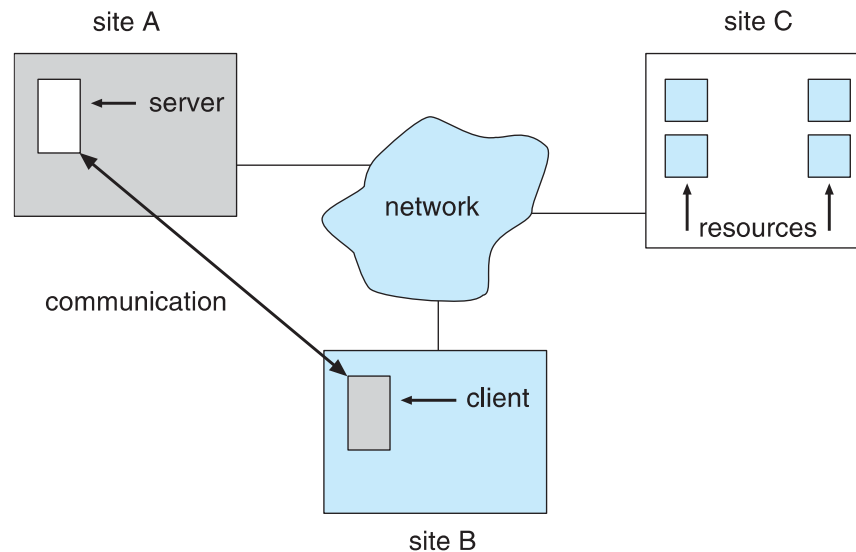
Overview (cont.)

- Nodes may exist in a ***client-server, peer-to-peer***, or ***hybrid*** configuration.

    - In client-server configuration, server has a resource that a client would like to use

    - In peer-to-peer configuration, each node shares equal responsibilities and can act as both clients and servers

- Communication over a network occurs through **message passing**

    - All higher-level functions of a standalone system can be expanded to encompass a distributed system

Reasons for Distributed Systems

- **Resource sharing**

    - Sharing files or printing at remote sites

    - Processing information in a distributed database

    - Using remote specialized hardware devices such as *graphics processing units* (GPUs)

- **Computation speedup**

    - Distribute subcomputations among various sites to run concurrently

    - **Load balancing** – moving jobs to more lightly-loaded sites

- **Reliability**

    - Detect and recover from site failure, function transfer, reintegrate failed site
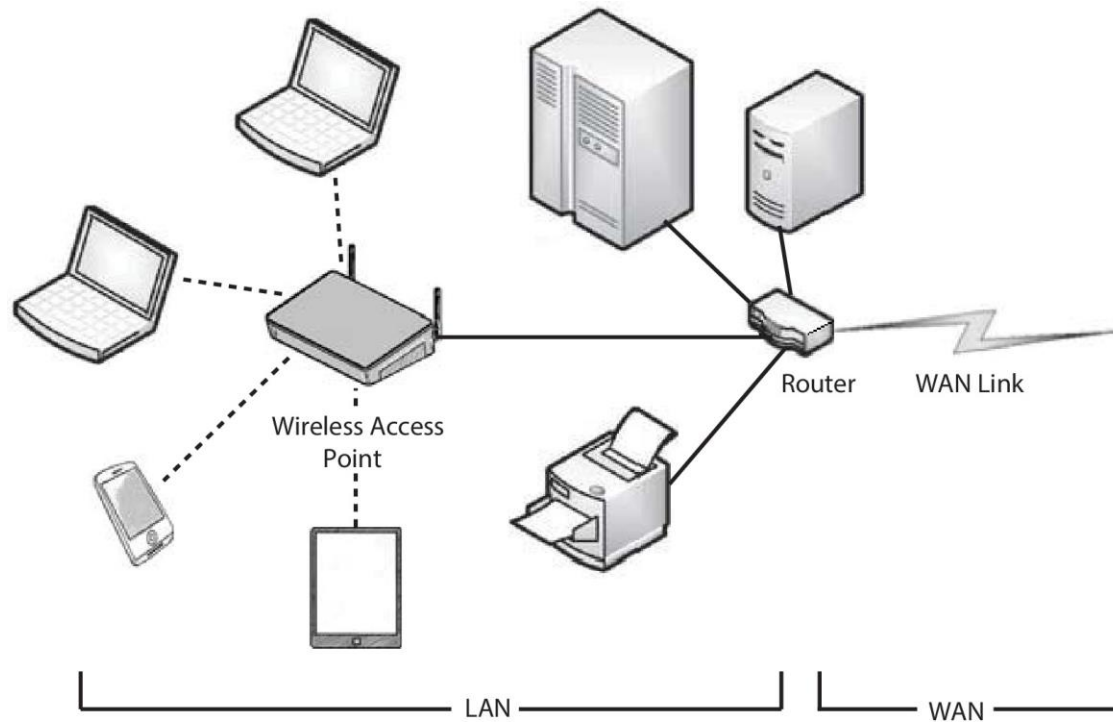
Network Structure

- **Local-Area Network** (**LAN**) – designed to cover small geographical area

    - Consists of multiple computers (workstations, laptops, mobile devices), peripherals (printers, storage arrays), and routers providing access to other networks

    - Ethernet and/or Wireless (**WiFi**) most common way to construct LANs

        ▸ Ethernet defined by standard IEEE 802.3 with speeds typically varying from 10Mbps to over 10Gbps

        ▸ WiFi defined by standard IEEE 802.11 with speeds typically varying from 11Mbps to over 400Mbps.

        ▸ Both standards constantly evolving

Local-Area Network (LAN)

Network Structure (Cont.)

- **Wide-Area Network** (**WAN**) – links geographically separated sites
    - Point-to-point connections via links
        - ▸ Telephone lines, leased (dedicated data) lines, optical cable, microwave links, radio waves, and satellite channels
    - Implemented via **routers** to direct traffic from one network to another
    - Internet (World Wide Web) WAN enables hosts world wide to communicate
    - Speeds vary
        - ▸ Many backbone providers have speeds at 40-100Gbps
        - ▸ Local **Internet Service Providers** (**ISPs**) may be slower
        - ▸ WAN links constantly being upgraded
    - WANs and LANs interconnect, similar to cell phone network:
        - ▸ Cell phones use radio waves to cell towers
        - ▸ Towers connect to other towers and hubs

Wide-Area Network (WAN)
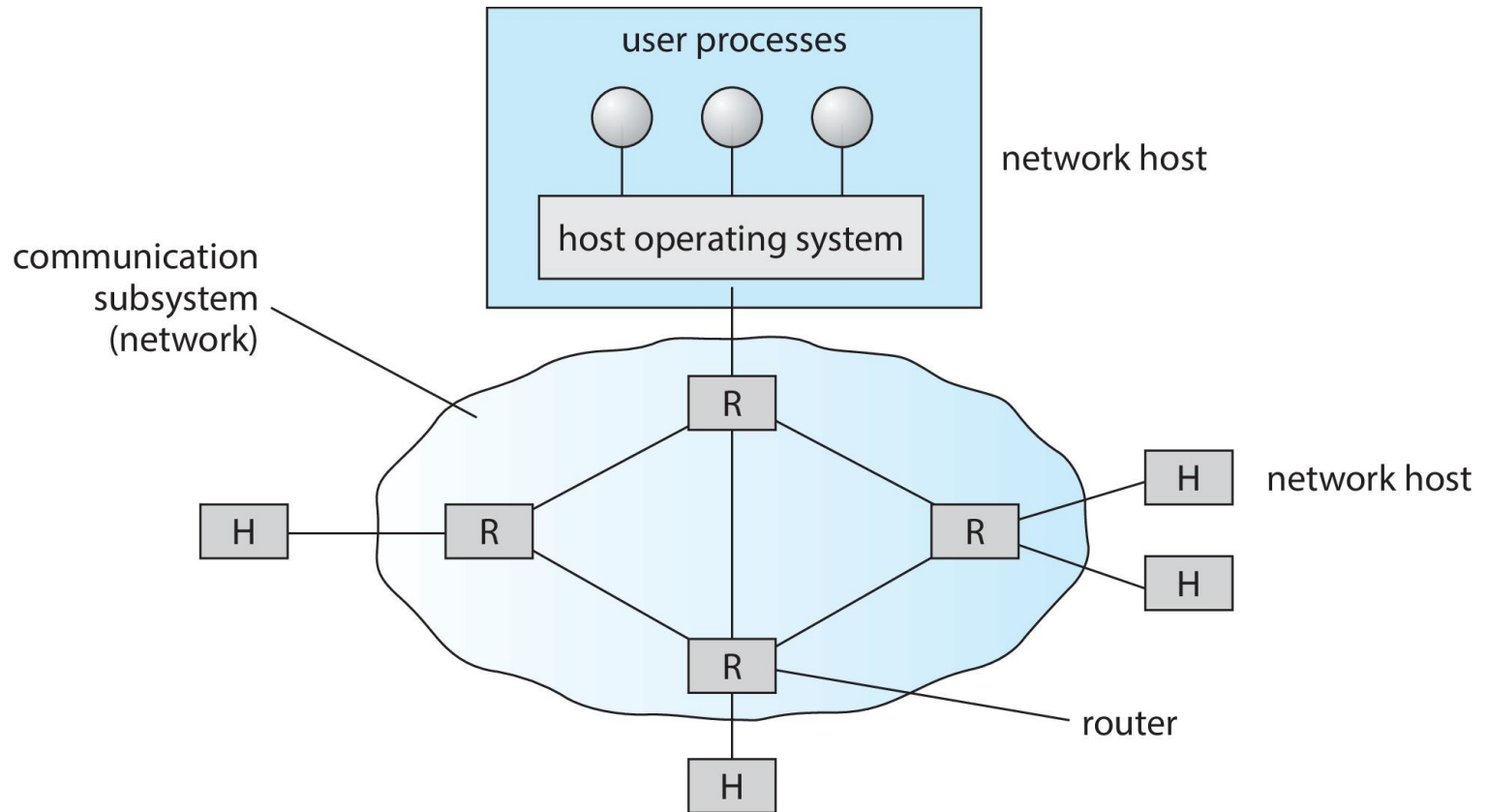
Naming and Name Resolution

- Each computer system in the network has a unique name

- Each process in a given system has a unique name (process-id)

- Identify processes on remote systems by

    <host-name, **identifier>** pair

- **Domain name system** (**DNS**) – specifies the naming structure of the hosts, as well as name to address **resolution** (Internet)

```
/**
 * Usage: java DNSLookUp <IP name>
 * i.e. java DNSLookUp www.wiley.com
 */
public class DNSLookUp {
  public static void main(String[] args) {
    InetAddress hostAddress;

    try {
      hostAddress = InetAddress.getByName(args[0]);
      System.out.println(hostAddress.getHostAddress());
    }
    catch (UnknownHostException uhe) {
      System.err.println("Unknown host: " + args[0]);
    }
  }
}
```

Figure 19.4 Java program illustrating a DNS lookup.

Communication Protocol

- Communication via OSI model (7 Layers)
- Transport Protocols:
    - TCP (Transmission Control Protocol)
        - ▸ Reliable and Connection-oriented
    - UDP (User Datagram Protocol)
        - ▸ Unreliable and Connectionless

Network-oriented Operating Systems

- Two main types

- **Network Operating Systems**
    - Users are aware of multiplicity of machines

- **Distributed Operating Systems**
    - Users not aware of multiplicity of machines

multiplicity : ความมากมายหลากหลายรูปแบบ

Network Operating Systems

- Users are aware of multiplicity of machines

- Access to resources of various machines is done explicitly by:

  - Remote logging into the appropriate remote machine (ssh)

    - `ssh kristen.cs.yale.edu`

  - Transferring data from remote machines to local machines, via the File Transfer Protocol (FTP) mechanism

  - Upload, download, access, or share files through cloud storage

- Users must change paradigms – establish a **session**, give network-based commands, use a web browser

  - More difficult for users

Distributed Operating Systems

- Users not aware of multiplicity of machines
    - Access to remote resources similar to access to local resources
- **Data Migration** – transfer data by transferring entire file, or transferring only those portions of the file necessary for the immediate task
- **Computation Migration** – transfer the computation, rather than the data, across the system
    - Via remote procedure calls (RPCs)
    - Via messaging system

migration : การเคลื่อนย้ายจากที่หนึ่งไปยังอีกที่หนึ่ง

Distributed-Operating Systems (Cont.)

- **Process Migration** – execute an entire process, or parts of it, at different sites

  - *Load balancing* – distribute processes across network to even the workload

  - *Computation speedup* – subprocesses can run concurrently on different sites

  - *Hardware preference* – process execution may require specialized processor

  - *Software preference* – required software may be available at only a particular site

  - *Data access* – run process remotely, rather than transfer all data locally

- Consider the World Wide Web

Design Issues of Distributed Systems

- We investigate three design questions:

    - *Robustness* – Can the distributed system withstand failures?

    - *Transparency* – Can the distributed system be transparent to the user both in terms of where files are stored and user mobility?

    - *Scalability* – Can the distributed system be scalable to allow addition of more computation power, storage, or users?

Robustness

- Hardware failures can include failure of a link, failure of a site, and loss of a message.

- A **fault-tolerant system** can tolerate a certain level of failure

  - Degree of fault tolerance depends on design of system and the specific fault

  - The more fault tolerance, the better!

- Involves *failure detection*, *reconfiguration*, and *recovery*

Failure Detection

- Detecting hardware failure is difficult

- To detect a link failure, a **heartbeat** protocol can be used

- Assume Site A and Site B have established a link

  - At fixed intervals, each site will exchange an *I-am-up* message indicating that they are up and running

- If Site A does not receive a message within the fixed interval, it assumes either (a) the other site is not up or (b) the message was lost

- Site A can now send an *Are-you-up?* message to Site B

- If Site A does not receive a reply, it can repeat the message or try an alternate route to Site B

Failure Detection (Cont.)

- If Site A does not ultimately receive a reply from Site B, it concludes some type of failure has occurred

- Types of failures:
  - Site B is down
  - The direct link between A and B is down
  - The alternate link from A to B is down
  - The message has been lost

- However, Site A cannot determine exactly **why** the failure has occurred

Reconfiguration and Recovery

- When Site A determines a failure has occurred, it must reconfigure the system:

  - If the link from A to B has failed, this must be broadcast to every site in the system

  - If a site has failed, every other site must also be notified indicating that the services offered by the failed site are no longer available

- When the link or the site becomes available again, this information must again be broadcast to all other sites

Transparency

- The distributed system should appear as a conventional, centralized system to the user

  - User interface should not distinguish between local and remote resources

    - Example: NFS

  - User mobility allows users to log into any machine in the environment and see his/her environment

    - Example: LDAP plus desktop virtualization

NFS: Network File System (เป็นกลไกในการเก็บไฟล์ข้อมูลในระบบเครือข่ายเน็ตเวิร์ก)
LDAP: Lightweight Directory Access Protocol (ข้อตกลงที่ใช้ในการเข้าถึง
ไดเรกเตอรี่ในระบบเครือข่ายเน็ตเวิร์ก)

Scalability

- As demands increase, the system should easily accept the addition of new resources to accommodate the increased demand

  - Reacts gracefully to increased load

  - Adding more resources may generate additional indirect load on other resources if not careful

  - Data **compression** or **deduplication** can cut down on storage and network resources used

End of Chapter 9