TOPCIT

ESSENCE Ver. 3

Technical Field

03 Overview of System Architecture

TOPCIT ESSENCE is published to provide learning materials for TOPCIT examinees.
The TOPCIT Division desires the TOPCIT examinees who want to acquire the necessary practical competency in the field of ICTto exploit as self - directed learning materials.

For more information about TOPCIT ESSENCE, visit TOPCIT website or send us an e-mail.

As part of the **TOPCIT ESSENCE** contents feed into authors' personal opinions, it is not the TOPCIT Division's official stance.

Ministry of Science, ICT and Future Planning
Institute for Information and Communications Technology Promotion
Korea Productivity Center

Publisher TOPCIT Division
+82-2-398-7649 www.topcit.or.kr helpdesk@topcit.or.kr

Date of Publication 1st **Edition** 2014. 12. 10.
2nd **Edition** 2016. 02. 26.
3rd **Edition** 2020. 02. 26.

Copyright © Ministry of Science, ICT and Future Planning
All rights reserved.
No part of this book may be used or reproduced in any manner whatever without written permission.

TOPCIT

ESSENCE Ver. 3

Technical Field

03 Overview of System Architecture

TOPCIT

ESSENCE Ver. 3

Technical Field

03 Overview of System Architecture

CONTENTS

I. System Concept 20

01 Concept of System Architecture 22

A) Understanding system architecture 22
B) Components of information system 23

02 Types of System Architecture 26

A) Types of system architecture 26
B) System architecture classification according to system layout 26
C) Classification according to how the application programs are provided 27
D) Classification by system layer 28

03 Server's Stack Architecture 29

A) Concept of server's stack architecture 29
B) Computer hardware 30

II. Network Concept 32

01 What is the Protocol? 36

02 OSI Reference Model and TCP/IP Protocol Layer Structure 37

03 Internet Address System 39

04 Internet Standard 40

III. Operating System (OS) 41

01 Operating System (OS) 42

A) Overview of OS 42
B) Main functions of OS 43
C) Types of main OS 44

02 Process and Thread 45

A) Understanding of process 45
B) Process management technique 46
C) Thread 47

03 Process Synchronization and Deadlock 48

A) Concept of process synchronization 48
B) Critical section problem 48
C) Solving the critical section problem 49
D) Deadlock status 49

04 Memory Unit Management 50

A) Understanding of memory unit management 50
B) Memory unit management technique 50

05 Scheduling 53

CONTENTS

A) Purpose of scheduling 53
B) Type and characteristics of scheduling 53

06 Virtual Memory Nnit 54

A) mplementing a virtual memory unit 54
B) Page replacement technique of the virtual memory unit 55
C) Factors that affect the virtual memory unit performance 56

07 File System 56

A) Understanding of file system 56
B) Concept of directory 56
C) Allocation by a file system 57
D) Types of files system for each OS 59
E) UNIXi-node 59

08 Input/ Output System 59

IV. Computer Architecture 61

01 Computer Structure and Architecture 62

A) Basic computer structure 62
B) Types of computer architecture 63

02 CPU 64

A) Definition of CPU의 64
B) CPU execution 64

C) CPU components 64
D) Instruction cycle 65
E) Instruction set structure, CISC and RISC 66

03 Memory 67

A) Memory unit's hierarchical structure 67
B) Factors for performance evaluation of memory unit 68
C) Type and characteristics of the memory unit 68
D)Addressing mode 68
E) Locality 69

04 I/O Device 69

A) Concept of I/O device 69
B) I/O controller structure and addressing methods 69
C) DMA) 70

05 Latest Technologies and Trends 72

A) Neuromorphic chip 72
B) Quantum computer 73

V. Data Processing Technology 75

01 Parallel Processing System 76

A) Concept of the parallel processing system 76
B) Flynn's classification of parallel processing systems 77
C) Classification of parallel processing systems, according to the memory structure 78

CONTENTS

D) Types of parallel processor technology 79
E) Parallel programming technology 81
F) Parallel programming technology 82
G) GPU-based parallel programming technology 82

02 Storage Technology 83

A) Concept of storage 83
B) Connection of storage unit and server 84
C) IP-SAN 86
D) Storage capacity management 87
E) Storage disk scheduling 88

03 High Availability Storage 91

A)Redundant array of independent disks (RAID) technology 91
B) Backup storage: LTO and VTL 94

04 Graphic Compression Technology 94

A) Graphic compression type 94
B) Multimedia data 95

VI. Embedded System 98

01 Overview of Embedded System 99

A) Definition of embedded system 99
B) Characteristics of embedded system 100

C) Embedded system development process 101

02 Embedded Hardware 103

A) Microprocessor, the key to embedded systems 103
B) Technical trends of embedded hardware 104

03 Embedded Software 105

A) Definition of embedded software and technology application 105
B) Characteristics of embedded software 105

04 Embedded OS 106

A) Concept of embedded OS 106
B) Embedded OS structure and applied technology 106
C) Types and trends of embedded OS 107

Ⅶ. Information System Implementation 109

01 Information System Structure 110

A) Concept of information system 110
B) Information system structure 111

02 Information System Capacity Calculation 115

A) Concept of size calculation 116
B) Size calculation targets 116
C) Reference architecture for size calculation 117

CONTENTS

D) Size calculation procedure 118
E) Size calculation method for each hardware component 119

03 Latest Technology and Trends of Information Systems 120

Ⅷ. Fault Response Technology 122

01 High Availability (HA) 123

A) Concept of HA 123
B) HA configuration type 124

02 Fault-Tolerant System 126

A) Fault-tolerant system 126

03 Disaster Recovery System (DRS) 126

A) Definition of DRS 126
B) Disaster recovery goal 127

Ⅸ. Cloud Computing Technology 128

01 Definition of Cloud Computing 129

A) Comparison of cloud computing and other computing technologies 130

02 Cloud Computing Types 130

A) Classification according to the service type 130
B) Classification according to the cloud operation form 132

03 Server Virtualization Technology 132

A) Hypervisor 132
B) Hypervisor types 132
C) Server virtualization methods 133

04 Storage and Network Virtualization Technologies 136

A) Storage virtualization 136
B) Network virtualization 136

05 Cloud Platform 136

A) Definition of cloud platform 136
B) Openstack 136
C) CloudStack 137
D) Kubernetes 137
F) Mesos 137

X. Big Data System 138

01 Bid Data System Structure 139

A) Hadoop ecosystem 139
B) Hadoop's main technology elements 140
C) Hadoop support program 141

CONTENTS

D) Commercial Hadoop solutions 142

02 Big Data System Trends and Forecast 143

A) Big data system trends 143
B) Forecast of big data systems 143

XI. Datalink Layer 145

01 Concept of datalink layer 147

02 Encapsulation of the Datalink Layer 147

A) Encapsulation of the datalink layer 147
B) Frame header and trailer 147

03 Datalink Layer Configuration 148

A) Sub-layers of datalink layer 148
B) Logical link control (LLC) 148
C) MAC 149

04 MAC Address Search 149

A) IP address and MAC address conversion protocol 149
B) MAC address search scenario 150

05 Techniques of Datalink Layer Error Detection and Correction 150

A) Concept of error control 150
B) Error detection and correction 151

06 IEEE 802 Standard 152

A) Concept of IEEE 802 152
B) IEEE 802.3 standard 153
C) IEEE 802.11 standard 153
D) IEEE 802.15 standard 154

XII. Network Layer 156

01 Network Layer Overview and Equipment 158

A) What is a network layer? 158
B) Network layer function 158
C) Internetworking equipment 158
D) What is a router? 159
E) What is a switch? 161
F) Virtual local area network (VLAN) 161

02 Network Layer Encapsulation 162

A) Network layer encapsulation 162
B) IPv4 header 163

03 Packet Switching and Network Layer Protocol/Instruction 163

A) Packet switching 163

CONTENTS

B) Network layer protocol/command 163
C) Network layer command 164

04 Network Service Quality 165

A) Quality of Service (QoS) characteristics 165
B) Quality implementation technique 165

05 Routing Protocol Algorithm and Type 166

A) What is a routing protocol? 166
B) What is a routing algorithm? 166
C) Routing protocol types 168
E) Routing protocol type 169

06 IPv4 Overview 169

A) What is an IPv4 (Internet Protocol Version 4) address? 169
B) Network using IPv4 addresses (Example) 170

07 IPv4 address Designation System and Subnetting 171

A) IPv4 expression 171
B) IPv4 address designation system 171
C) Special IPv4 address 172
D) Subnetting 173
E) CIDR(Classless Inter-Domain Routing) 174
F) Supernetting 174
G) IPv4 address allocation 175

08 IPv6 Overview 176

A) What is an IPv6 (Internet Protocol Version 6) address? 176
B) Background of IPv6 176

09 IPv6 Addressing System and Header Structure 176

A) IPv6 addressing system 176
B) IPv6 header structure 177

XIV. Transport Layer Protocol 179

01 Concept of Transport Layer Protocol 180

02 TCP 181

A) Characteristics of TCP 181
B) Virtual TCP operation scenario 182
C)TCP protocol 183

03 UDP(User Datagram Protocol) 189

A) Characteristics of UDP 189
B) UDP protocol 190

04 SCTP(Stream Control Transmission Protocol) 190

A) Characteristics of SCTP 190
B) SCTP 191

CONTENTS

XV. Application Layer Technologies, Including Web Application 194

01 What is Application Layer Protocol? 197

02 HTTP 197

03 File Transfer Protocol 198

XVI. Mobile Communication and Multimedia Communication Technology 201

01 Multimedia Network 202

A) Graphic compression type 202
B) Multimedia data 202
C) QoS(Quality of Service) 203

02 Internet Telephony Concept and Call Signal Protocol 204

A) What is VoIP (Voice over Internet Protocol)? 204
B) VoIP call signal protocol 205
C) H.323 206
D) VoLTE(Voice over LTE) 207

03 Media Transport Protocols 208

A) RTP 208

B) RTCP 208
C) RTSP 208
D) IMS 208

XVII. Latest Technologies 210

01 IoT Network-Based Technology 212

A) IoT concept 212
B) IoT standardization trend 212
C) Core IoT technologies 213
D) Main IoT protocols 214

02 Software-Based Network 216

A) Limitations of the existing communication environment and paradigm shift 216
B) SDN (Software Defined Network) 216
C) NFV(Network Function Virtualization) 218
D) SDN and NFV 220

I. System Concept

▶▶▶ Recent trends and issues

The system architecture is becoming more complex as new technologies, such as big data, cloud computing, and IoT, have gained attention. While the system architecture consisted of a simple combination of servers, storage, and network in the past, its configuration is changing to meet the requirements of the new technologies to provide high-performance computing power, high-capacity memory, and high-speed communication. In line with these changes, the system architecture is becoming more important. The system architecture is an essential concept in a practical environment for system implementation and operation.

▶▶▶ Learning objectives

1. To be able to understand the concept of the system architecture and explain its components.
2 .To be able to understand and explain the system concept and system layers.
3. To be able to explain the types of information systems.

▶▶▶ Keywords

System architecture, System layer, Information system

✚ Preview for practical business

"Discovery of a 'Chinese hacking chip' in a well-known server board in the United States...Industry thrown into commotion" 2018. 10. 05.

Citing the words of six senior US intelligence officials on the 4th (local time), Bloomberg reported that a hacking microchip, that was not in the design, was found on the server mainboard of a well-known server manufacturer, Super micro, and that the Chinese Government was allegedly involved.

(Source: http://it.chosun.com/site/data/html_dir/2018/10/05/2018100502710.html)

There is not a moment in our daily lives when we do not use the services provided by computer systems. With the wide-spread use of smartphones, we are spending even more time with computer systems. In such an environment, Chinese hacking chips can expose our lives to hackers. This hacking chip was found on a computer system's board. It contains stolen key US information through hardware hacking, which is more threatening than software hacking. This tiny chip has the capability to steal information from a computer over a network. How can such a small chip contain this function? While the early computer, ENIAC, was the size of a building, computers have become drastically smaller since then, and the computing power of smartphones that we carry in our hand is so high, that it is not even comparable to that of ENIAC. It is necessary to learn the system architecture to understand the concept of a computer system.

01 Concept of System Architecture

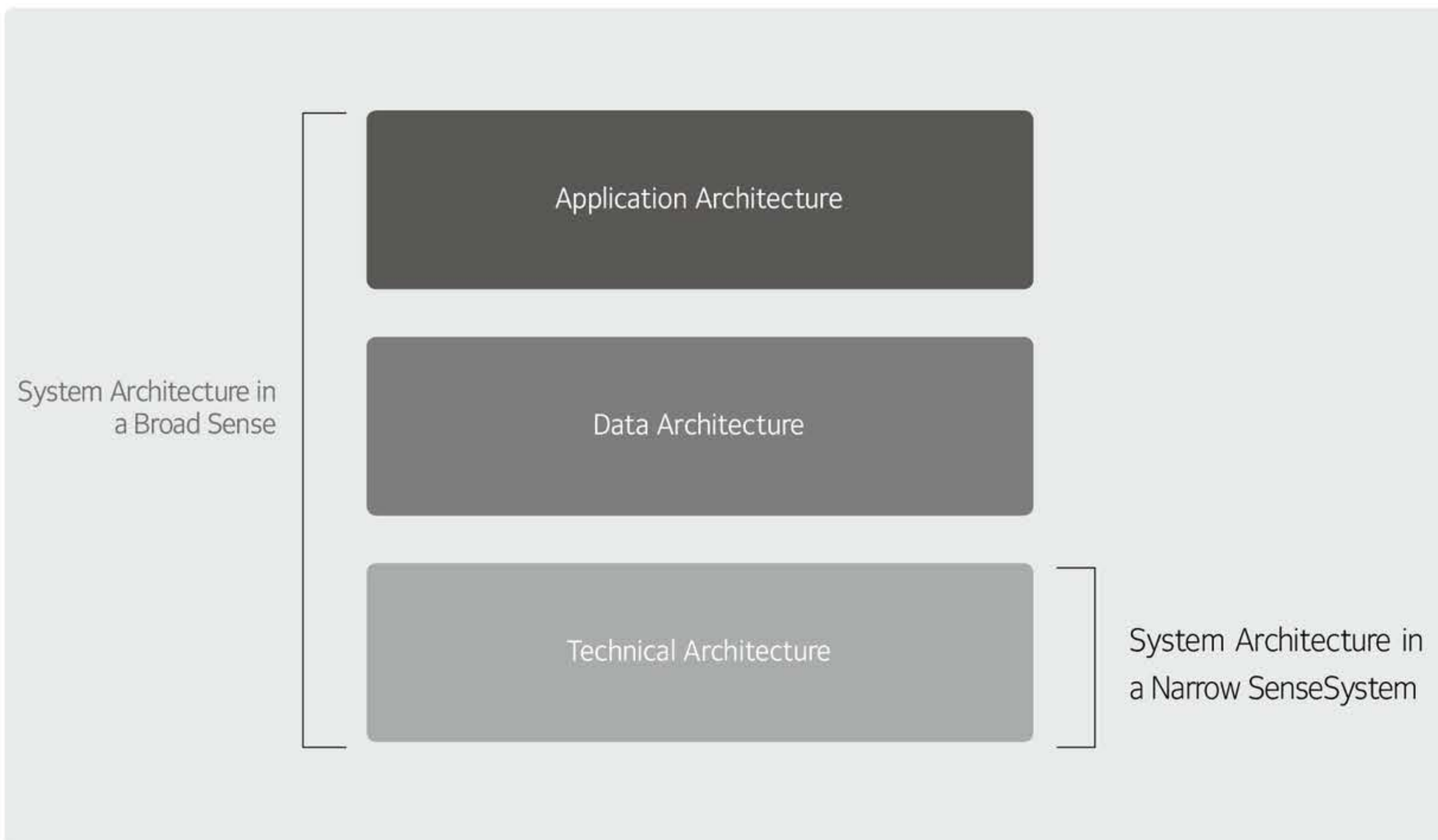
A) Understanding system architecture

① Overview of system architecture

An information system is a system that uses information technology (IT). The information system provides services using a computer, and a computer operates with a combination of hardware and software. Implementing an information system used inside an enterprise, or provided as a service to users, requires the design and construction of IT infrastructures, such as hardware and software design and development. Software designs the software components and operations, such as the connection of the software components and constraints based on software architecture. Hardware, like software, designs and develops the hardware architecture. System architecture refers to the architecture for a system to provide services, based on hardware and software architecture. In a broader definition, it refers to all architecture, including the application architecture (AA), data architecture (DA), and technical architecture (TA), for constructing an information system. A narrow definition refers to the document that defines the configuration and relationship of hardware, such as a server, storage, network and security, and certain software parts, such as OS and middleware.

② Definition of system architecture

- Definition as defined by the International Council on Systems Engineering (INCOSE)
  The fundamental and unifying system structure defined in terms of system elements, interfaces, processes, constraints, and behaviors.



[Figure 1] Definition of System Architecture

• Broad definition of system architecture

A system architecture is a document that defines the structure of an information system provided to support the business process required to achieve the objective of an enterprise. A system architecture is divided into TA, which defines the structure of hardware, such as a server, network, and security, as well as the distributed structure of middleware operating on the hardware, DA, which defines the data structure to assure data integrity, and AA, which defines the software components and the relationship and constraints of said defined components.

• Narrow definition of system architecture

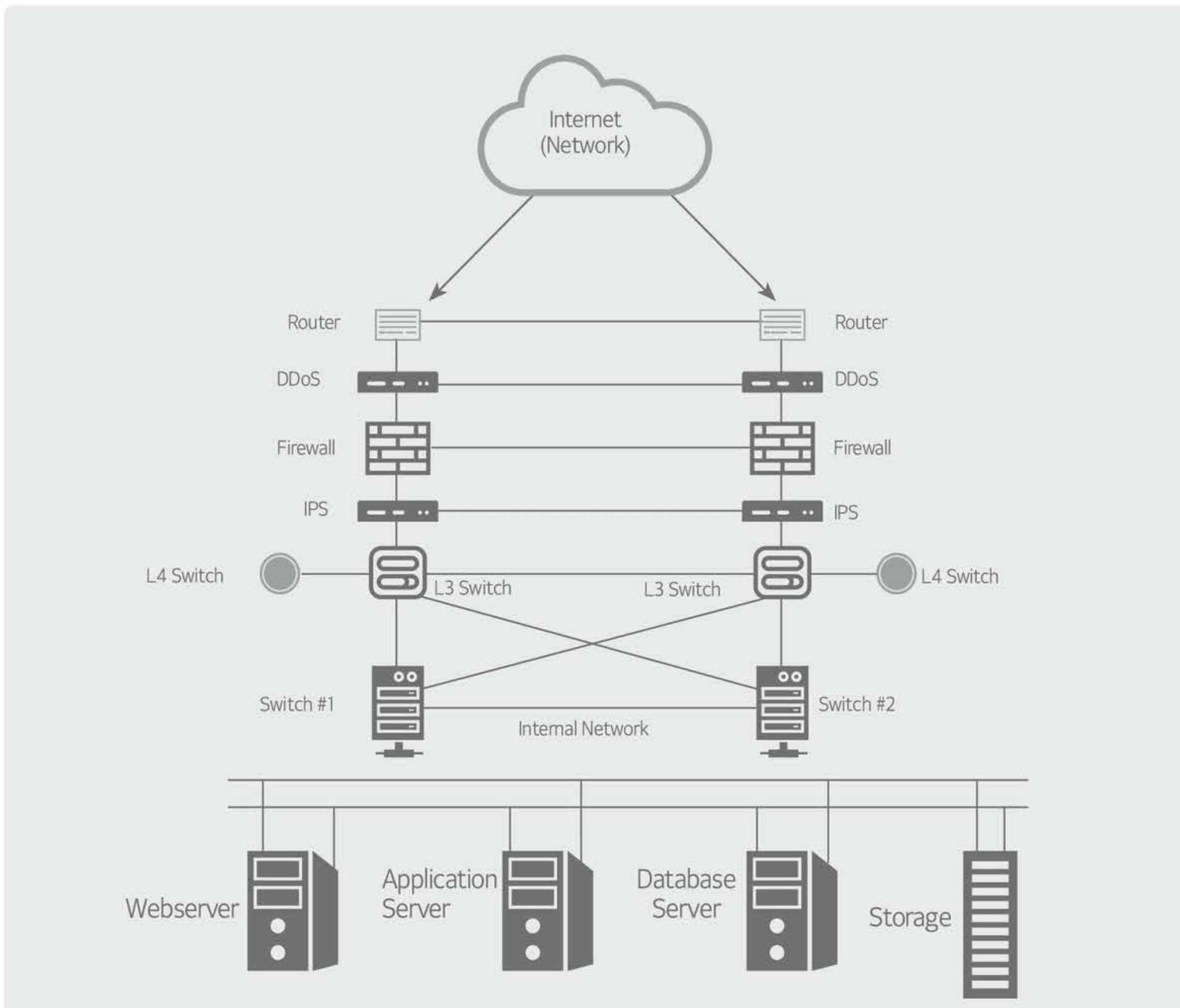
A system architecture refers to the structure of an information system, which includes principles and guidelines that define the components, such as hardware, software, security, interactions, and constraints. In other words, it refers to TA and defines the layout and connection mode of the middleware, such as the DB server and WAS, to distribute the application programs of the developed system and hardware, such as a server, storage, and network. The following table shows the types of detailed architecture defined in system architecture.

Type	Description
Server design architecture	It defines the server resources. - CPU clock speed and number of cores - Memory capacity and network card (NIC) performance and count - Type and capacity of storage units, such as SSD and SAS disk - Server type, such as rack-mount type and blade type - Server size from 1 unit to 4 units
Network design architecture	It defines the network of components that provide services. - Capacity of external networks (Bandwidth) - Determination of connected switches and routers - L4 switches for load balancing - Security systems, such as DDoS, IPS, and firewalls - Network addresses used by services (Subnet, IP, and gateway)
Storage design architecture	It defines separate network storage. - Storage capacity and performance (IOPS and bandwidth) - SSD, SAS, and SATA media type as storage - RAID configuration type - SAN, iSCSI, and NFS storage network type

B) Components of information system

① Definition of information system

System components are a configuration of items to provide the functions of information system services. They include a server that provides the computing power of an information system, storage that stores the data, a network that provides communication between each component, such as a server and storage, and security to provide confidentiality, integrity, and availability of the information system. The detailed components include computer hardware, OS, and middleware operating on a physical server. The following diagram shows the details of each component.



[Figure 2] Information System Components

• Server

A server provides the computing power of an information system. The information system uses this computing power to process the business logic and data by running the application programs. Server components include the stacked structures of the computer's hardware, OS, middleware, and application programs. The system architecture design defines the server capacity, count, layout, and role, and the network facilitates communication between the servers. Mainframes, Unix servers, and x86 servers are the most commonly used server types, and the x86 servers using Intel CPUs have been used more often as the cloud services expand.

For example, the servers of an information system providing web services include a webserver that responds to user requests through a web browser, organizes web pages, and provides a screen to display the contents to users, a web application server (WAS) that processes business logic, according to the user request and delivers the results to a database server for storage, or to a WAS for display, and a database server that creates, requests, modifies, and deletes data.

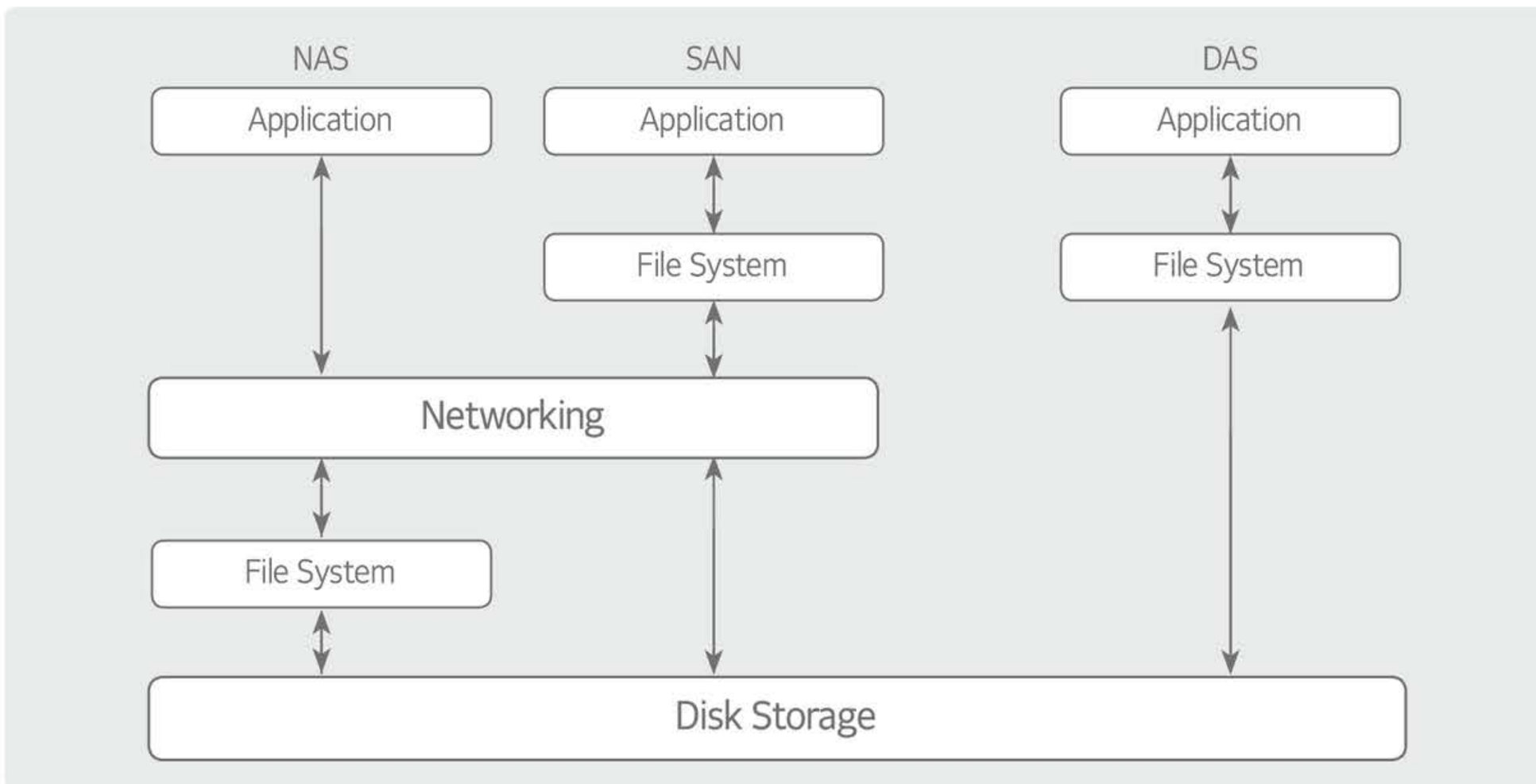
• Network

A network connects the information system components for communication. Network devices process communication between servers, between servers and storage units, and between the enterprise's internal and

external networks. A network's devices and roles vary, based on the OSI layer 7 standard. Network devices include switches, routers, load balancers (L4/L7 load balancers), and bridges. Additionally, AP (access point) is used in wireless networks. There is an overlap between the network domain and the security domain. Since most of the hacking attacks intrude through the network, the internal network (LAN) gate, connected to the external network (WAN), is protected by a security device.

- Storage

A storage system holds the data of an information system. It is categorized into three types according to how it stores data. Block storage saves data in fixed block units, like 16KB and 64KB. It stores the OS, such as MS Windows, Linux, and Unix. File storage saves data in file units, instead of in fixed blocks. Network attached storage (NAS), widely used as the shared file repository, is a type of file storage. Object storage saves data in object units, instead of in fixed blocks or files. It is mostly used in cloud storage. Storage types can also be categorized by their connection method. Direct access storage (DAS) is mounted in each server, NAS is connected through a network, and storage area network (SAN) is connected through the network used exclusively for storage. DAS can only be used with a filesystem and includes IDE, SATA, and SAS. SAN is provided to the OS through an SAN network, and it requires the creation of a filesystem before it can be used as DAS. It includes SAN, iSCSI, and FCoE. NAS becomes usable after it is mounted to the OS. It includes NFS, CIFS, and AFS.



[Figure 3] Storage Type According to Connection Method

- Security

Information security is configured through the network connection. Some security hardware is installed between the external Internet section and the internal network to protect the internal systems, while some are installed in the internal network. It includes the DDoS defense system to defend against DDoS attacks, firewalls to protect the system through monitoring, logging, and blocking network traffic, and IPS/IDS to prevent abnormal traffic or abuse. It also includes web firewalls protect webservers by logging, monitoring, blocking web service traffic, and access control solutions to provide management functions that allow or to deny access to servers.

02 Types of System Architecture

A) Types of system architecture

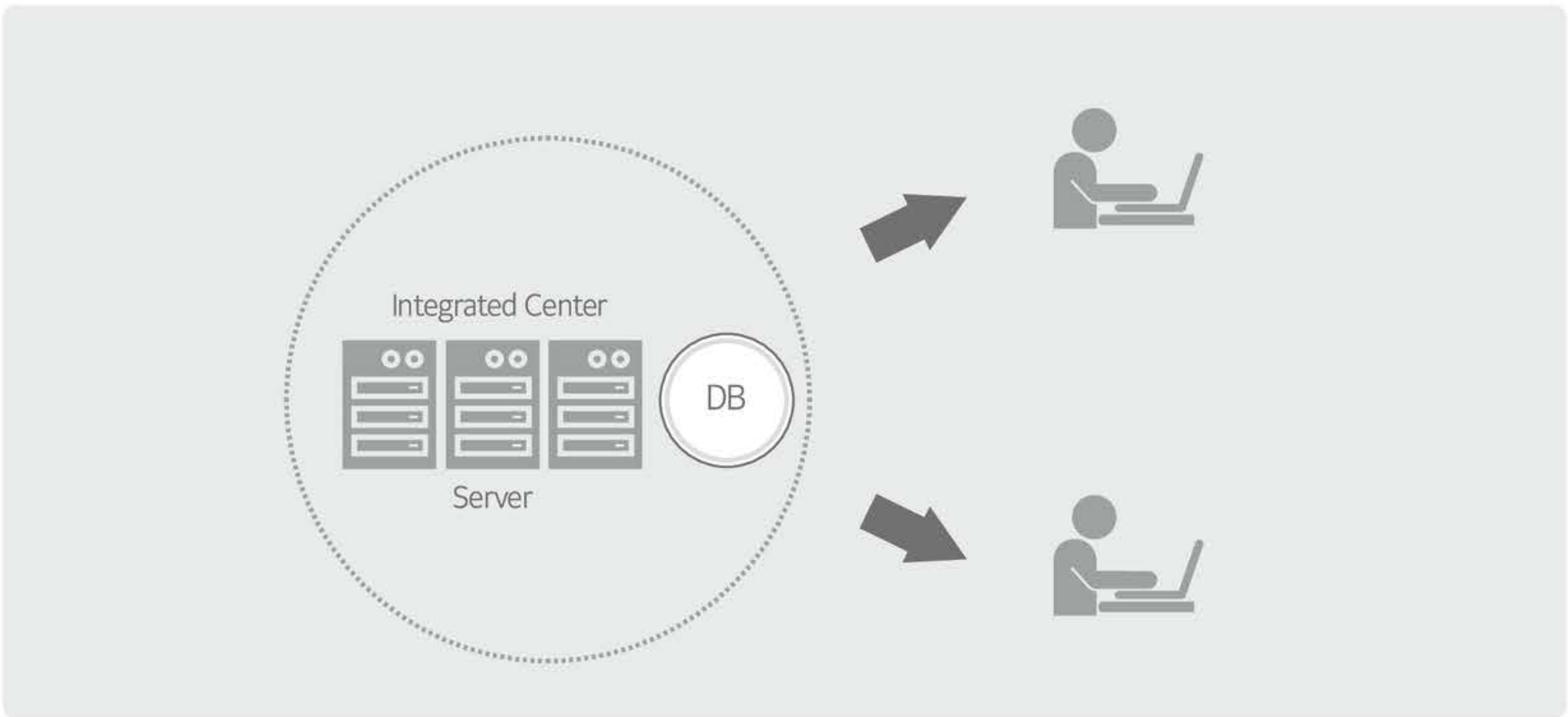
Various ways of configuring a system are provided. System configuration is changing in various ways, according to the changes in hardware and software technology, and the changes to the user environment. As the use of mobile devices, such as smartphones, has recently increased, most systems provide services based on the system architecture centered around a web service. Moreover, the system layout is also changing into a centralized structure, as the cloud services have expanded.

B) System architecture classification according to system layout

Architecture types can be classified according to the information system layout. The centralized architecture arranges all the systems in a centralized place. Multi-region distributed architecture operates the system by distributing regional and application systems regionally.

① Centralized architecture

A centralized architecture stores and operates the system and data from an integrated center. It configures an integrated database on a large-capacity server and it has a relatively simple system configuration. Having an integrated database makes it easy to assure data integrity, and having all components in one place makes them convenient to manage and operate. It can also quickly respond to system failures. However, all business services are terminated if the integrated center fails. Moreover, the load is concentrated during peak time.

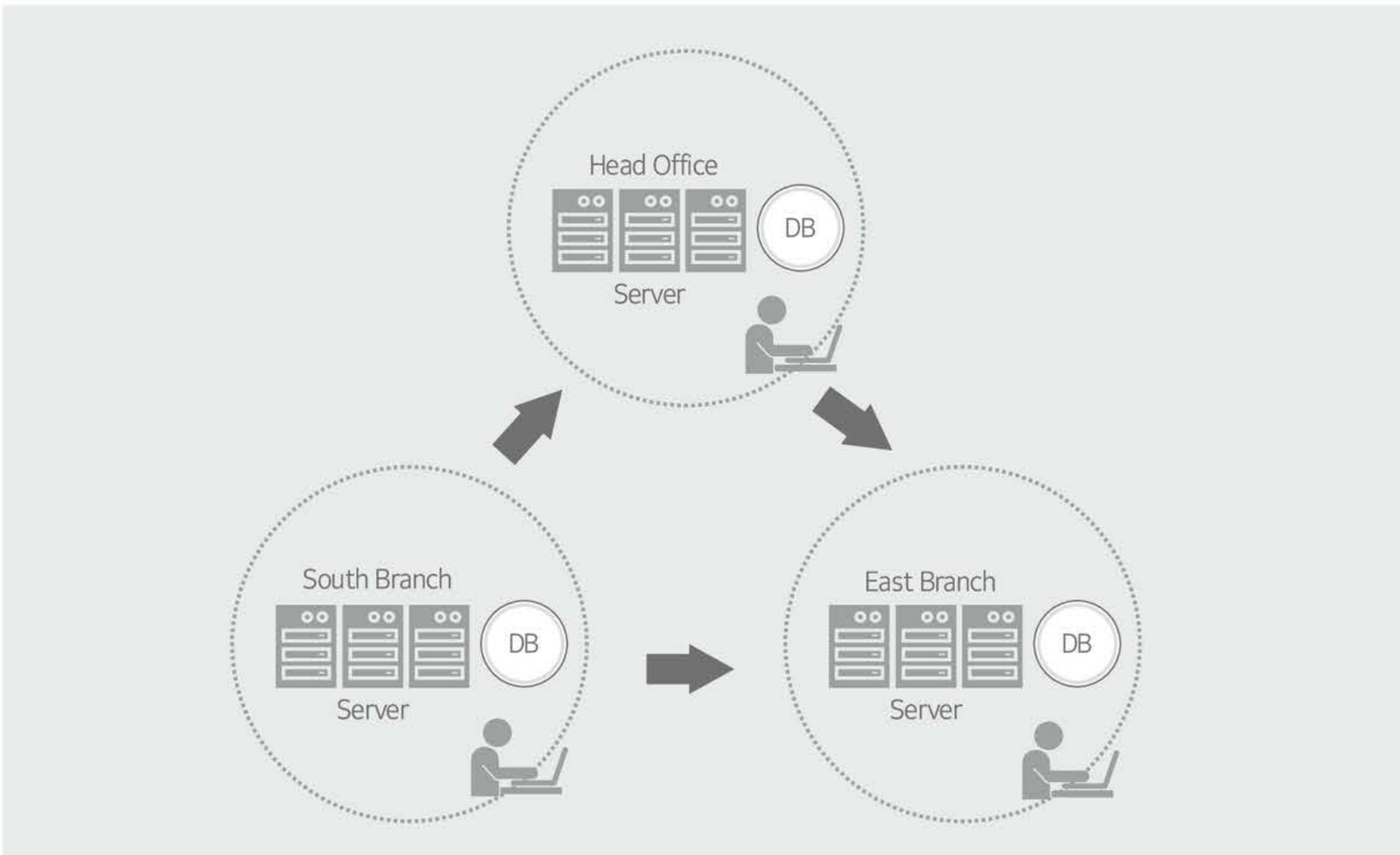


[Figure 4] Centralized Architecture

- Multi-region distributed system architecture

The multi-region distributed architecture operates the systems, and the application systems are distributed regionally. Each region manages the distributed data using small or medium-sized servers. This structure can reduce the load of each server, since it distributes the user load. When there is a failure, only the regional business services

are terminated. However, the distributed databases make it difficult to manage the data integrity, and both the system configuration and management become complex.

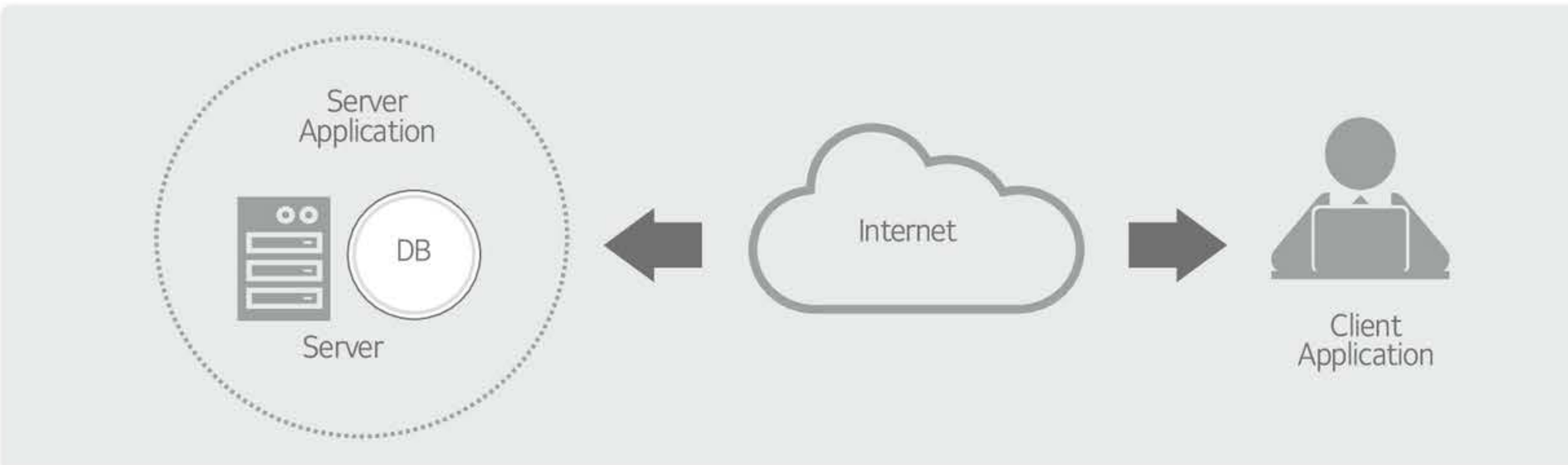


[Figure 5] Multi-region Distributed System Architecture

C) Classification according to how the application programs are provided

① Client-server architecture

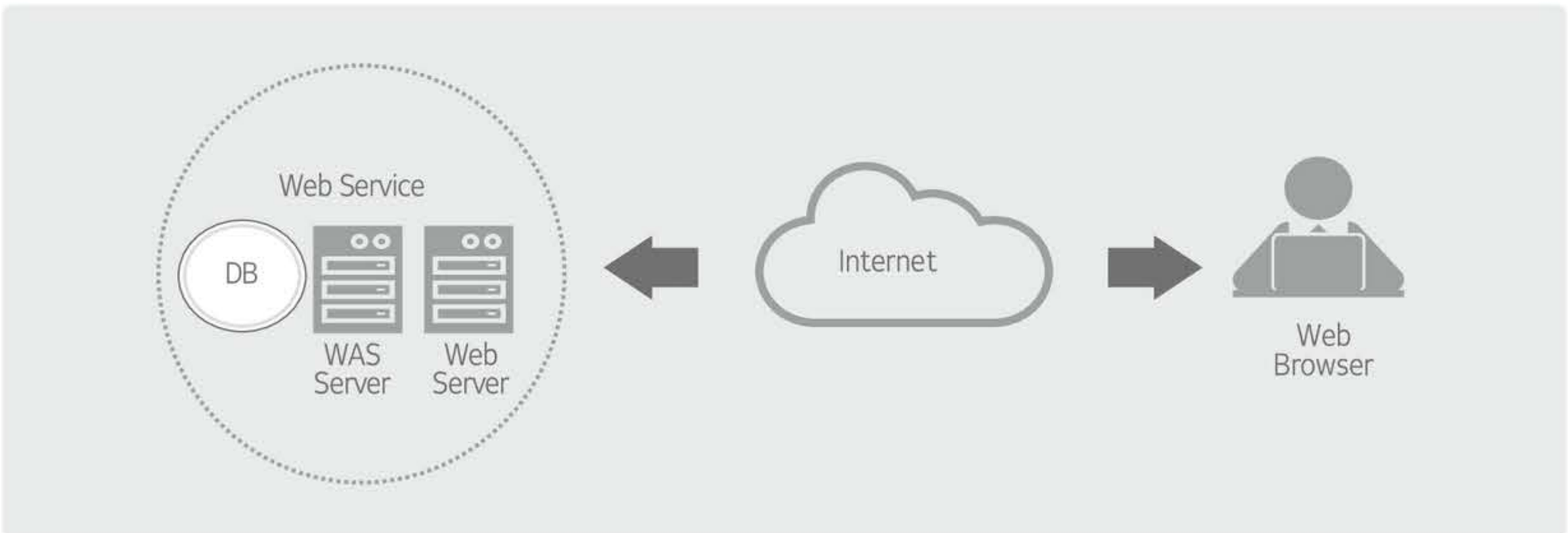
The architecture places system functions in the servers and clients and configures them to use the service, depending on the business size or environment. The server and client's role differs, depending on where the system functions, such as data management, business processes, and presentation, are located. It can provide the convenience of a graphical user interface by running the application program from the client side. However, the configuration is complex, and the development and management are difficult because the architecture is separated into servers and clients. Service types include games, chatting, messenger, FTP servers, and terminal servers.



[Figure 6] Client-Server Architecture

• Web system architecture

The architecture runs the server's application programs, and the client uses the service with a web browser. It generally consists of a webserver, a web application server, and a database server. The middleware assures stable performance, and the program reusability is high. Although the web system architecture was limited by only providing web services with a static display, the introduction of HTML5, and the performance enhancement of PC environments, have enabled them to provide services in an impressive GUI environment. Most recent information systems use web system architecture. The client can use all the services with a PC or mobile device, if the environment runs a web browser.



[Figure 7] Web System Architecture

D) Classification by system layer

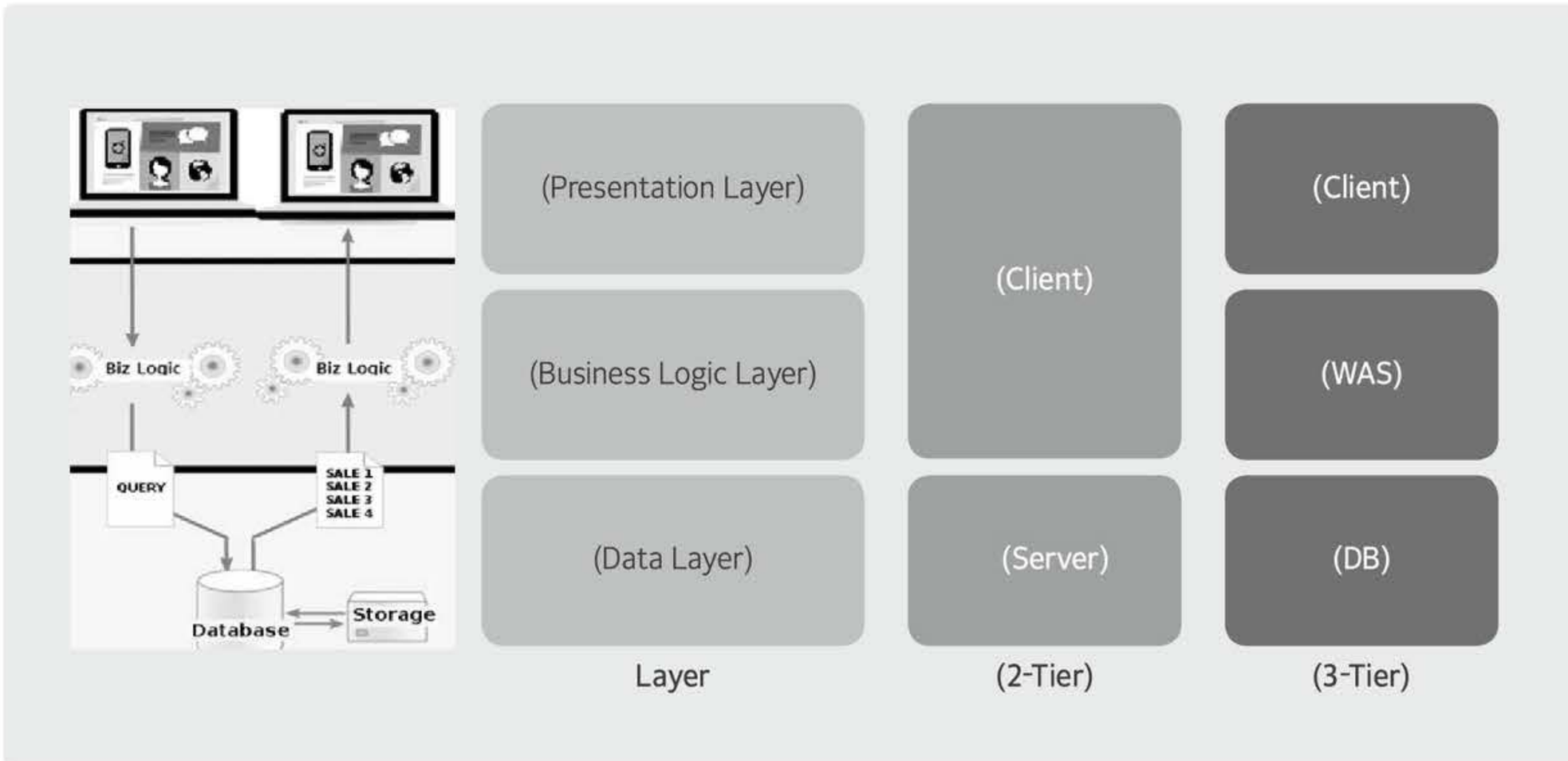
The function and role of each system component are classified by tier, or layer. The most widely known layer is the OSI 7 layer, which shows the network structure. Layers and tiers are considered similar concepts because they both represent functions and rules in a tiered structure. The most common difference is that layers refer to structures from a logical point of view, and tiers refer to structures from a physical perspective. An information system is divided into a presentation layer, a business logic layer, and a data layer. This logical layer architecture can be divided into an N-tier structure as a physical system architecture. The typical N-tier structures are a two-tier structure and a three-tier structure.

Type	Description
Presentation layer	This layer is at the highest level of an application program. It provides the service information and the interface to users. This layer is also called GUI or front-end. In a physical tier structure, webservers and client's web browsers belong to this layer.
Business logic layer	This layer runs the program information processing rules that fulfill business requirements (business logic). It determines which data is needed according to the business logic, and it refers them to the data tier. This layer is called the transaction tier. In a physical tier structure, application program servers, such as WAS, belong to this layer.
Data layer	This layer accesses data resources, such as the database, and reads or writes data. The data layer may be divided into the data access layer and the data store layer. In a physical tier structure, the database server or file system belongs to this layer.

• Two-tier architecture

Two-tier architecture stores and processes data in the server and processes business logic and presentation for the

client. It is generally the same structure as the client-server architecture. Since the client processes most logic, it can be fast and easily implemented when the number of users is small. However, the performance becomes poor as the number of users increases, and the architecture is disadvantageous in terms of scalability, reusability, and resource use. Actual cases include application implemented with 4GL tools, such as PowerBuilder, Visual Basic, and Delphi.



[Figure 8] Types of Information System Tier Architectures

- Three-tier architecture

  Three-tier architecture is a structure created to overcome the limitations of two-tier architecture, and it is also called multi-tier architecture. This structure has an additional tier between the presentation tier and the data tier, in order to process business logic to be flexible and scalable. The problems of two-tier structures include security problems due to the clients' ability to access the database through the network, the difficulty of user authority management, and the difficulty of changing programs. In three-tier architecture, the client does not have database access information. It is easy to manage the database object reference authority through the application server. The distribution of the separate presentation tier becomes flexible. However, the development environment is complex, and the cost increases because of the need to procure middleware and additional hardware.

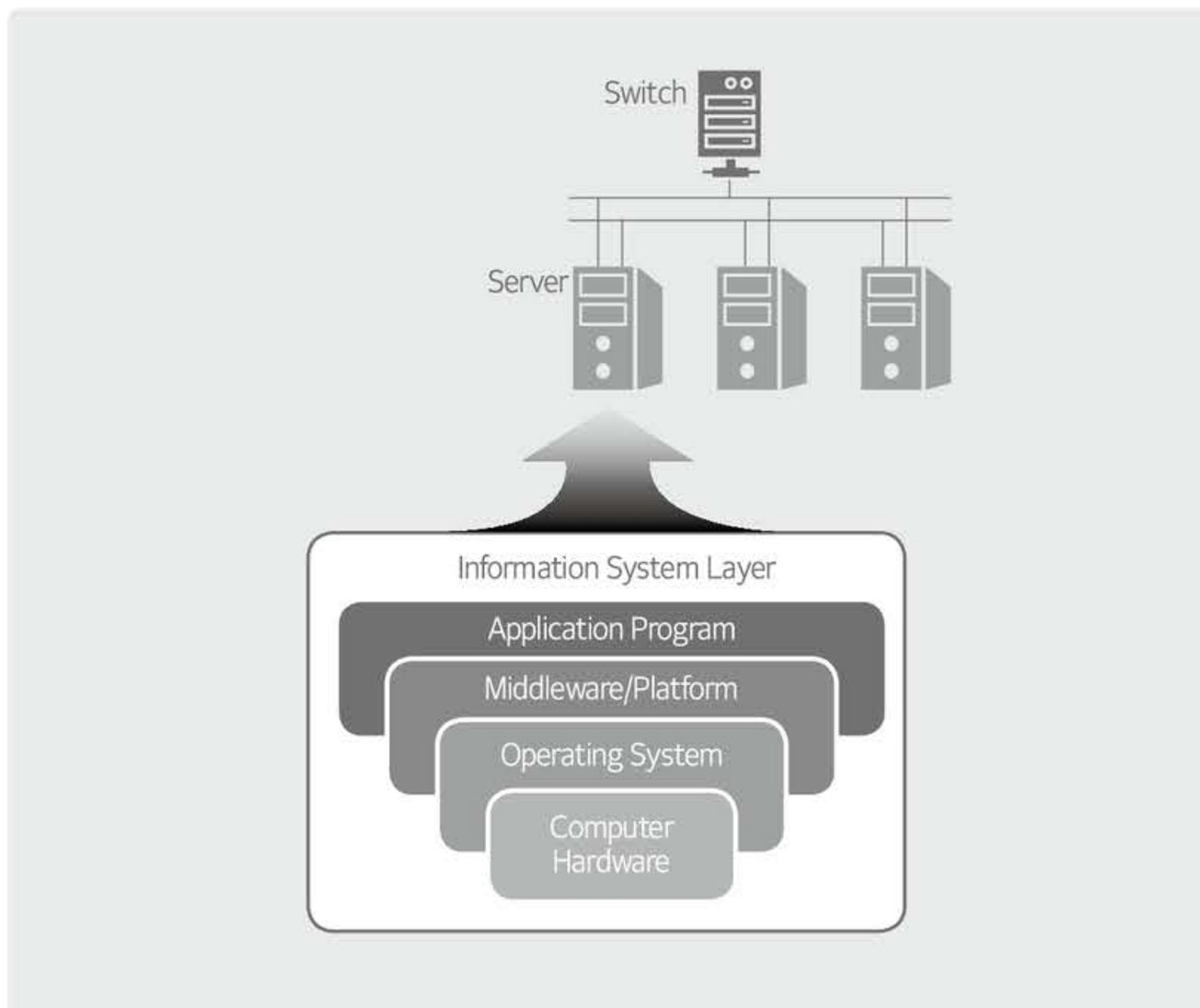
03 Server's Stack Architecture

A) Concept of server's stack architecture

"Server" refers to a computer or program that provides information or services to users through a network. It provides the computer power needed for the information system services. The concept of the server was created when IBM started making a server called the mainframe. The term mainframe refers to a large steel enclosure, like a cabinet,

that houses the CPU and the main memory. The mainframe features excellent stability and performance, but it is very expensive because an enterprise made all the elements from the CPU to OS, the development language, and the application programs. The mainframe began downsizing gradually in 1997 when the attention to super Unix servers, such as HP, Sun, and Silicon Graphics, increased. Downsizing refers to replacing a mainframe server with a super Unix server. While Unix servers were generally equipped with a Power CPU, it is worth noting that more mid- to large-sized servers began featuring Pentium CPU in the server market in 1997. It was around this time that Windows NT servers, using Intel CPU, were on the sharp rise. Twenty years later, in 2017, x86 servers, featuring Intel CPU, are dominating the Unix server market. Currently, the server market is divided into x86 servers and non-x86 servers. Although the market is divided into two major server types in terms of sales,x86 servers account for more than 70% when comparing the number of server units sold.

This server has a stack structure. The computer hardware, consisting of a CPU, main memory, and auxiliary memory is at the bottom. An OS that abstracts computer hardware and provides services to software is located above it. On top of the OS, middleware, such as a webserver, WAS, and DBMS server to support the flexibility of convenient expansion, as well as the reduction of application programs and independence of computer hardware, is located. Application programs provide services to users using this middleware.

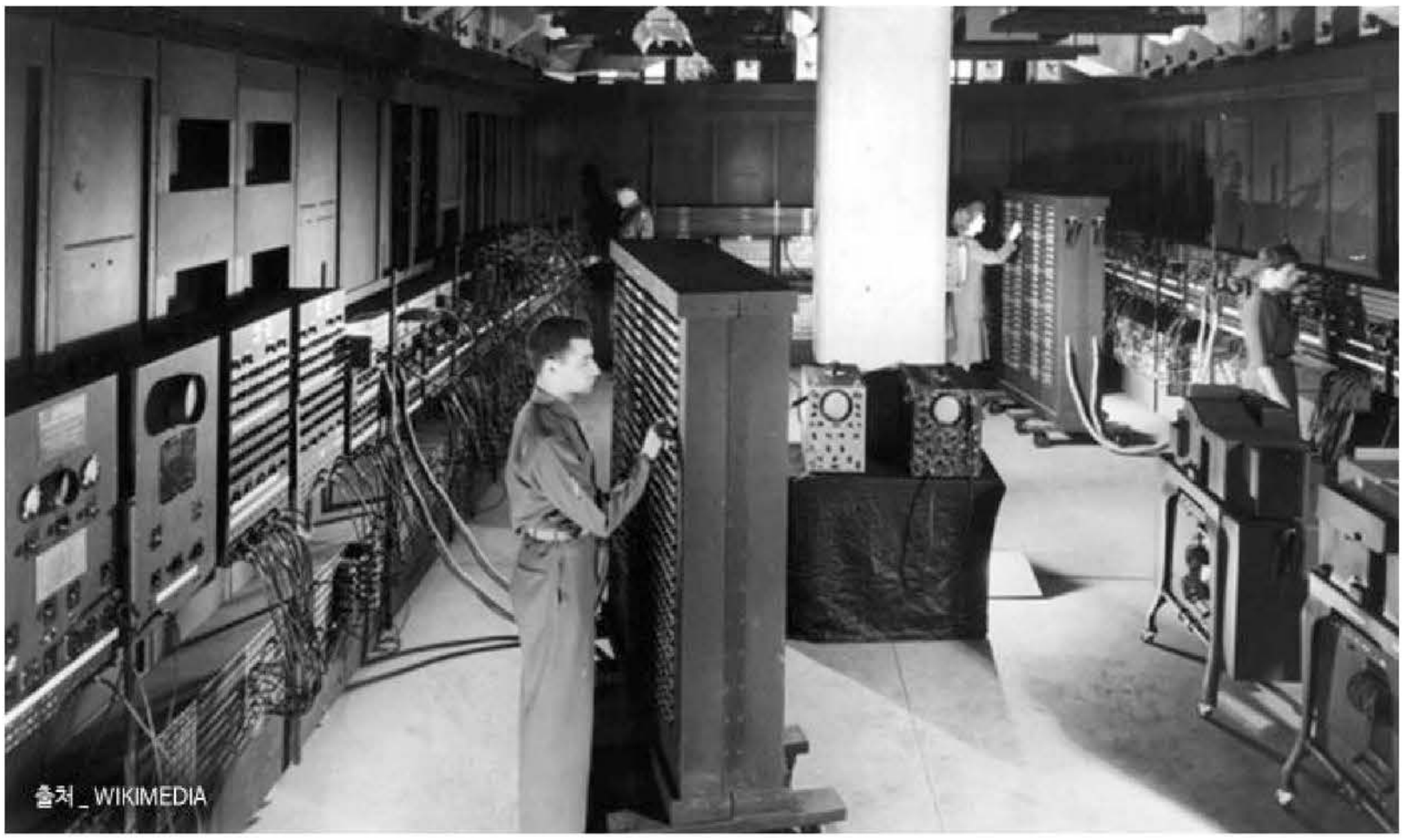


[Figure 9] Information System Server's Stack Structure

B) Computer hardware

While some argue that an abacus used in calculations in ancient times is the origin of computers, the Turing Machine, devised by the English mathematician Allen Turing in 1936, is generally regarded as the beginning of modern computers. This Turing Machine proved the mathematical concepts and calculation processes. Afterward, ENIAC, created in 1946, is known as the first programmable general-purpose computer. Using 18,000 vacuum tubes, it weighed 30 tons and was literally as large as a house. Later, EDSAC, the first computer that embedded the program

in a memory unit, as proposed by John von Neumann, was developed in 1949. While computers were mostly used for military purposes in the past, the popularization of computers began in the 1980s when a PC, developed by IBM, and the Windows operating system, developed by MS, were successful. In recent years, mobile computers installed in smartphones have become popular and opened the era of handheld computers.



[Figure 10] ENIAC, the First Electronic Vacuum Tube Computer

II. Network Concept

Recent trends and issues

In network technology have evolved from data communication, through existing 56kbps wired media, to wireless networks and mobile communication of hundreds of Mbps and even Gbps. Moreover, the advent of various connected terminals has led people to in an inseparable relationship with communication in their daily lives, as well as in more diverse business areas. Accordingly, software used in a terminal or system has a close relationship with a network, and it is based on mutual communication. It is necessary to understand and utilize the standard protocols used by networks to design and develop organic and optimized software, based on this network concept.

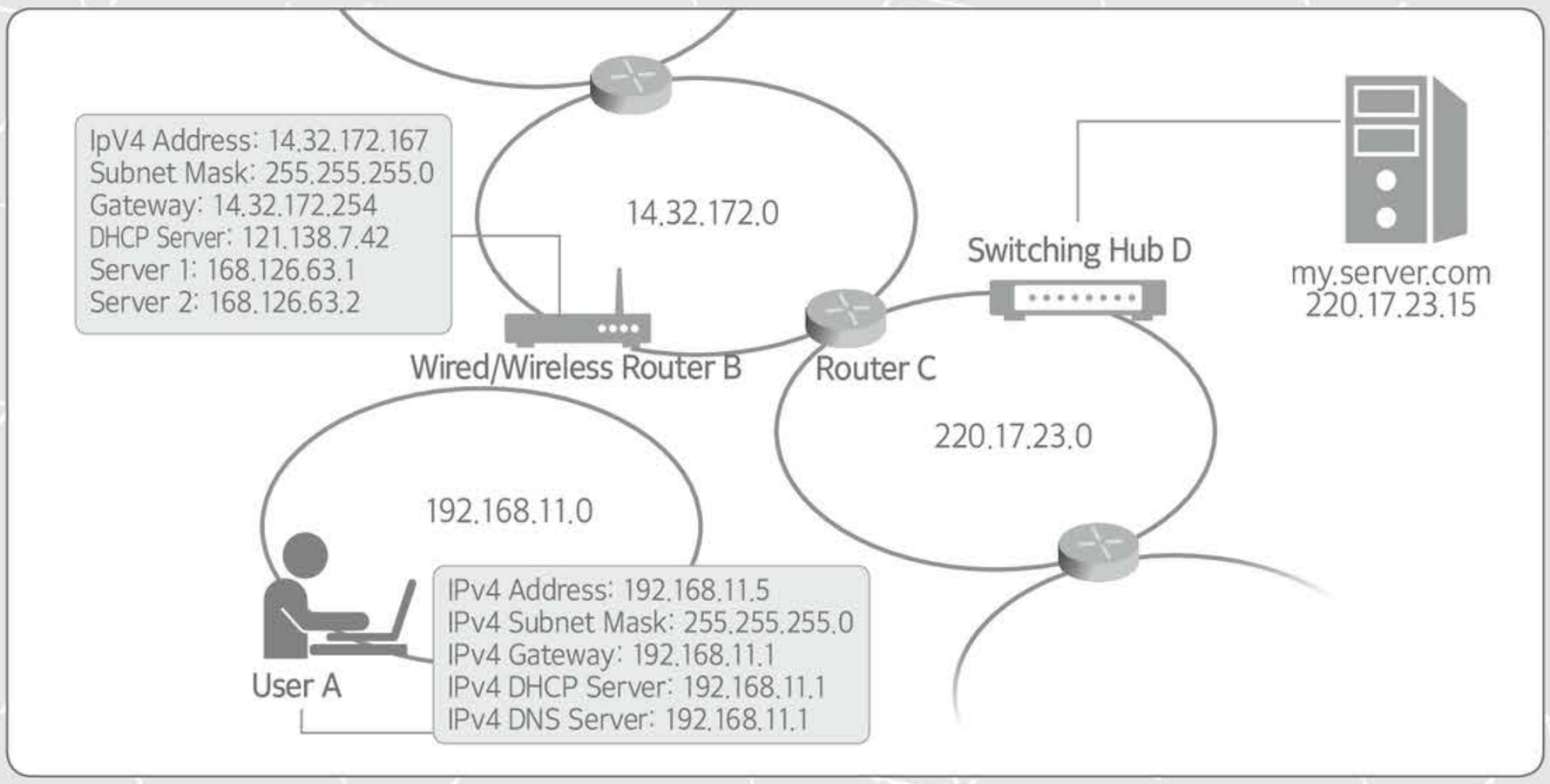
Learning objectives

1. To be able to explain the concept of the Internet and the open protocol.
2. To be able to explain how the OS manages processes.
3. To be able to explain the concept of virtual memory and management techniques.
4. To be able to explain the storage unit, file systems, and input/output (I/O).
5. To be able to explain the latest OS technologies and trends.

Keywords

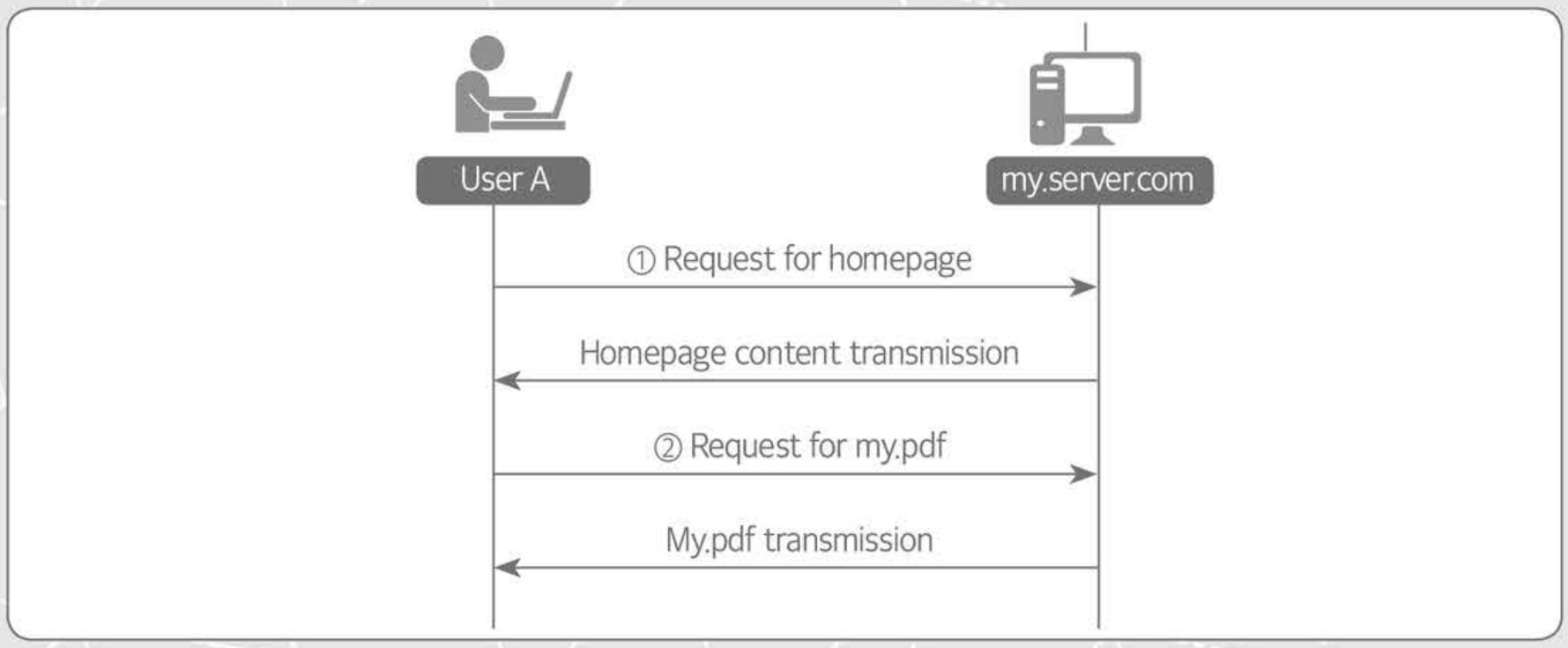
Protocol, Internet, IETF, 3GPP, 3GPP2, ITU-T, OSI reference model, Application layer, Presentation layer, Session layer, Transport layer, Network layer, Datalink layer, Physical layer, Internet protocol layer, Application layer, Internet layer, Network interface layer, Internet address scheme, MAC address, IP address, Port number, IPv4 address, IPv6 address, Private network, NAT, CIDR, RFC

✚ Preview for practical business How to understand networks



[Figure 10] Network between Users and Servers

[Figure 10] shows a virtual network connection in a situation where user A connects to a webserver (my.server.com) and uses a computer to locate the data. Although the situation frequently occurs inhomes, schools, or workplaces, many people use their computer network information without knowing it. The following example shows a situation of using a network service in detail.



[Figure 11] Web Use by User A

[Figure 11] shows user a connecting to a webserver (my.server.com) and searching for the data. <Table 1> describes, in detail, which changes occur in the network at each stage of the operation in [Figure 11]. The standard protocol used at each stage is described in parentheses, and its concept and detailed operation are described in the following chapters.
This example shows how a simple network service, as shown in [Figure 11], is executed.

Network operation procedure to "①Homepage request"

〈Table 1〉 Web Use Scenario by User A

Step 1) User A enters http://my.server.com/ on the web browser and presses the Enter key.

Step 2) The web browser intends to send User A's request to the webserver, my.server.com, but cannot do it because my.server.com is an arbitrary domain name for users to recognize easily. The web browser requests the IP address of my.server.com to the domain name server to solve the problem (DNS).

Step 3) On receiving the IP address of my.server.com, 220.17.23.15, from the DNS, the web browser connects to the webserver using the address and requests data in the document route (HTTP).

Step 3-1 The web browser hands over the requested information to the TCP protocol to request my.server.com. The information is properly capsulized and delivered to the webserver program of my.server.com (TCP).

Step 3-2 Information created by the TCP protocol is capsulized to be routed to the 220.17.23.0 network, in which my.server.com is present, through several hoops (IP).

Step 3-3 An initial packet that contains the information for routing between networks is created. However, it must be further modified to be transferred between networks. The IPv4 address of user A's computer is 192.168.11.5 and cannot be transmitted to my.server.com directly. It must be transmitted to the computer's gateway wired/wireless router B (192.168.11.1). However, an IP address cannot be used for data transfer within a network, so the MAC address is needed for this. After finding the MAC address corresponding to 192.168.11.1, the gateway's IP address, through network broadcasting, the packet is transmitted to wired/wireless router B (ARP, Ethernet, private network, and NAT).

Step 3-4) On receiving the packet from user A's computer, wired/wireless router B, aware that the packet is not sent to the router as the final destination, examines the address to which the packet must be transmitted. The IP header indicates that the packet's final destination is my.server.com, and since it does not belong to network 192.168.11.0 or 14.32.172.0, to which wired/wireless router B belongs, it transmits the packet to router C, to which it belongs (routing protocol).

Step 3-5) On receiving the packet, router C recognizes that the packet is sent to network 220.17.23.0, to which the router belongs, and intends to transmit it to the webserver's IP address 220.17.23.15. As in Step 3.3, since the packet movement in a network requires the physical address of my.server.com, the MAC address, it obtains the MAC address corresponding to the IP address of my.server.com, 220.17.23.15, though the network broadcasting, and transmits the packet (ARP).

Step 3-6) On receiving the packet, my.server.com examines the IP address and recognizes that the packet was sent to it. It removes the header data, such as the IP address, and sends the remaining data up to the higher protocol layer (IP).

Step 3-7) It determines which application of my.server.com to which the packet should be transmitted, removes unneeded data, and sends the remaining data to the intended webserver application (TCP).

Step 3-8) The webserver application recognizes that the message is requesting information in the document route, and it sends the information to user A's computer.

Step 4) On receiving information in the document route from the webserver, the web browser displays the contents for user A's viewing. The user can select the desired hyperlink in the document to view, including videos on a web browser (HTTP, HTML).

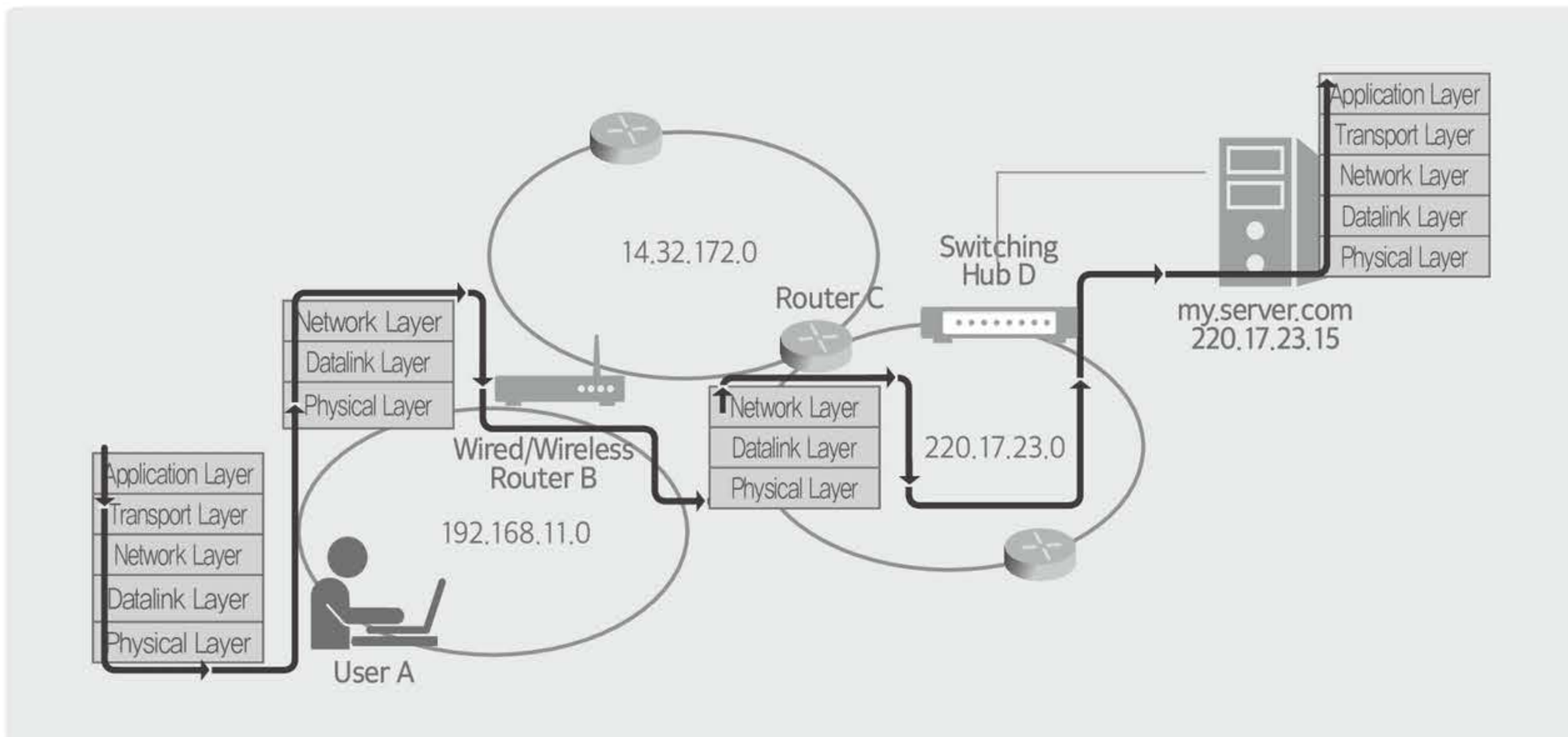
01 What is the Protocol?

<Table 1> above showed many standard protocols, such as ARP, IP, TCP, HTTP, DNS, and NAT. A protocol is a standardized communication rule for sending and receiving data through a network. The Internet, as we know, is a large group of networks that connect multiple PCs and servers.
For tens or hundreds of thousands of PCs, servers, and routers that connect them to exchange data, it is necessary to have a common protocol that can be found by each other, and there are several international organizations that standardize these protocols.

- IETF (Internet Engineering Task Force) is responsible for most Internet-related standard protocols.
- 3GPP (Third Generation Partnership Project) and 3GPP2 are responsible for wireless communication protocols such as GSM, CDMA, UMTS, LTE, and LTE-A.
- ITU-T (International Telecommunication Union Telecommunication Standardization Sector) is responsible for protocols related to telephones.

Therefore, it is necessary to have the competitiveness to understand and implement (if necessary) the standard protocols enacted by these standardization organizations to develop software in the network/communication sector.

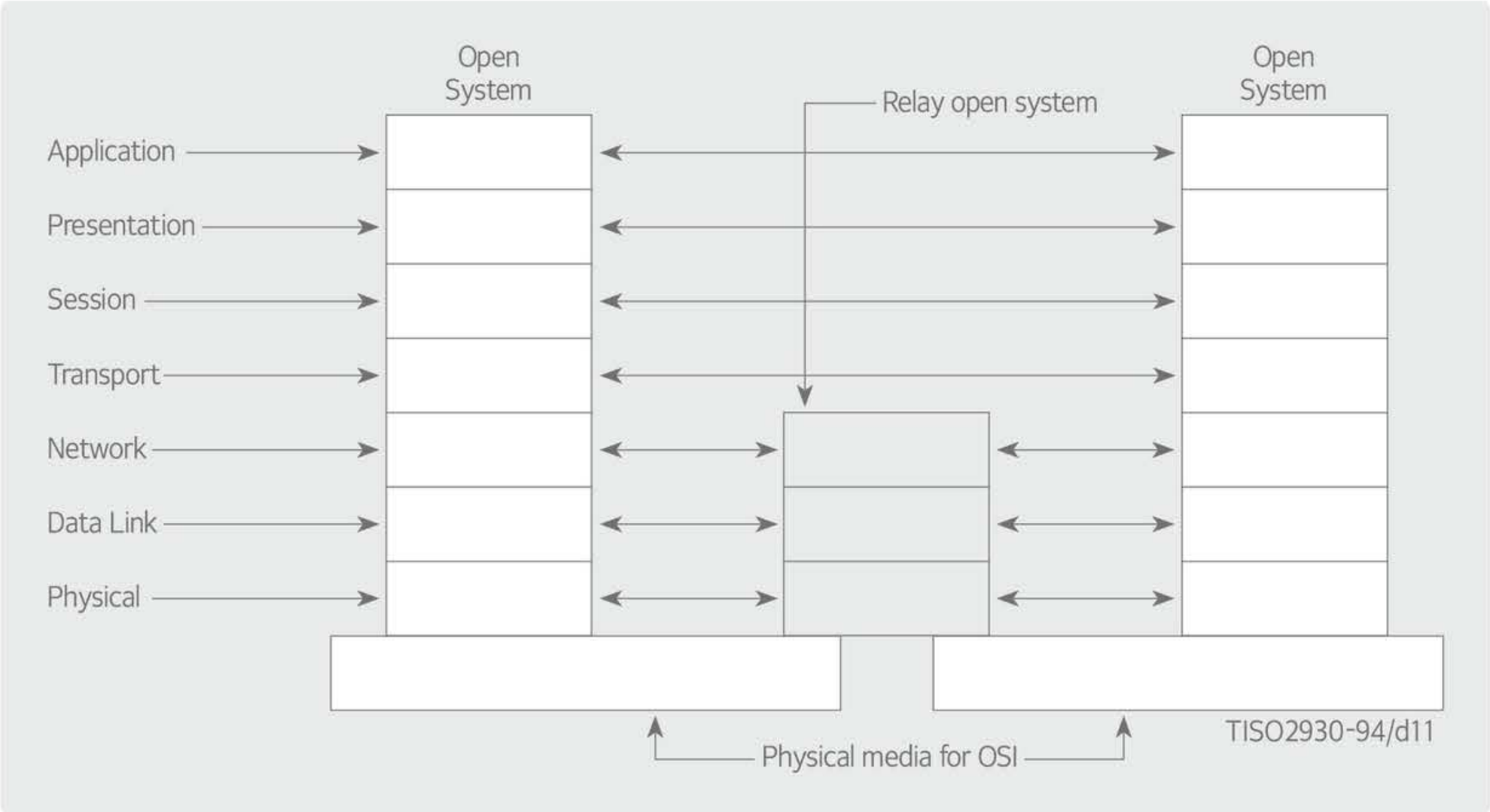
[Figure 12] below shows the details of [Figure 11] by reflecting the network protocol concepts described above. When a user searches through a web browser, data requests and response data flow in the network through devices, such as the user's computer, a wired/wireless router, a switching hub, and a router, using the related protocols. While the physical layer is directly connected with hardware, such as wires and switches, and operates with electric signals, the protocol layer above the data link layer is implemented and operated in software. Understanding how many software components are connected to each other for operation should enable application programs or system software over a network. Later sections describe in detail how each protocol layer shown in [Figure 12] works with the others in each stage.



[Figure 12] Virtual Network Reflecting Protocol Layer Concept

02 OSI Reference Model and TCP/IP Protocol Layer Structure

[Figure 13] shows the concept of the OSI reference model communication, which includes the open relay system described in the standard document ISO/IEC 7498-1:1994(E) [8]. The diagram is often used when discussing ISO's OSI reference model, or OSI layer, which is commonly referred to when dealing with networks. Here, ISO is derived from the Greek word "ισος (isos)", meaning "identical", and should not be confused with the English name of the International Organization for Standardization (IOS). OSI is an abbreviation of Open Systems Interconnection.



[Figure 13] Communication Concept of OSI Reference Model Containing Relay Open System

<Table 2> shows in detail the protocol and its function of each OSI reference model layer [1].

〈Table 2〉 Protocol and Function of Each Layer of the OSI Reference Model

Layer	Protocol	Function
Application layer	HTTP, SMTP, SNMP, FTP, Telent, etc.	The application layer provides services, such as a user interface, email, and database management.
Presentation layer	JPEG, MPEG, XDR, etc.	The presentation layer supports the syntax of data exchanged between two systems and the encoding translation and encryption related to the semantic.
Session layer	TLS, SSH, RPC, NetBIOS, etc.	The session layer is for communication sessions. It establishes, maintains, and synchronizes the interaction between communication devices.
Transport layer	TCP, UDP, SCTP 등	The transport layer provides functions like the reliable message transfer between end-to-end processes, as well as error control.
Network layer	IP, IPX, ICMP, X.25, ARP, OSPF, etc.	The network layer supports packet transmission between networks from the sending point to the destination.

Data link layer	Ethernet, token ring, wireless LAN, etc.	The data link layer provides the function of transferring frames between hops without error.
Physical layer	Radio wave, optical fiber, PSTN, etc.	The physical layer transfers the actual bits through physical media.

As shown in [Figure 5], the Internet protocol layer is based on the ARPANET reference model, which was released before the OSI layer model [3][9][10]. While documents use slightly different terms for each layer, this section uses the most commonly used terms.

① Application layer

Application programs, such as web (HTTP), DNS, telnet, FTP, and email sending/receiving (SMTP/POP3/IMAP4), allow access to services on other layers. In the OSI reference model, this layer's concept includes the presentation layer and the session layer.

② Transport layer

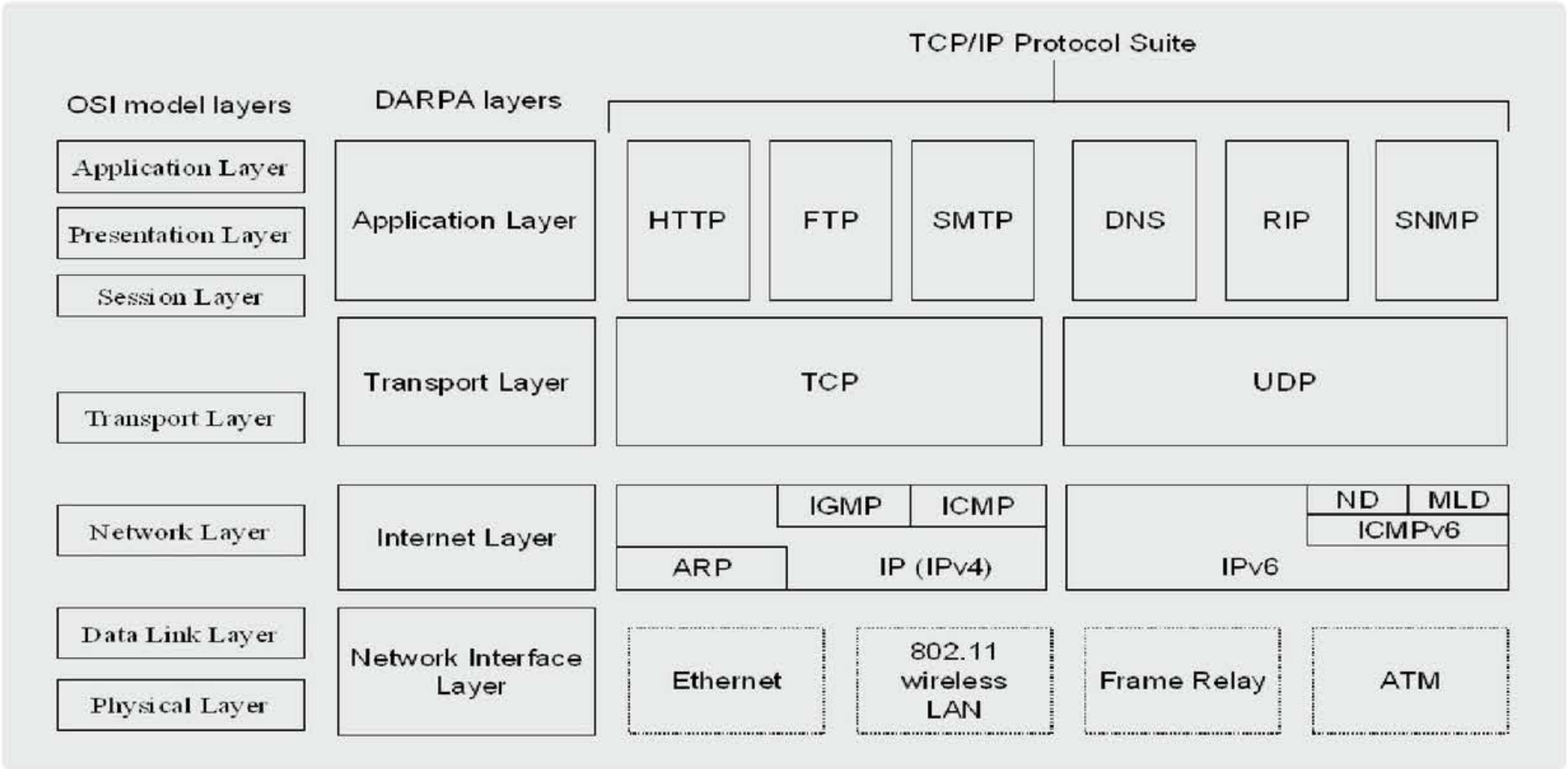
Also called the host-to-host transport layer, the transport layer is responsible for data exchange between abstract ports, managed by application programs such as TCP, UDP, and SCTP protocols. It is equivalent to the transport layer in the OSI reference layer.

③ Internet layer

Also called the network layer, it is responsible for addressing and routing functions. It is equivalent to the network layer in the OSI reference layer.

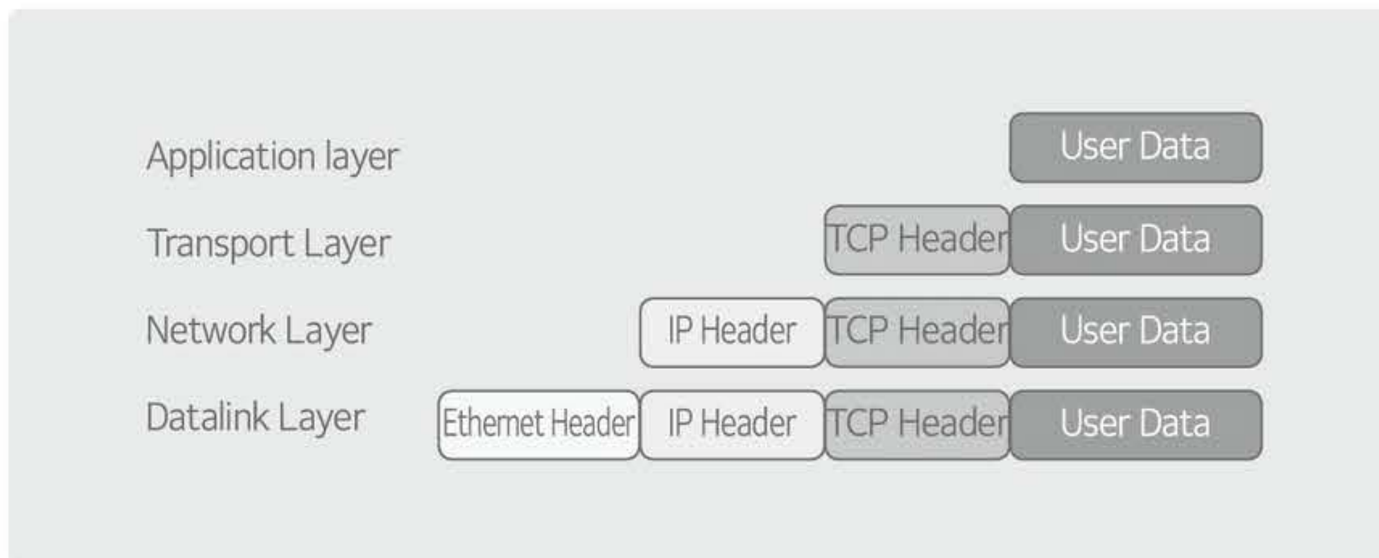
④ Network interface layer

Also called the network access layer, the network interface layer has the role of actually sending and receiving TCP/IP packets through physical media, such as IEEE 802.3 and IEEE 802.11 WiFi. It includes the data link layer responsible for the MAC function in the OSI reference model, as well as the physical layer that defines electric signals.



[Figure 14] TCP/IP Protocol Layers: Based on the ARPANET Reference Model [10]

As shown in [Figure 13], these protocols specify communication with the same protocol layers of other hosts connected through a network. For communication, a header for the exchanging of data between protocols of the same layer with data downloaded from the layer immediately above are generally required.
As shown in [Figure 15], an application layer can transmit the user data through a transport protocol TCP (or UDP), which is an end-to-end protocol connected to the other host. It requires a TCP header (or UDP header).
The transmission of user data, including a TCP header, uses IP, which is a network layer protocol that has routing information to the network of the other host, and the IP head has the information.
The network layer considers both the TCP header and the user data received from the transport layer as data. The data link protocol, which has the necessary information for physical delivery to other hosts in the same network, mostly uses Ethernet. In that case, both the TCP header and the user data, including the Ethernet header with the necessary information and the IP header downloaded from the network layer, are regarded as data. It should be noted that each layer uses only the header that corresponds to the given function.



[Figure 15] Example of Network Protocol Header and Data Configuration

03 Internet Address System

Internet users come across many terms related to numbers, such as a MAC address, an IP address, and a port number. It is unnecessary to memorize these numbers, but understanding the numbers helps figure out each layer protocol. The MAC address, IP address, and port number in the Internet address system have the following meanings:

① MAC address: The MAC (medium access control) address, or the address system used by the data link layer, is used to transfer frames between the physically connected nodes. The information is generally stored on the LAN expansion card hardware, called NIC. It is also called the physical address or Ethernet hardware address (EHA). It is formed by48 bits. The first 24 bits represent the hardware manufacturer's unique ID, and the remaining 24 bits represent the NIC-dependent address.

② IP address: The IP address is the address system of the network layer to transfer datagram between two hosts/ routers. It facilitates data to be transferred from the source node to the destination node, through multiple Internet networks. While the MAC address is a physical address, the IP address is a logical address. Existing IP addresses, or

IPv4 addresses based on the 32-bit address system, faced exhaustion as Internet use increased rapidly worldwide. Although the IPv6 address system based on 128 bits was introduced to solve the problem, IPv4 addresses are still mainly used. Although the IANA (Internet Assigned Numbers Authority) was responsible for IP address allocation in the past, the ICANN (International Corporation for Assigned Names and Numbers) is currently in charge.

③ Port number: The port number used to transfer messages between two processes (running applications) is responsible for the connection between the web browser on user A's computer, shown in [Figure 10], and the webserver program on my.server.com. The 16-bit number system can identify up to 65,535 application programs. The port numbers from 0 to 1023 are called well-known ports and are allocated by the IANA. Application programs, such as telnet and email, use it for sending and receiving data. The port numbers from 1024 to 49,151 are called registration ports and, like well-known ports, are mostly used by servers. The Internet chatting protocol, IRC (TCP port number 6667), and Git version control system (TCP port number 9418) are examples. The port numbers from 49,152 to 65,535 are mostly used as a temporary number by clients, and are called dynamic ports or temporary ports.

04 Internet Standard

The IETF (Internet Engineering Task Force) is a leading organization responsible for Internet-related standards. It defines the basic protocols for Internet transmissions, such as IP, TCP, and UDP, as well as the most widely used protocols in the Internet application layer protocols, such as HTTP, SSH, FTP, SMTP, POP3, and IMAP. <Table 3> lists other international organizations responsible for Internet-related standardization.

〈Table 3〉 Internet Standardization Organizations

Standardization Organization	Description	Enactment Standard (Example)
ISO	• International Organization for Standardization • An international organization established in 1946 for internal cooperation in intellectual activities, or for scientific, technological, and economic activities	OSI reference model
ITU-T	• International Telecommunications Union-Telecommunication Standards Sector • An organization responsible for telecommunication standardization under ITU	ITU-T G.711
IEEE	• Institute of Electrical and Electronics Engineers • IA specialized organization, authorized by the US ANSI to develop the national standards in the United States	IEEE 802.3 IEEE 802.11 IEEE 802.15
EIA	• Electronic Industries Association • An organization that plans to unify standards, such as dimensions, measurement methods, and labeling methods, for IEEE	TIA/EIA-568-B(T568B)
W3C	• World Wide Web Consortium • An international consortium related to the Internet, established in 1994, for the long-term development of the web by establishing web standards	HTML 4.01/5 CSS 2/3
OMA	• Open Mobile Alliance • A forum to develop technical specifications and to verify interoperability for the global promotion of mobile data services	Push-to-talk over Cellular (PoC)

III. Operating System (OS)

Recent trends and issues

Merely ten years ago, nobody could have imagined web surfing, shopping, or watching movies in bed. The introduction of the smartphone made these things possible, as it was a computer for your hand. A smartphone is also a computer system that runs on the OS. Most mobile operating systems have recently disappeared, and iOS and Android have become the two dominant operating systems on the market. These two operating systems are in a heated battle for the diversity of functions and user-friendliness. Apple and Google, who provide these two operating systems, are two global enterprises among the top ten global companies in market value. In the past, when smartphones were unknown, Microsoft, who provided operating systems for personal computers, and IBM, who provided operating systems for servers, such as mainframes and UNIX, were in the top ten companies in the world in terms of market value. The operating system (OS) is the base platform of a computer system for providing information services. Any company that can monopolize this platform can gain tremendous added value using this platform. As such, the essential knowledge for understanding the system architecture, is the operating system.

Learning objectives

1. To be able to explain the concept and role of the OS.
2. To be able to explain how the OS manages processes.
3. To be able to explain the concept of virtual memory and management technique.
4. To be able to explain the storage unit, file system, and input/output (I/O).
5. To be able to explain the latest OS technologies and trends.

Keywords

Flynn's taxonomy, parallel processing technology, parallel program technology, disk scheduling, SAN, NAS, DAS, LTO, VTL, RAID, GPGPU, video compression standards

✚ Preview for practical business **Operating systems, the start and end of computer operation**

As of 2016, Korea's computer user rate was 85.7%, meaning that more than 8 out of 10 people were using computers. During the same period, the smartphone user rate grew much faster than PCs, reaching 88.5%. With the rapid proliferation of mobile devices, the PC user rate is gradually decreasing, and the us of laptops,which have gradually replaced desktops, have also continued to decline after exceeding 30% for the first time in 2014. Smartphones and tablet PCs are increasingly taking over the place of computers.

Electronic devices, such as desktops, laptops, and smartphones, are collectively referred to as "computers" and they have things that are commonly applied to them. They contain indispensable software, an operating system (OS), and major hardware. Having the best processor and components is useless without an OS. The OS is a medium that delivers close to 100% of the outstanding performance of a manufacturer's computer to users.

The elements, such as the CPU, mainboard, memory, graphics card, and storage device, that make up a computer cannot do anything by themselves. Users can get work done with a word processor and can enjoy playing World of Warcraft, only when the OS gives the commands of what to do to each piece of hardware.

The operating system, which is software, is responsible for the device operation from start to finish. It is the OS's role from the moment it recognizes the power supply, until it shuts off the power on command. Without an OS, the user must manually turn the power on or off in an analog way, like the older products. Commands for device operation are mechanical, and the device can only perform one function with an input. In today's electronic devices, which have been mostly converted to operate electronically, the volume and power buttons remain manual.

(Source: EPNC, 2018/05)

01 Operating System (OS)

A) Overview of OS

OS is a system software that provides a computer resource interface to users or application programs by efficiently managing the limited resources of computer hardware.

The first computers were set to operate single-purpose programs, as they did not have an OS. The programs

ran slowly and were difficult to develop in this form, since the developer needed to implement every interface for accessing the computer's resources and allocating and recovering resources. OS implements input, control, memory, operation, and output functions to provide services in order to use computer resources efficiently. In other words, an OS controls the computer's resources and implements usage policies to allocate resources to users through scheduling. It exchanges data through input/output devices, or outputs exceptions or errors. The purposes of using the operating system are as follows.

- Abstraction: Provides standardized API to application programs by abstracting the complexity of the computer's hardware.
- Virtualization: Provides a virtualization function to enable application programs and users to share computer resources, and to use a single virtual computer hardware.
- Management: Maximizes computer resource performances and provides them to application programs, while meeting computer resource constraints.

B) Main functions of OS

OS abstracts and virtualizes computer hardware to provide resources to application programs, or to manage them efficiently. Therefore, OS can be understood by the functions of managing main hardware components. The main functions of OS are described as follows.

① Process management

In simple terms, a process is a running program. However, it is actually more than just a program, as a process is a system work unit that allocates, uses, and recovers various resources, such as the CPU, storage unit, files, and input/output units to perform its tasks. An OS manages processes and performs the following tasks:

- Creating and disposing of user and system processes
- Terminating and re-running processes
- Providing a technique for process communication and synchronization
- Providing techniques to prevent deadlock

② Main memory unit management

A hard disk (HDD) is the hardware that most people think of when they consider a storage unit. However, the main memory is a different type of storage. The main memory is a depository that allows the CPU to fetch commands and data to use from the instructions. The addresses that the CPU uses to access data are only available in the main memory. Even if the CPU intends to retrieve data in an HDD, it cannot read the data if it is not stored in the main memory. The main memory must handle the following management functions to provide data to the CPU:

- Tracking and managing the space and the users that use the storage device
- Deciding which process will occupy the memory space
- Allocating and recovering memory space

③ File management

OS can store data through various physical devices. It is necessary to abstract media with different physical characteristics, such as magnetic tape, magnetic disks, optical disks, and flash memory, for the OS to use them. File management provides the function of abstracting these physical characteristics into files, the logical storage units. It handles the following tasks:

- Creating and disposing of files
- Creating and disposing of the directory
- Providing primitive files and directory management
- Mapping for the OS to use the file in the auxiliary memory unit
- Storing the file in a nonvolatile storage medium

④ Input/output system management

The OS helps users use a device, even if they do not know its characteristics. Here, a device refers to an input/output device and includes a mouse, keyboard, printer, and speaker. Users can easily use these devices with the functions provided by the OS.

⑤ Auxiliary memory unit management

The volume of data used by a computer system is growing larger, from gigabytes and even to terabytes. However, the data used by a computer's CPU is that which is in the main memory unit. The main memory does not provide the capacity at the terabyte level. The OS hands the following auxiliary memory unit management function to store the managed data:

- Empty space management
- Storage allocation
- Disk scheduling

⑥ Networking

A distributed system is a set of memory units or separate processors. Each processor communicates at a high speed through the network. The network integrates scattered computer resources into a system to improve the calculation speed, share data, and to enhance reliability.

⑦ Command-interpreter system

A command-interpreter system is one of the system programs that enables users to access the main OS functions. In MS-DOS of UNIX shell, a user can enter a command with a keyboard in order to run an OS function.

C) Types of main OS

〈Table 4〉 Types of Main OS

Type	Operating system (OS)	Description
PC OS	Windows	- Developed by Microsoft - Stable and standardized GUI - Supported by many third party programs
	Mac OS	- Developed by Apple - Optimized for Apple's hardware
Mobile OS	Android	- Developed by Google (Acquired Android in 2005) - High openness - Installation using APK files
	iOS	- Developed by Apple - Very secure - Installation of apps only available from the AppStore

Server OS	UNIX	- Developed by AT&T - Multi-user environment - Commercial products mainly use the OS, as it is relatively inexpensive. Each server developer develops and installs a proprietary UNIX operating system.
	Linux	- Many open-source versions are available, and the source code is disclosed. - Low-cost implementation - Compatible with UNIX - Types: RedHat, CentOS, Ubuntu, Fedora, Suse, etc.
	Windows Server	- Developed by Microsoft - Windows interface to provide the same UI/UX - Supports many application programs

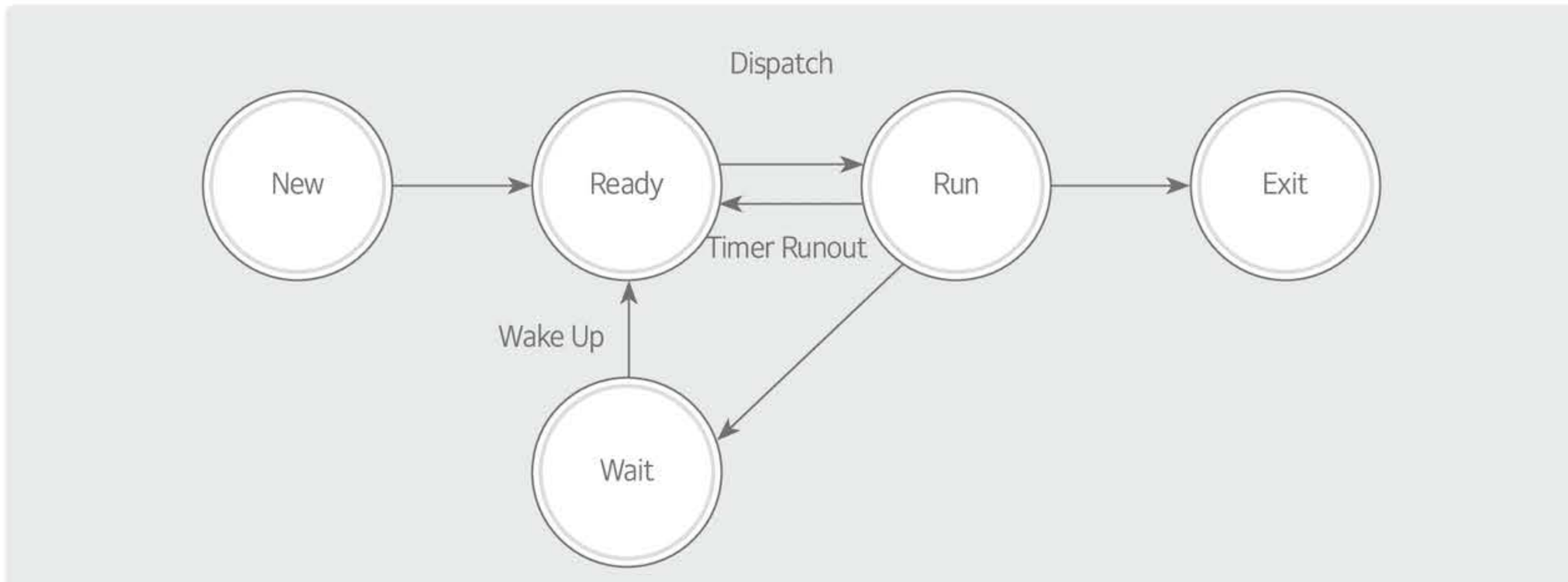
02 Process and Thread

A) Understanding of process

Early computer systems could only execute one application program at a time. In other words, one program occupied all the system's resources. However, modern computer systems run multiple programs concurrently. The OS must manage computer resources so that multiple programs share resources and operate in stable condition at the same time. A process refers to a running program, and in today's concurrent multi-process environment, it is a work unit of a time-sharing system.

① Process status

A process has a distinct process status during its lifecycle, as shown in [Figure 16].



[Figure 16] Process Status Change

- Created: A process is created but is not running by the OS.
- Preparing: The process is waiting for the CPU allocation to run.

- Running: The process has the CPU allocation.
- Finished: The process has completed its running, and the CPU allocation is released.
- Standby: The process ran after getting the CPU allocation, and is waiting for an event, such as the completion of input or output.

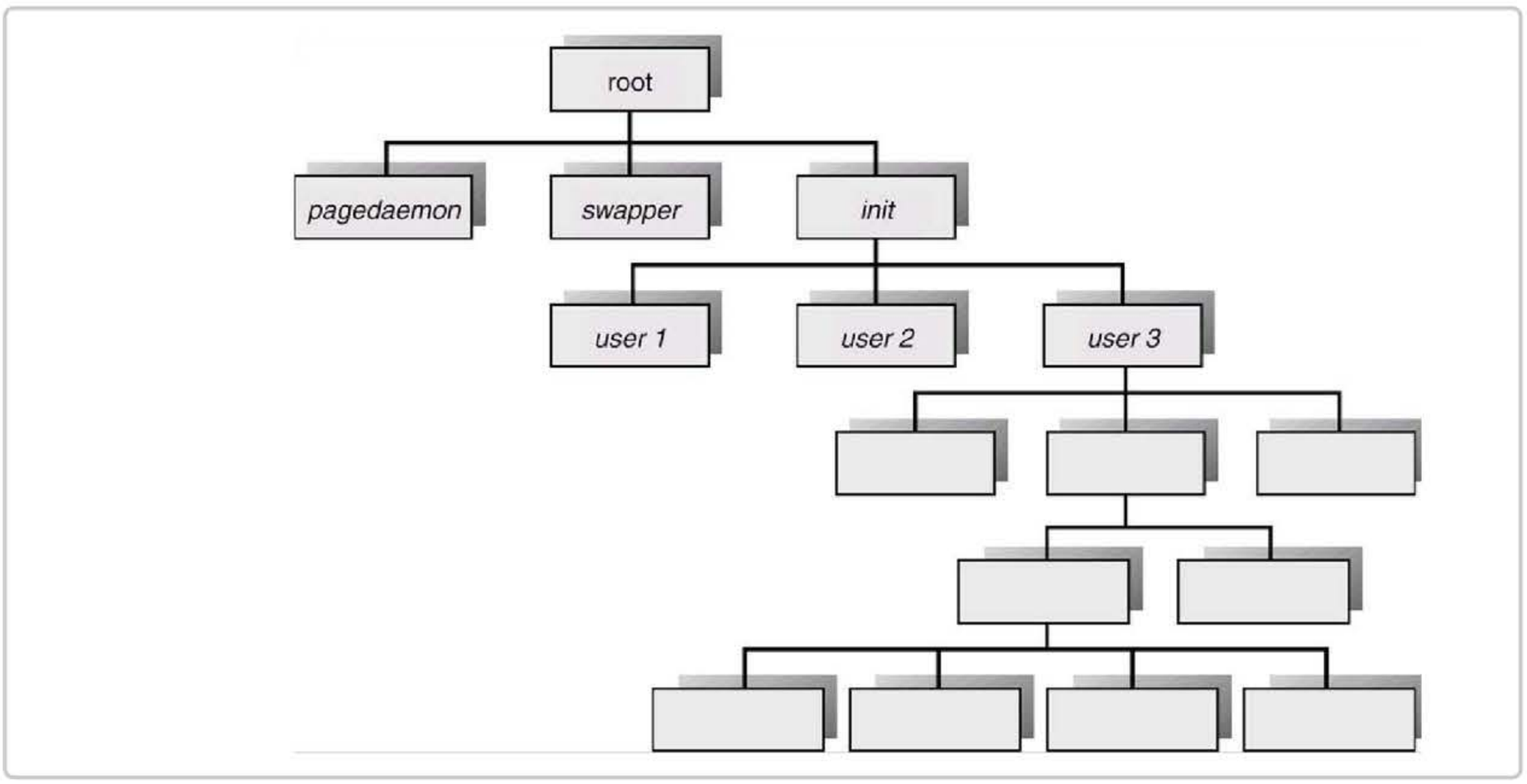
② Process control block (PCB)

PCB stores information necessary for process management. A unique PCB is generated when a process is created, and the PCB is removed from the process completion. A PCB includes a process identification number (PIC), a process status, a program counter (the value that indicates the instruction to run next), a scheduling priority, registered information, and main memory unit information.

B) Process management technique

① Process creation

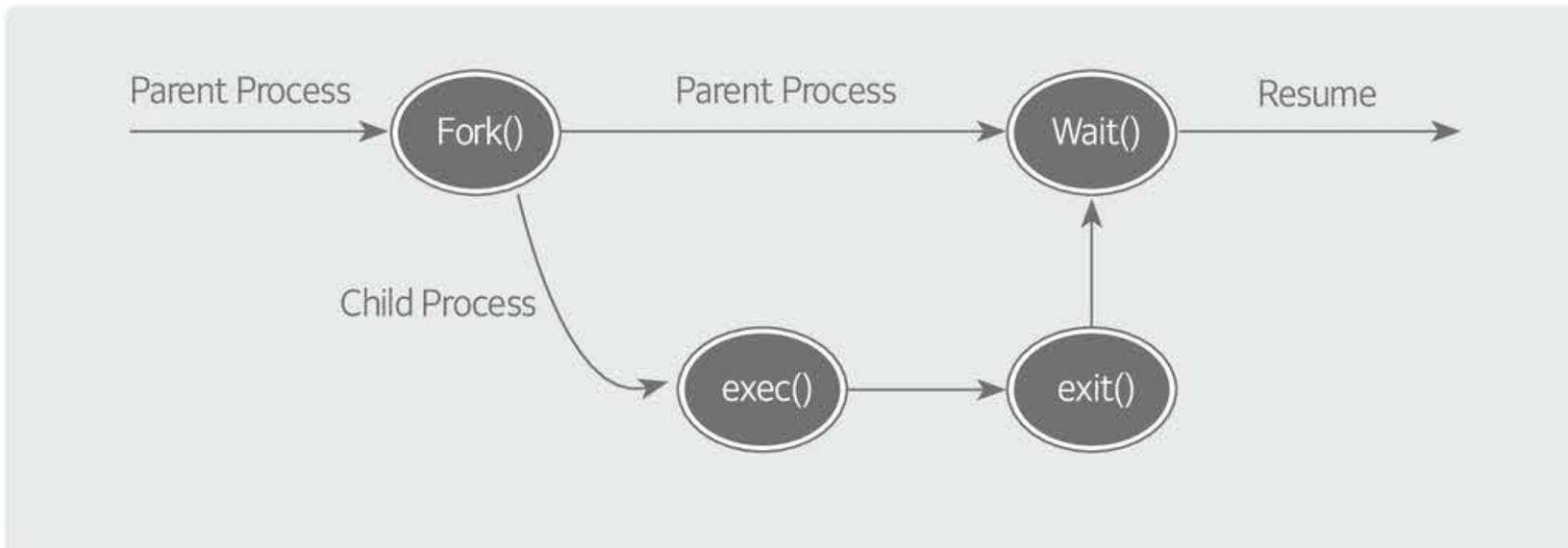
Processes must be able to run in parallel, and also be created and removed dynamically. A process can create other new processes through the "fork()" system call while it runs. The process that creates other processes is called a parent process, and a newly created process is called a child process. A child process can create another child process, and this relationship is formed in a tree structure, as shown in [Figure 17].



[Figure 17] Process Tree Structure in the UNIX OS (Example)

② Process termination

After completing the last code of the program, the process requests the OS to delete the process with the "exit()" system call, as shown in [Figure 18]. At that point, the child process returns data to the parent process with the "wait()" system call, and the OS recollects all the resources allocated to the deleted process.

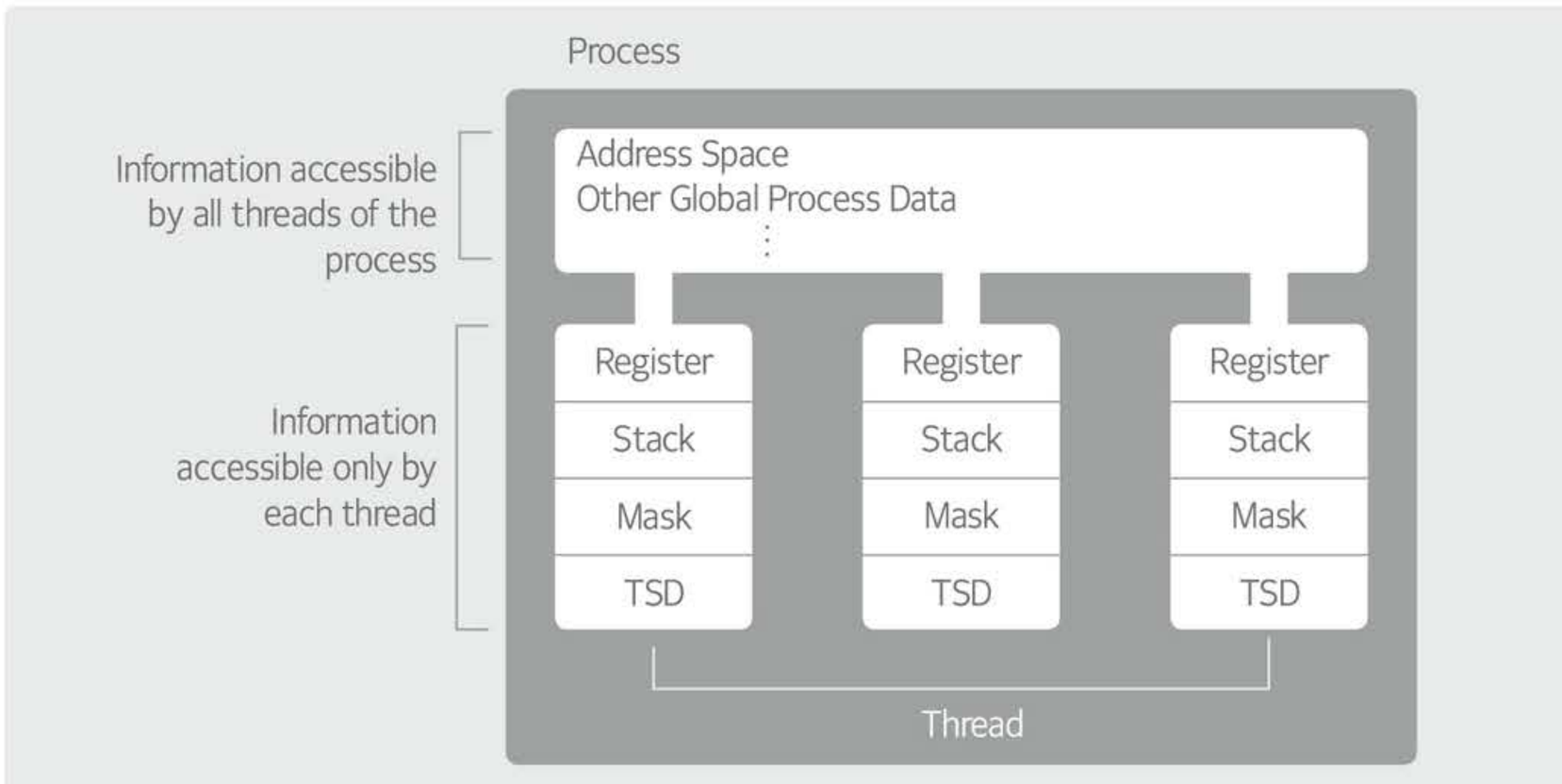


[Figure 18] Process Creation and Termination in the UNIX OS

C) Thread

① Concept of thread

A thread is a basic unit for using a CPU and is all called a lightweight process. As shown in [Figure 19], a process generally has one or more threads. A thread shares the memory unit, such as codes, data, and files, and creates its own register and stack. While processes use more CPU time than threads for context exchange, threads can exchange context more economically because threads share the memory unit.

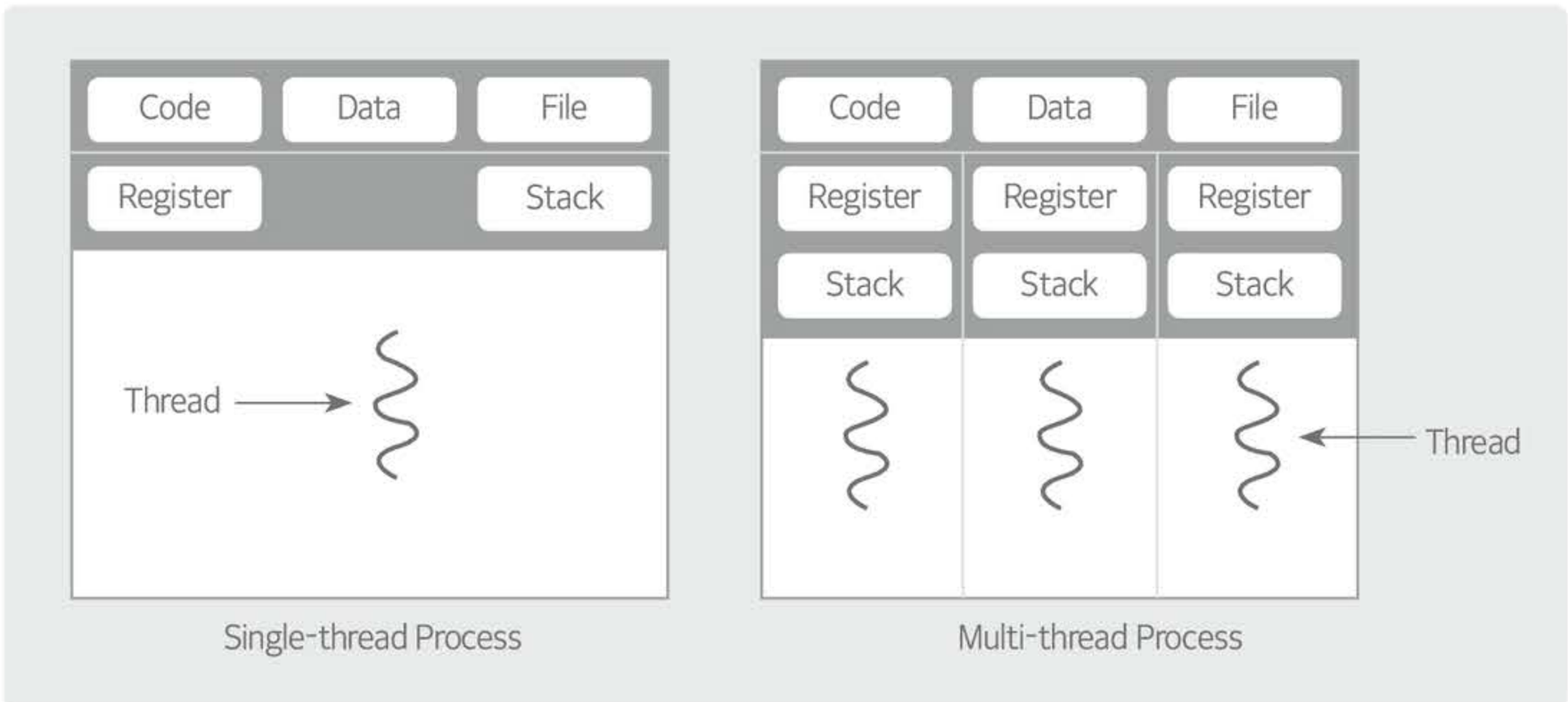


[Figure 19] Relationship between Thread and Process

② Multi-thread

As shown in [Figure 20], a single-thread process is a case in which a thread exists for a process, and a process, having multiple threads, is called a multi-threaded process. A thread operates with the status of "ready", "blocked", "running", and "terminated". A CPU can occupy only one thread at a time. In other words, only one thread runs at a time, and the status of the other threads is "blocked." When a thread is blocked, other threads can run. There are certain benefits to using multi-threads.

- The memory occupied by a process can be shared, and another thread can access the same memory address in a multi-thread system.
- The cost of thread creation and context exchange is more economical than the process.
- Threads increase the parallel attributes of the process and enable the utilization of multiprocessors.



[Figure 20] Single Thread and Multi-thread

03 Process Synchronization and Deadlock

A) Concept of process synchronization

A race condition refers to two or more parallel processes simultaneously accessing and changing the same data, or the order of manipulating the data affecting the execution result. In a race condition, only one process must be protected to manipulate data, and process synchronization is necessary for its protection.

B) Critical section problem

For process synchronization, each process defines a critical section in a part of the code and performs the task of changing the data that is shared with other critical section processes. Two or more processes that are mutually exclusive cannot run in the critical section at the same time. [Figure 21] shows the structure of the critical and remainder sections.

```
do {
entry section
critical section    // Critical section
exit section
remainder section // Remainder section
} while (TRUE);
```

[Figure 21] Critical Section Structure in a Process Code

The critical area problem is to establish rules so that the processes cooperate with each other. Each process must request permission to enter, even if it enters its critical section. The part of the code that implements such a request is called the entry section. The exit section is after the critical section, and the rest of the code is collectively called the remainder section.

Solving the critical section problem requires satisfying the following conditions:

- Mutual exclusion: If a process runs in its critical section, the other processes cannot run their own critical section.
- Progress: Only processes that are not running in the remainder section can enter a critical section without a running process.
- Bounded waiting: After a process makes a request to enter its critical section, the time allowed for other processes to enter the critical section should be limited until the process enters the critical section.

C) Solving the critical section problem

One of the ways to solve the critical section problem is to configure the synchronization hardware, not to allow the interruption of generation while the shared data in the critical section is being changed. However, the hardware method is not feasible, since it deteriorates the system efficiency in a multiprocessor environment. Another way is to use a semaphore, which is a synchronization tool. The semaphore,'S', is an integer variable, and only the P operation (wait) and V operation (signal) are enabled. If one process modifies the semaphore value, the other processes cannot modify the same semaphore value at the same time. It works by setting the semaphore value to "wait" when a process enters the critical section and changing the value to "signal" when the process exits the critical section. Another process checks the semaphore value and does not enter the critical section if the value is in the "wait" state, preventing multiple processes from simultaneously accessing the shared data.

D) Deadlock status

In a multi-programming environment, the processes compete with each other to use limited resources. When a process requests a resource, it obtains a "standby" status if the resource is not available. There may be a case when the status of a process in "standby" is never changed because the requested resource is allocated to other standby processes. This state is called the deadlock.

The deadlocked processes cannot finish the running program, and the system cannot start other tasks because the resource is tied to other processes. <Table 5> shows the conditions that can cause deadlock. They include mutual exclusion, hold & wait, non-preemption, and circular wait. The deadlock occurs when these four conditions are met.

〈Table 5〉 Conditions Causing Deadlock

Condition	Description
Mutual exclusion	• The state in which only one process can use the shared resource.
Hold & wait	• The state in which processes are requesting other resources while occupying the current resource.
Non-preemption	• The state in which the resource allocated to each process cannot be forcibly released until its use is complete.
Circular wait	• The state in which the requests for a resource by different processes are continuously repeated.

<Table 6> shows how to solve the deadlock state.

〈Table 6〉 Ways to Solve Deadlock State

Condition	Description
Prevention	• It removes the deadlock causing conditions in advance.
Avoidance	• It does not remove the condition causing the deadlock, but avoids them.
Detection	• It allows the deadlock condition to occur and finds its cause to resolve it.
Recovery	• It restarts the deadlocked process, or restores it into the original state in order to resolve the deadlock state.

04 Memory Unit Management

A) Understanding of memory unit management

A CPU uses scheduling to improve the utilization rate as if multiple programs share it at the same time. A CPU can only execute the instruction and read data when they are stored in the memory. Therefore, multiple processes must be stored in a memory unit for them to run. Virtual memory is the technique of running processes, even if they are not present in the memory unit. This technique allows a user program that is larger than the physical memory unit to run. This section describes various techniques for managing the storage unit.

B) Memory unit management technique

① Memory unit allocation technique

It is the technique of searching for the best way to allocate processes to the available main memory area when a process requests the memory unit for a new use. As shown in <Table 7>, the first-fit, best-fit, and worst-fit are the generally used techniques for selecting the available areas from a set of available areas.

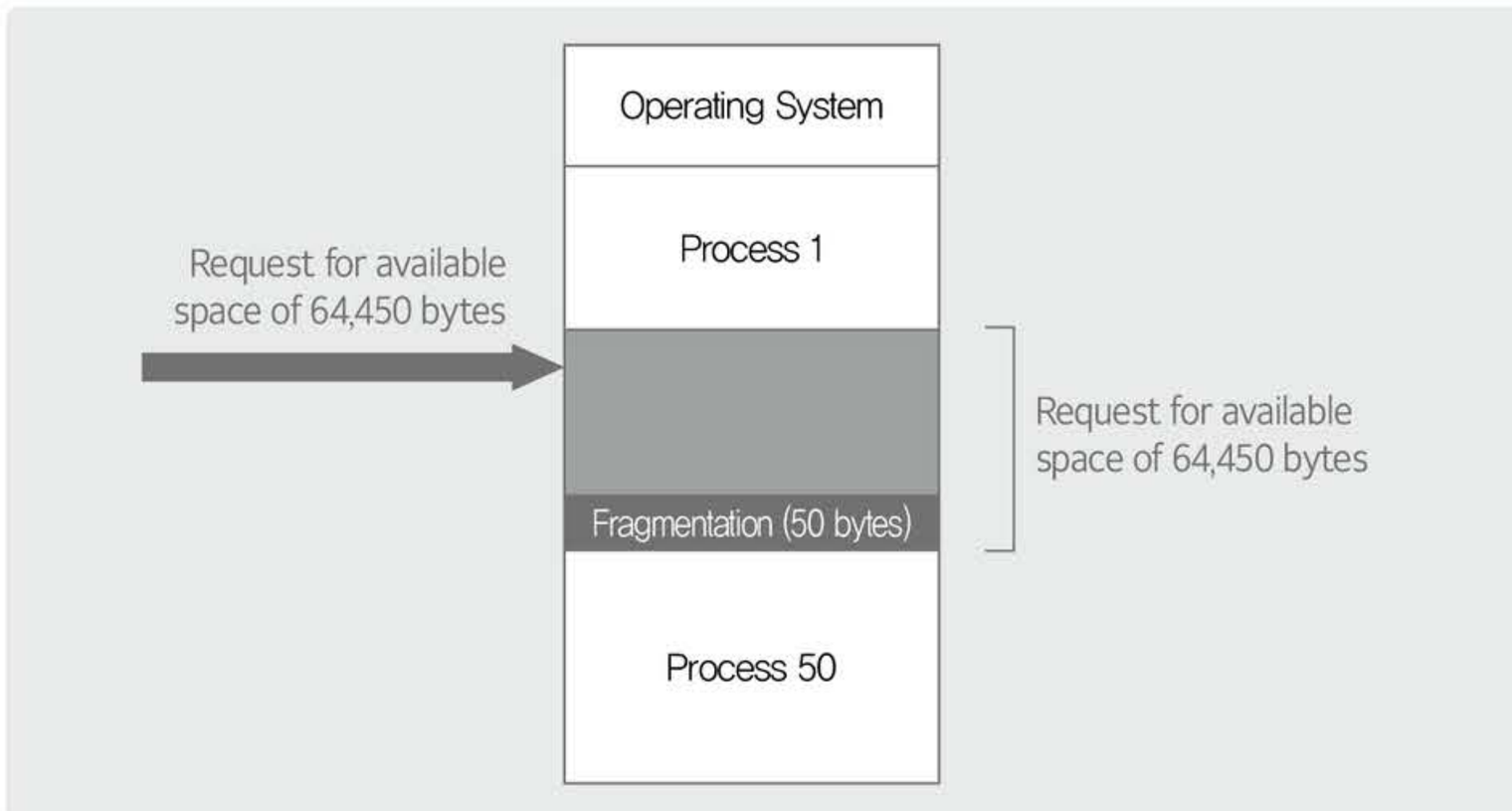
〈Table 7〉 Techniques of Allocating Memory Unit

Type	Description
First-fit	• It is a technique of comparing the request for the capacity of the memory unit and allocating the first searched space, which should be sufficiently large. Using this technique can make the allocation decision quick, but it increases fragmentation.
Best-fit	• This technique allocates the smallest available space that can meet the required memory unit capacity. This technique creates another smallest available space.
Worst-fit	• This technique allocates the largest available space. Since it allocates the largest available space, it can use the available space better than the best-fit technique, which can cause fragmentation by creating the smallest available space.

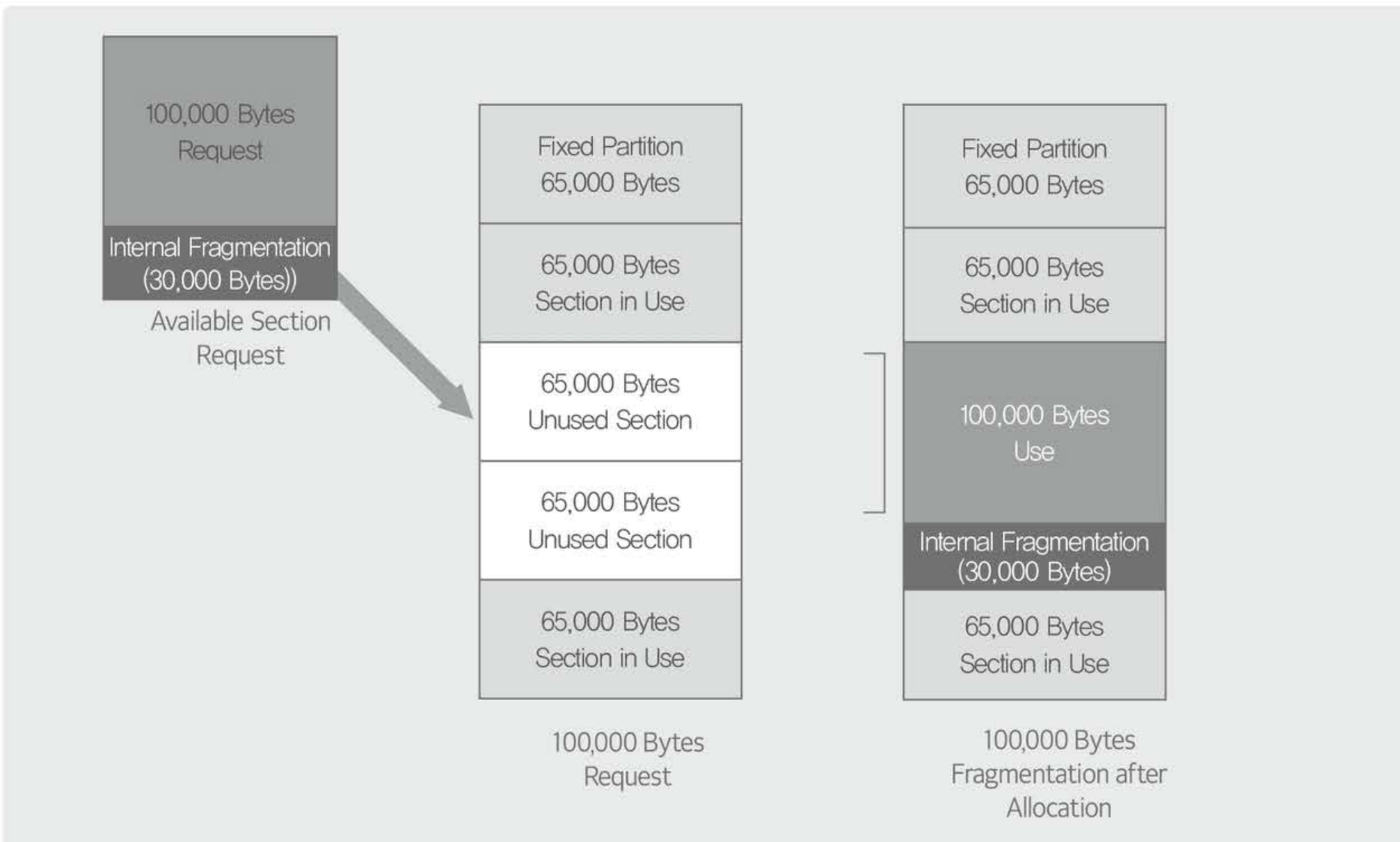
② Fragmentation problem

Memory space is allocated when the memory is requested and recollected after use. Repetition of the allocation and

recollection can result in memory space that is wasted without being used. Using the first-fit, best-fit, and worst-fit techniques to allocate the memory unit can cause external fragmentation, as shown in [Figure 22]. Let us assume that a process requests 64,450 bytes when there are 64,500 bytes of available space. In that case, 50 bytes of available space remains after allocating the requested space. If the memory is partitioned into a fixed size to provide multiple fixed-sized available spaces to processes, it may be slightly larger than the required space. The remaining space is called the internal fragmentation, as shown in [Figure 23].



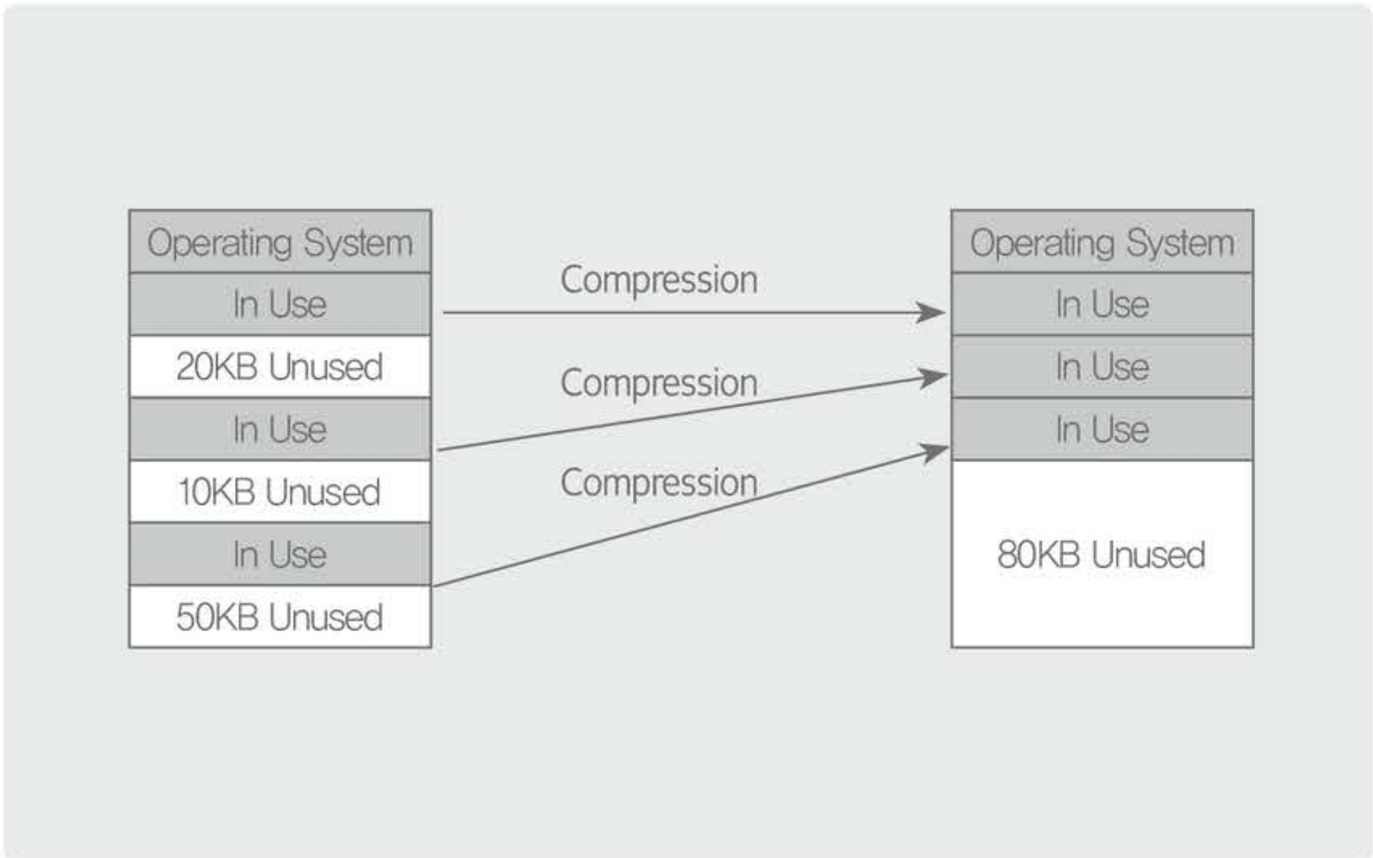
[Figure 22] Case of External Fragmentation



[Figure 23] Case of Internal Fragmentation

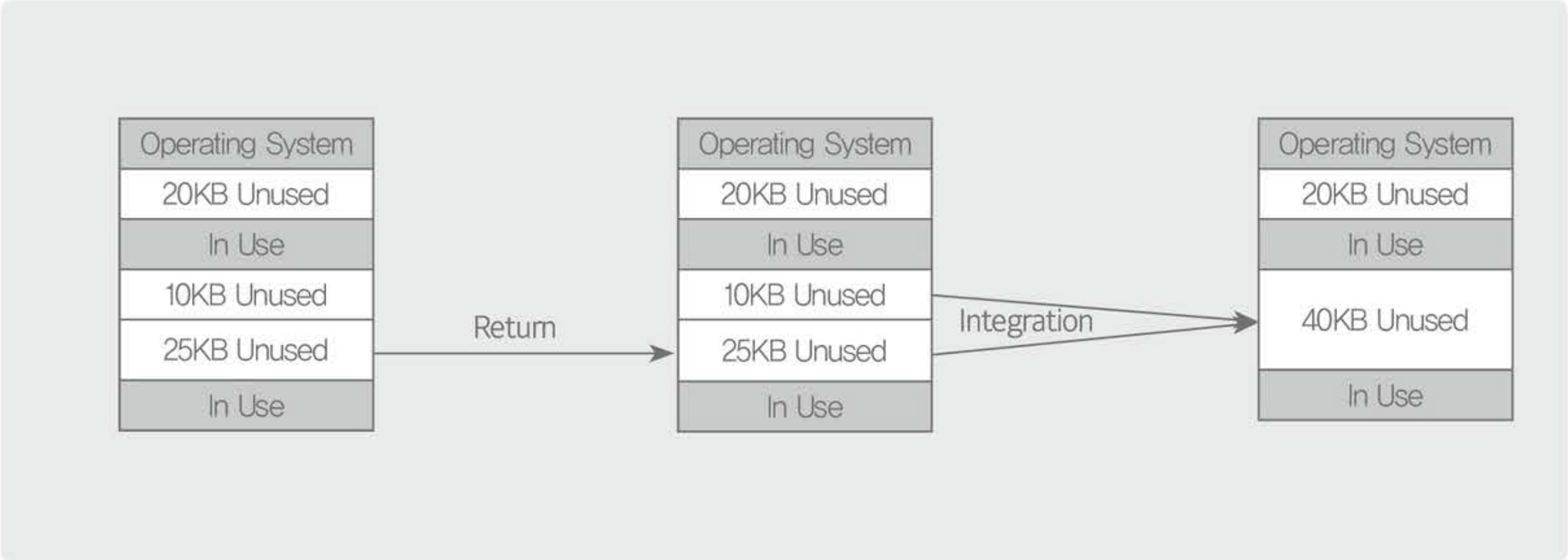
③ Solving the fragmentation problem

- Compaction technique:
  The compaction technique merges small-sized available memory. It is used to remove the external fragmentation that generated in a variable-size memory partition system, as shown in [Figure 24]. The technique is not widely used because it is complex to implement, and it causes overload as it requires a long time.



[Figure 24] Solving Fragmentation Problem through Compaction Technique

- Coalescing technique:
  The coalescing technique merges spaces with adjacent addresses in the unused empty space list in order to create a larger space and to prevent multiple small spaces, as shown in [Figure 25].



[Figure 25] Solving Fragmentation Problem through Coalescing Technique

05 Scheduling

OS supporting multiple programming uses scheduling to allocate processes to the CPU efficiently.

A) Purpose of scheduling

- Fairness to processes
- Maximizing throughput per unit time
- Minimizing response time
- Predictable execution time
- Preventing system overload
- Balanced resource utilization
- Preventing indefinite process execution
- Implementing priority schemes

B) Type and characteristics of scheduling

<Table 8> shows the types and characteristics of scheduling algorithms.

- Preemptive scheduling: A process can take CPU resources while another process is occupying them.
- Non-preemptive scheduling: If a CPU resource is allocated to a process, it cannot be allocated to another process until the task has been completed.

〈Type 8〉 Types and Characteristics of Scheduling Algorithms

Type	Method	Characteristics	Class
FIFO (First In First Out)	- The simplest scheduling technique to allocate CPU resources according to the order in which they are requested - A method of allocating CPUs in the order in which they entered the rows (queue matrix) of resource requesting processes	- Not suitable for interactive - Simple and fair - Prediction of response speed	Non preemptive
Priority	- A method of assigning the priority to each process and allocating the CPU resources to the highest priority	- Fixed priority - Variable priority - Purchased priority	Non preemptive
SJF (Shortest Job First)	- A method of allocating CPU resources to the process with the shortest expected job time	- Advantageous to jobs with the short execution time and a long time expected for large jobs	Non preemptive
SRT(Shortest Remaining Time)	- A method of allocating CPU resources to the process with the shortest expected remaining job time - Can be considered as preemptive SJF	- Job processing is the same as SJF, but it theoretically requires the least waiting time.	Non preemptive
R-R(Round Robin)	Like FIFO, it allocates CPU resources according to the order in which they are requested, but each process can use the CPU only for the specified period. It can be considered as a preemptive FIFO process. The most suitable method for time series segmentation (TSS)	- Effective for TSS - Same as FIFO if the specified allocation time is large - Context exchange frequently occurs if the allocated time is short.	Preemptive

Deadline	- A method of ensuring that each process is completed within the given time limit	- Significant overhead and complexity since the deadline must be calculated each time	Non preemptive
HRN(Hightest Response-ratio Next)	- Complementing the problem of large jobs taking too much time in SJF	- Priority = (Waiting time + Execution time)/Execution time	Non preemptive
MLQ(Multi-Level Queue)	- Processing different jobs by time slice in each queue	- Each queue using the unique scheduling algorithm	Preemptive
MFO(Multi-Level Feedback Queue)	- Processing by multiple feedback queues through a readiness queue	- Increased CPU and I/O device efficiencies	Preemptive

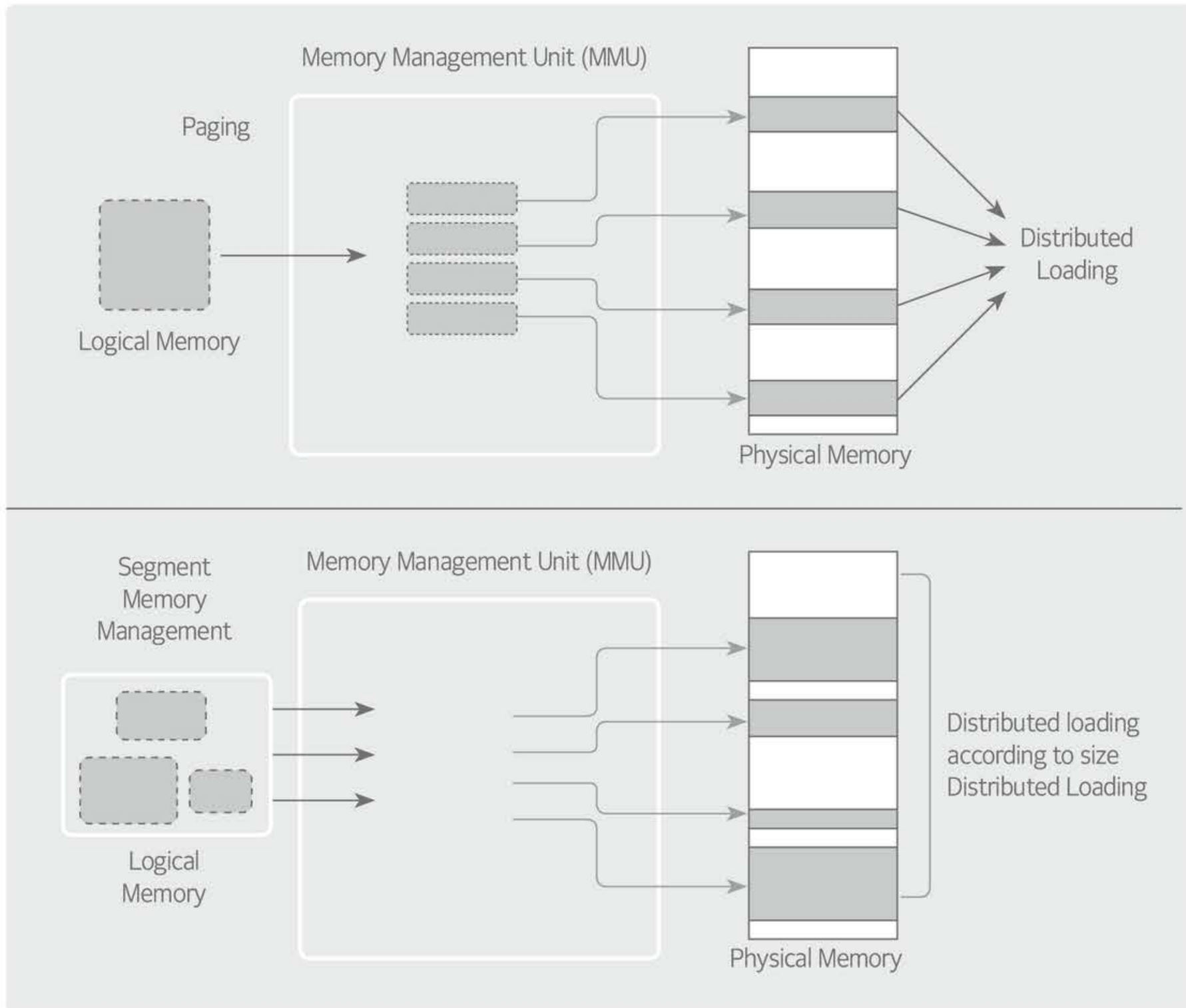
06 Virtual Memory Unit

The technique to solve the OS's limited memory space problem uses the main memory with a small capacity, as a large capacity with a large amount of free space. The virtual memory unit is not a physical device, but it logically uses a large-capacity auxiliary memory device as the main memory device. Therefore, it is necessary to map the address of the virtual memory unit to the address of the main memory unit in order to execute a program stored in the virtual memory unit.

A) Implementing a virtual memory unit

A virtual memory unit includes the virtual address that processes refer to and the physical address that points to the available main memory. Whenever a process accesses a virtual address, the system must quickly convert it into the physical address through the memory management unit (MMU).It is divided into a paging technique and segmentation technique, according to the block configuration. Some systems may use a combination of both techniques (see [Figure 26]).

- Paging technique: This technique partitions the main memory unit into fixed sizes, called frames, and also partitions the task to the process stored in the virtual memory unit into fixed sizes, called pages, and loads them into the frame to run the task.
- Segmentation technique: This technique partitions the task to the process stored in the virtual memory unit into size segments, which are logical units of various sizes, and then loads them into the main memory unit to run.



[Figure 26] Comparison of Paging and Segmentation Memory Allocations

B) Page replacement technique of the virtual memory unit

If all pages are filled in the paging virtual memory unit, free space must be created before loading new memory pages from the auxiliary memory unit. The system selects a page to replace the page replacement technique, affecting the system efficiency and performance.

- Optimal technique: This technique replaces a page that is not likely to be used for the longest time. Although it is the optimal replacement technique with the minimum absent page rate, it is not a realistic technique, since it is difficult to predict behavior.
- First In First Out (FIFO) technique: This technique tracks the order of loading in the main memory and replaces the first loaded page.
- Least Recently Used (LRU) technique: This technique replaces the page that has been unused for the longest time.
- Least Frequently Used (LFU) technique: This technique tracks each page's utilization frequency and replaces the least used or least intensively used page.
- Not Used Recently (NUR) technique: This technique uses the tendency that the page not used recently is less likely to be used in the near future, so it replaces the page.

C) Factors that affect the virtual memory unit performance

- Locality: A tendency to intensively refer to only some pages while a process is running. It is divided into temporal locality and spatial locality.
- Working Set: A set of frequently referred to page lists for a certain period to execute a process efficiently. It reduces page absence and page replacement by placing the frequently referred to working sets in the main memory.
- Thrashing: A phenomenon in which the CPU utilization rate decreases because page replacement takes longer than the processing time. The degree of multi-programming is reduced, the CPU utilization rate is increased, or a working set is used to prevent thrashing.

07 File System

A) Understanding of file system

A file system stores related data to find and access the OS data and programs quickly. It is a directory structure that provides information on an aggregation of files and all files in the system.

① Concept of file

It is a set of data with a name and stored in an auxiliary memory unit, such as a disk or tape. The OS maps files to physical devices. In other words, when the OS writes a program or data to a file, it is permanently stored in a nonvolatile physical device.

② File attributes

Files have several properties to enable the ability to identify and manage them. The most common is the file name, which is expressed as a single string. A process or user can independently create and access files using the file name attribute. The types of file attributes are described as follows:

- Name: Should be maintained in a legible form by users.
- Location: Points to the device and location of the file and includes the directory path.
- Size: Includes the current file size (in bytes, words, or blocks) and the maximum allowable size.
- Protection: Controls who can read, write, or run the file.
- Time, date, and user identification: Include the data related to the creation time, recent information, and recent usage.

B) Concept of directory

A directory is a logical structure to manage tens of thousands of files managed by a file system, and can be the total size of gigabytes or terabytes. It records the information about all files. The following actions are performed on a directory:

- File search: A user can search a file from the directory structure.
- File creation: A new file is created and added to the directory.
- File delete: Unneeded files are removed from the file.
- Directory list: Lists files in a directory and shows the directory item.

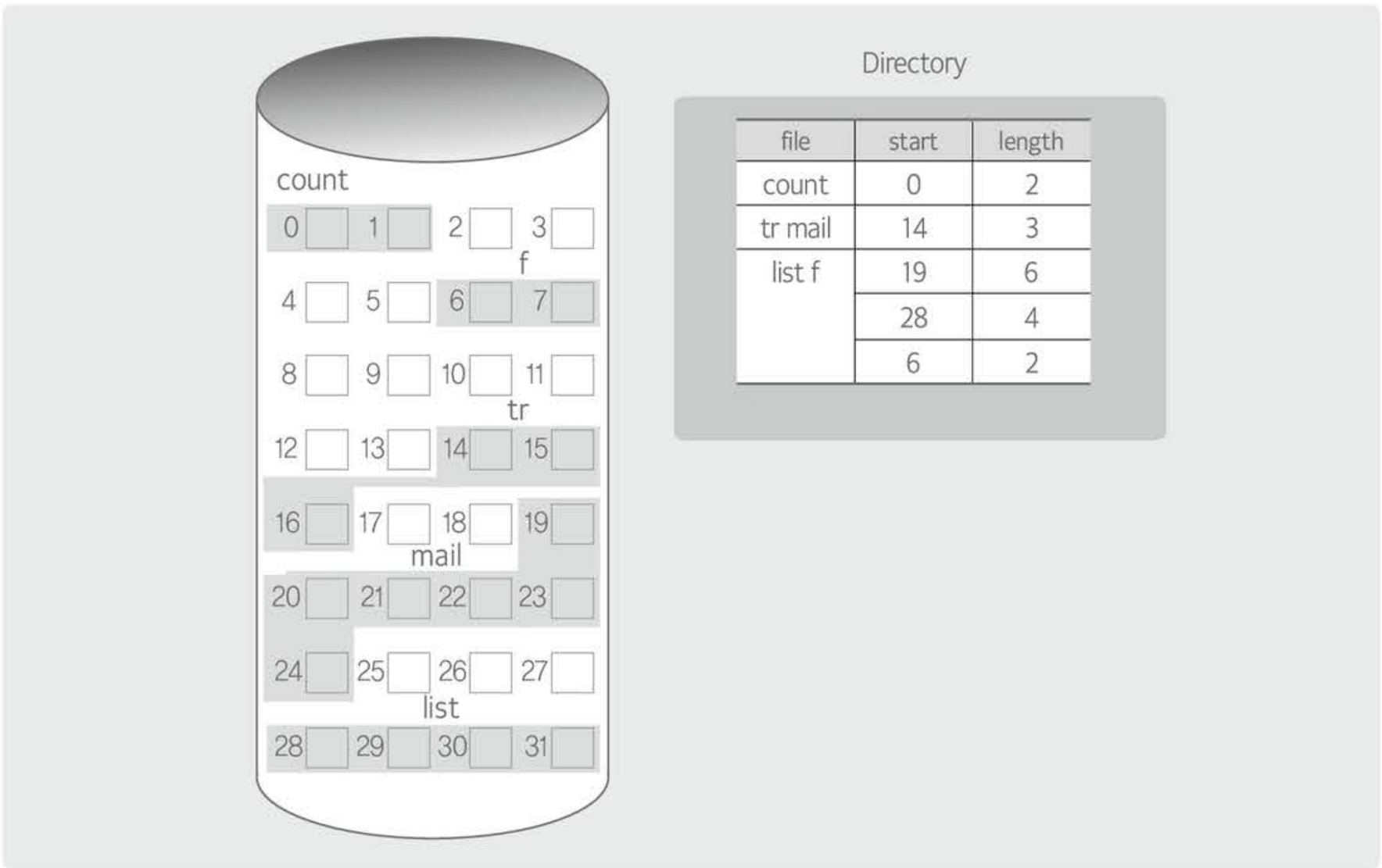
• Renaming file: A file name must be changeable, and a renamed file should exist at a point within the directory structure.

C) Allocation by a file system

A file system should consider which space on the disk to use and quickly access the files. An OS's file system generally uses the contiguous allocation, linked allocation, or indexed allocation.

• Contiguous allocation:

This method arranges files as a set of contiguous addresses on the physical disk, as shown in [Figure 27] below. The disk addresses of stored files are defined linearly. This method has the advantage of a relatively high reading speed, since the files are stored in contiguous addresses. However, it creates an external fragmentation problem.

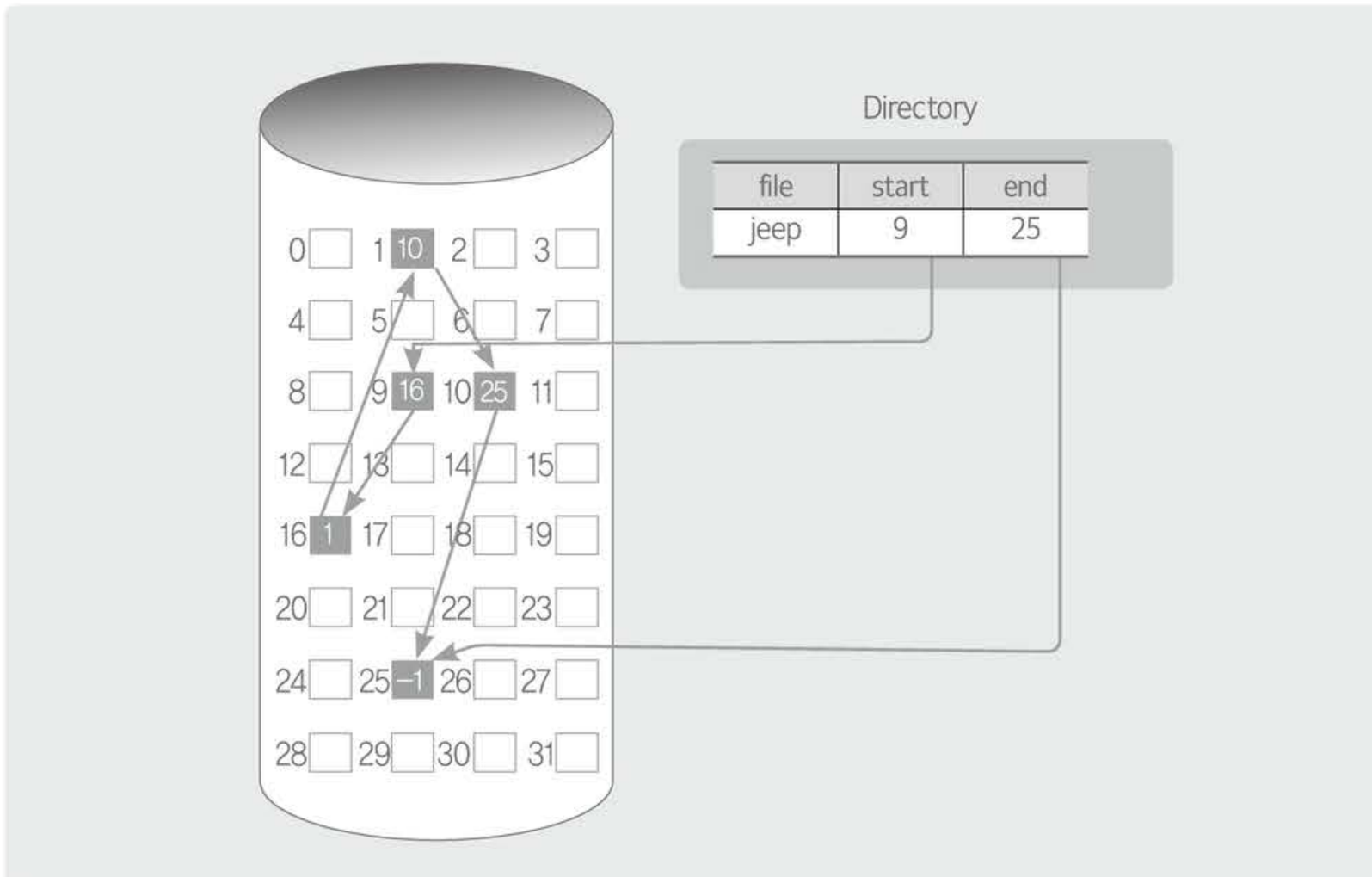


file	start	length
count	0	2
tr mail	14	3
list f	19	6
	28	4
	6	2

[Figure 27] Contiguous Allocation

• Linked allocation

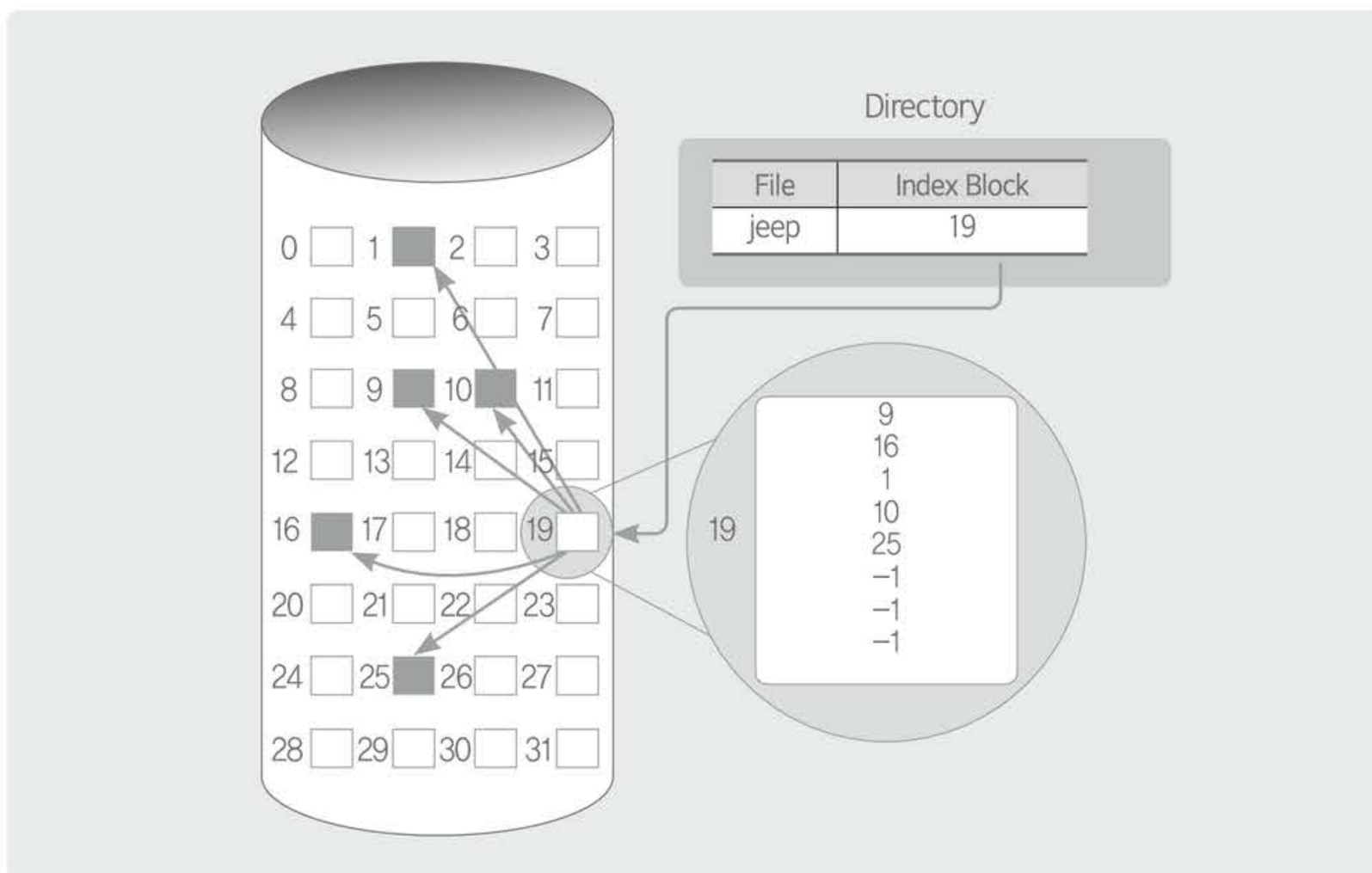
This method stores files in block units and links each block to a list, as shown in [Figure 28] below. The directory keeps pointers to the first and last blocks of the files. As shown in the figure, the directory contains 9 and 25, which are the first and last block pointers, respectively. Block 9 is linked to blocks 16, 1, 9, 10, and 25 in that order. This method can solve the external fragmentation problem of the contiguous allocation method. However, the linked allocation method is slow to read and requires the area to store the pointers separately, since data blocks are scattered on the physical disk. Moreover, it can cause data errors if the linked pointer data is lost or corrupt.



[Figure 28] Linked Allocation

• Indexed allocation

The linked allocation cannot support direct access and is difficult to manage the loss, since the block pointer addresses are scattered throughout the disk. The indexed allocation solves the problem by managing all pointer data of a file in an index block, as shown in [Figure 29]. The indexed allocation does not create external fragmentation and supports the direct addresses. However, the waste of storage space is larger than the linked allocation method.



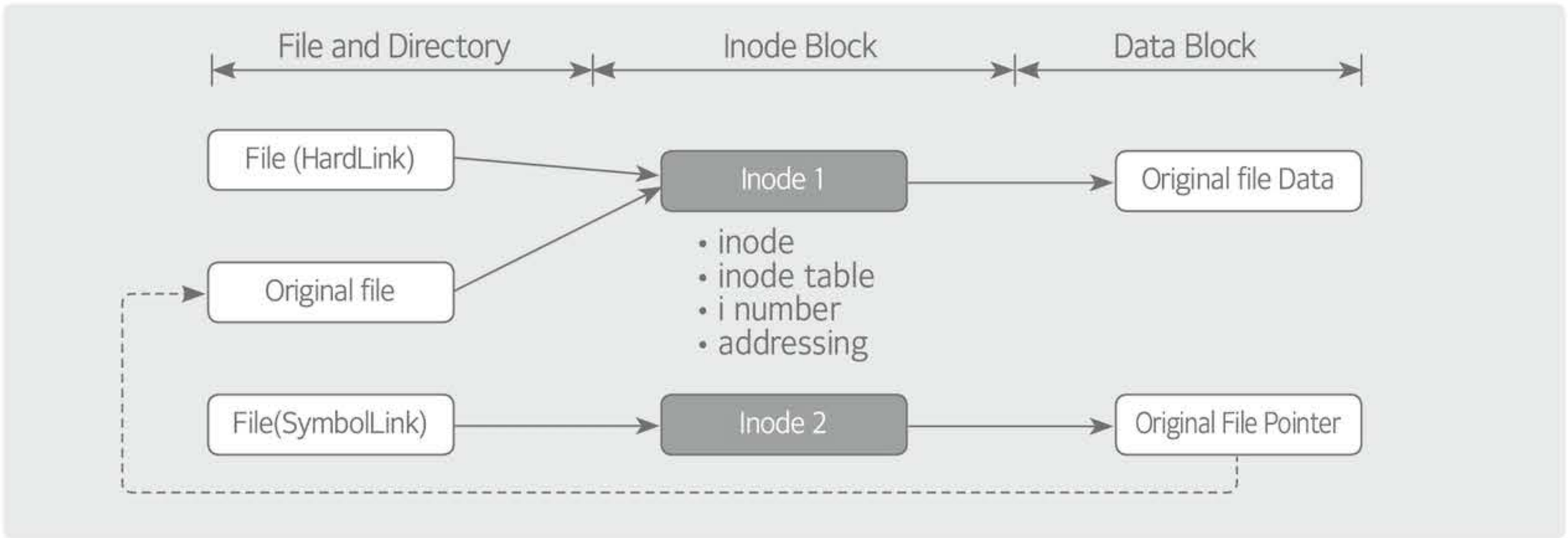
[Figure 29] Indexed Allocation

D) Types of files system for each OS

OS	File System
UNIX	Boot block, super block, bitmap block, i-node, and data block
Linux	File extension system (ext, ext2, ext3, and ext4), ZFS, ResierFS, and XFS
Solaris	ZFS and UFS
Mac OS	HFS and HFS+
Windows	FAT, NTFS

E) UNIXi-node

In a UNIX system, i-node has the role of allocation, application, creation, link, and deleting through the file/directory data.



i-node Component	Description
inode	- Contains all the data of a file or directory - Owner information, access information, file information, link, and typ
inode table	A file system table that has all the inodes of files or directories
i number	An entry number of an inode registered in the i-list
addressing	Managing the block location with 13 fields • Ten direct data blocks (0~9): 96kb data • One single indirect data block (10): 16MB • One double indirect data block (11): 32GB • One triple indirect data block (12): 70TB

08 Input/ Output System

The input/output system includes an input/output device and the system's input/output module (controller). A physical input/output device handles the inputs and outputs of data and between the actual processor and the computer user, and the input/output module provides the functions of input/output device control, timing

coordination, communication with the processor, communication with input/output devices, data buffering, and error detection, so the process can easily control multiple input/output devices.

IV. Computer Architecture

Recent trends and issues

In computer engineering, computer architecture is a conceptual design and the operational structure that is the basis for computer systems. It is a blueprint that functionally describes the design portability and requirements (especially speed and interconnection) of various parts of a computer. It mainly focuses on how the central processing unit (CPU) performs and accesses memory addresses internally.
Modern computer structures are based on the Von Neumann architecture, and the basic instruction processing structure has not changed much. Though parallel processing and other technologies have improved in their performance, only recently have new concepts, such as neuromorphic chips and quantum computers, been introduced.

Learning objectives

1. To be able to explain the computer structure and the operational principle.
2. To be able to explain the memory hierarchy and operational principle.
3. To be able to explain the latest computer structure technologies and trends.

Keywords

Von Neumann architecture, Harvard architecture, CPU, Memory unit, Input/Output device, CISS, RISC, Neuromorphic chip, quantum computer

✚ Preview for practical business

John Doe, who works for company A, is a software engineer with four years of development experience. Java is the programming language he generally uses, but he was recently given a task that required him to program using C. Since the Java language managed memory in a runtime environment, called the Java Virtual Machine (JVM), it was not necessary to pay much attention to memory during programming. When developing with the C language, he had to learn the concept of memory, such as memory allocation and return. Most importantly, he had to learn the memory address system through pointers. He had to be familiar with computer structure to be able to develop software in C effectively. This chapter describes the computer structure in detail.

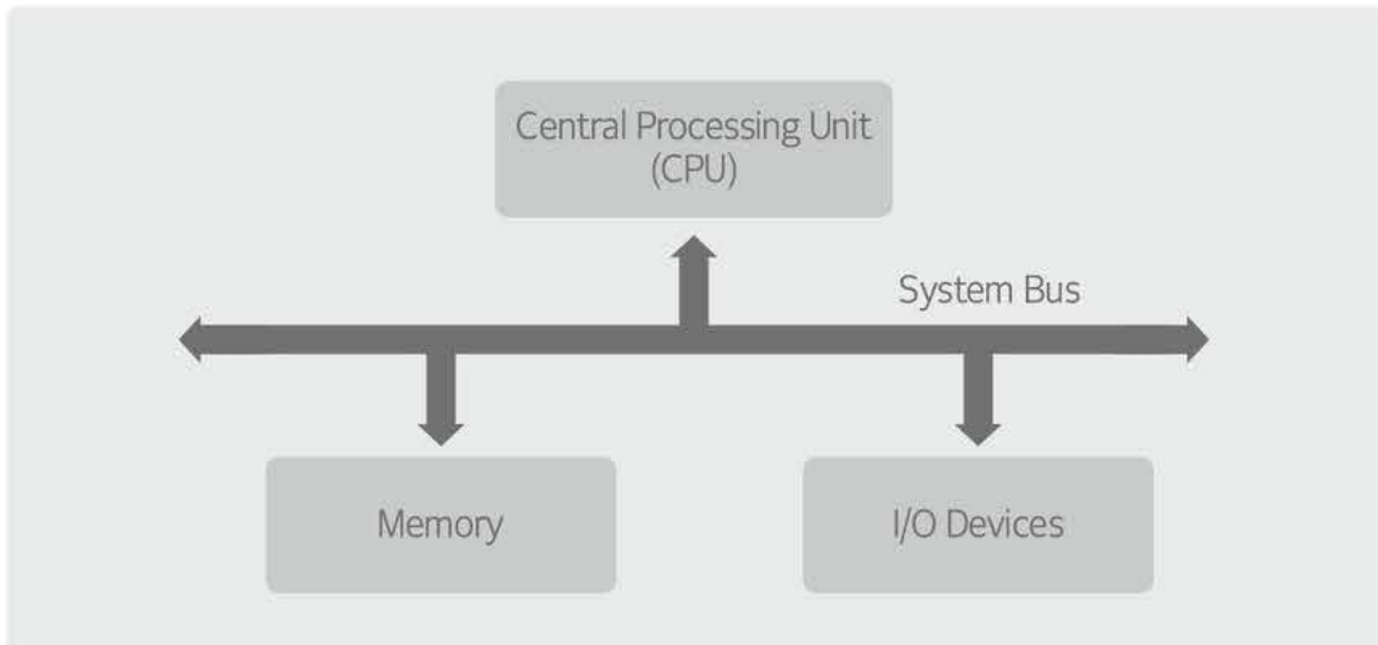
01 Computer Structure and Architecture

A) Basic computer structure

The basic function of a computer is to run a program code in the specified sequence. In other words, it reads, processes, and stores the needed data. [Figure 30] shows the main components of a computer.

- Central Processing Unit (CPU): Also, commonly known as a processor, it plays a key role of the program running and data processing.
- Memory:
  - Main memory: Closely located to the CPU, it consists of memory semiconductor chips. It can be accessed at a high speed and can be used only as temporary storage because it has no permanent storage capabilities.
  - Auxiliary storage device: The secondary storage device can be accessed at a low speed because it includes mechanical devices. It has a high storage density, and it is moderately priced. Disks and magnetic tapes are some examples.

- I/O device: consists of an input device and an output device to be used as the tool for interaction between the users and computers.

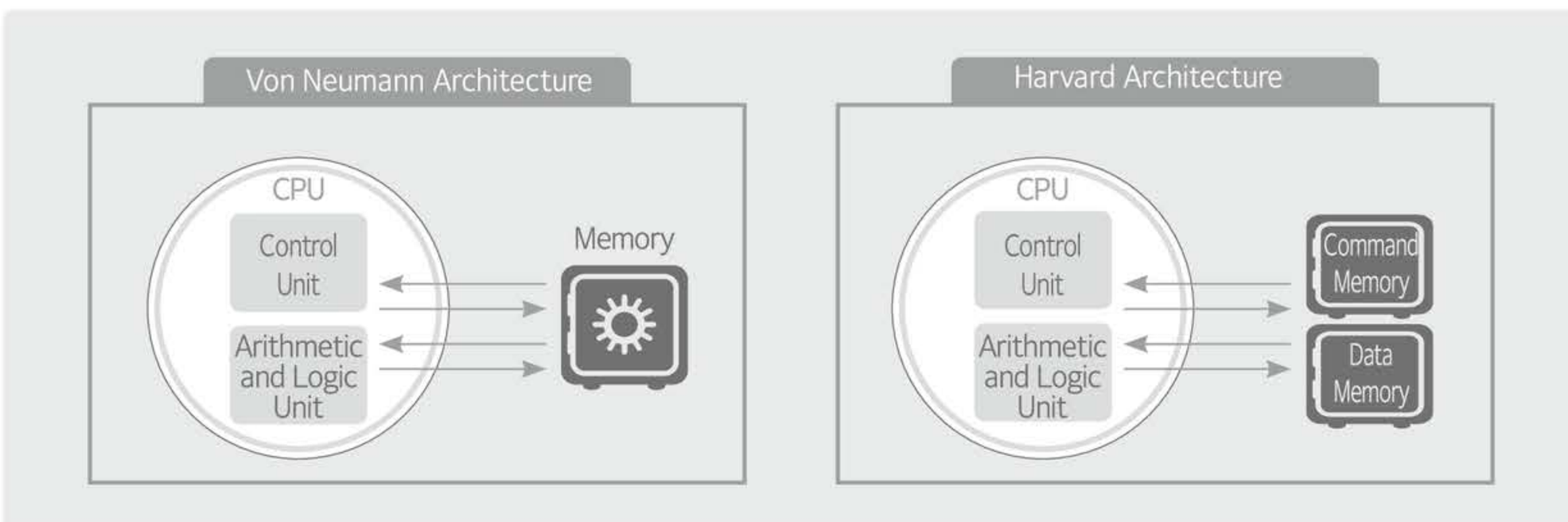


[Figure 30] Computer Structure

B) Types of computer architecture

The types of computer architecture include the von Neumann architecture, which is a structure that applies the stored-program design principle[1] , suggested by von Neumann in 1945, and the Harvard architecture,which separates instruction memory and data memory. As shown in [Figure 31], these two types have different memory structures, and each has advantages and disadvantages.

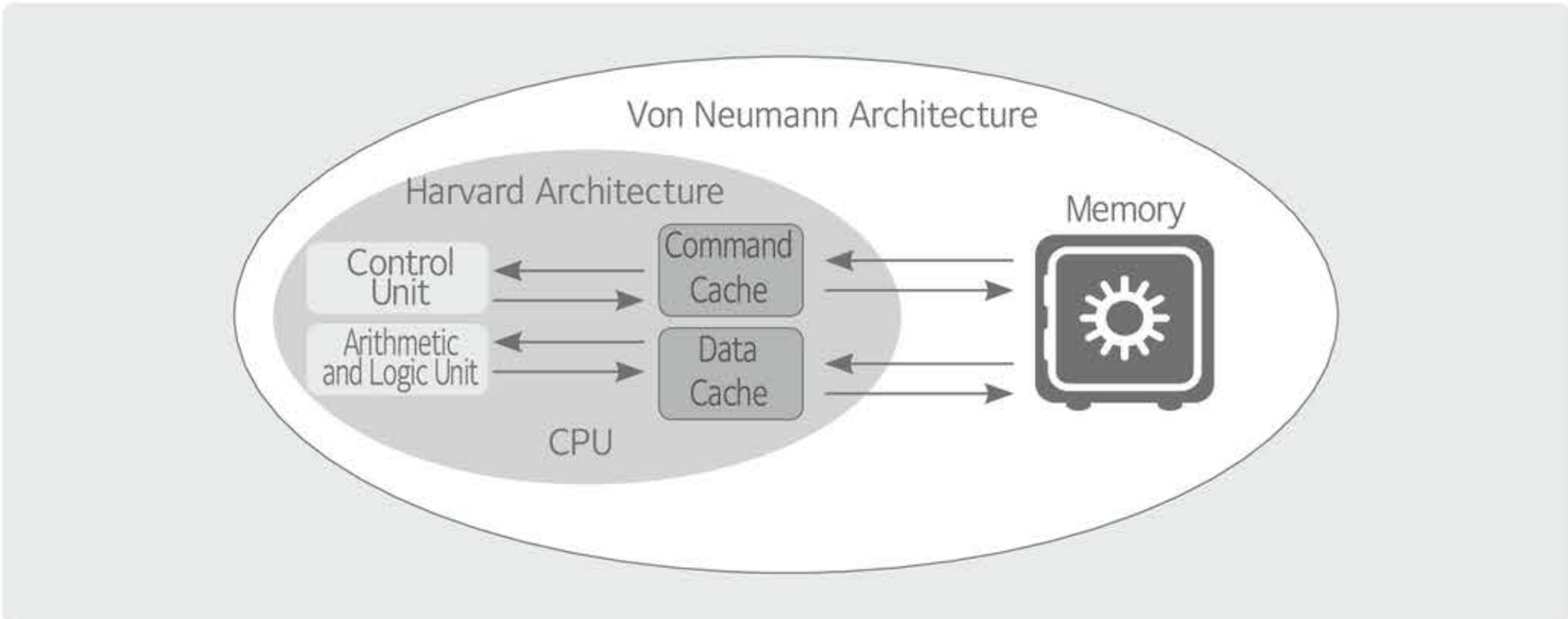
- Von Neumann architecture: The CPU can read commands from the memory and can read and write data both from the memory and to the memory. Instructions and data cannot be accessed simultaneously because they use the same signal bus and memory.
- Harvard architecture: It solves computer bottlenecks by storing commands and data in different memories and improves performance by reading and writing commands in parallel. However, the bus system design becomes complex.



[Figure 31] Von Neumann Architecture and Harvard Architecture

1 A memory unit is attached next to the CPU, and programs and data stored in the memory unit are sequentially fetched and processed according to the execution instructions.

Since the von Neumann and Harvard architectures each has advantages and disadvantages, the most recent high-performance CPUs apply both architectures in their design. In other words, they separate the cache memory for instruction and data, and they apply the Harvard architecture inside the CPU (CPU and cache) and the von Neumann architecture outside CPU (CPU and memory), as shown in [Figure 32].



[Figure 32] Computer Structure Applying Both Architectures

02 CPU

A) Definition of CPU

The CPU is the most important part of computers, as it interprets instructions and handles arithmetic or logical operations and data processing. It plays the key role of running programs and processing data.

B) CPU execution

As shown in <Table 9>, the CPU operation is divided into a function that commonly runs for all instructions, and a function that runs only when necessary, according to the instructions.

〈Table 9〉 CPU Execution

Type	Detailed Operation	Description
Commonly performed	• Instruction fetch	Reads instructions from the memory unit
	• Instruction decoding	Interprets the fetched instructions to determine the action to take
Performed when necessary	• Data fetch	Reads data from a memory unit of the I/O unit if the data is necessary to run the instruction
	• Data processing	Reads data from a memory unit of the I/O unit if the data is necessary to run the instruction
	• Data storage	Stores execution results

C) CPU components

As shown in [Figure 33], a CPU consists of a control unit, an arithmetic logic unit (ALU), registers, and buses that

connect them in order to deliver the data.

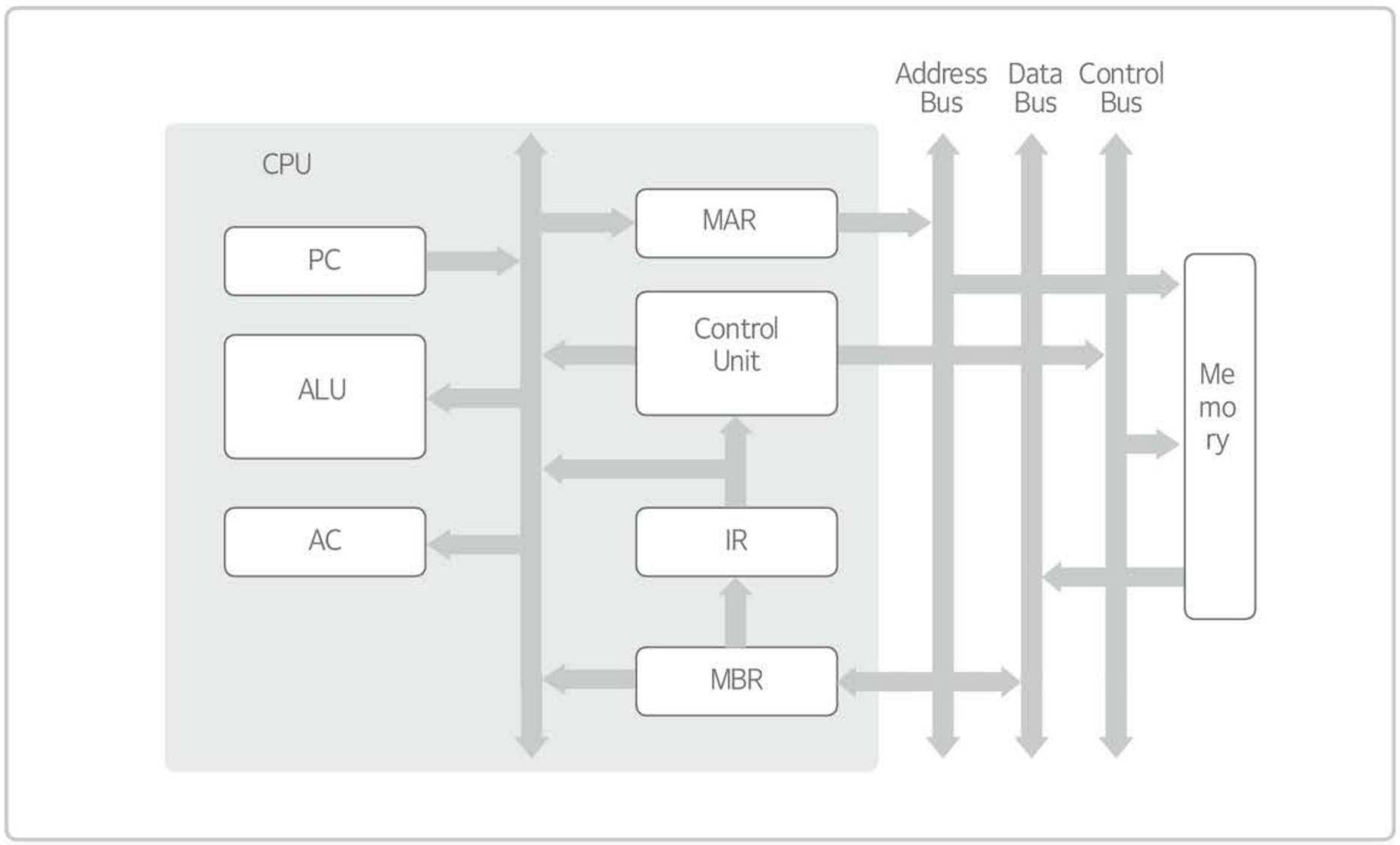
① Control unit: A hardware module that sequentially generates control signals to interpret the program codes (instructions) and run them.

② ALU: A key element of the CPU to execute arithmetic operations[2] and logical operations[3]

Register: A temporary storage area that temporarily stores instructions waiting to be processed by the CPU or the intermediate result values of the CPU operation. The register types include, PC (program counter), IR (instruction register), AC (accumulator), MAR (memory address register), MBR (memory buffer register), and SP (stack pointer).

③ Bus: A common transmission line that connects the CPU, memory, I/O unit, etc., in order to exchange necessary data. Buses are classified as follows:

- Address bus: A set of signal lines that transmits address data generated by the CPU.
- Data bus: A set of signal lines that transmits data from the CPU to a memory unit or an I/O unit.
- Control bus: A set of signal lines that are necessary for the CPU to control various system elements.



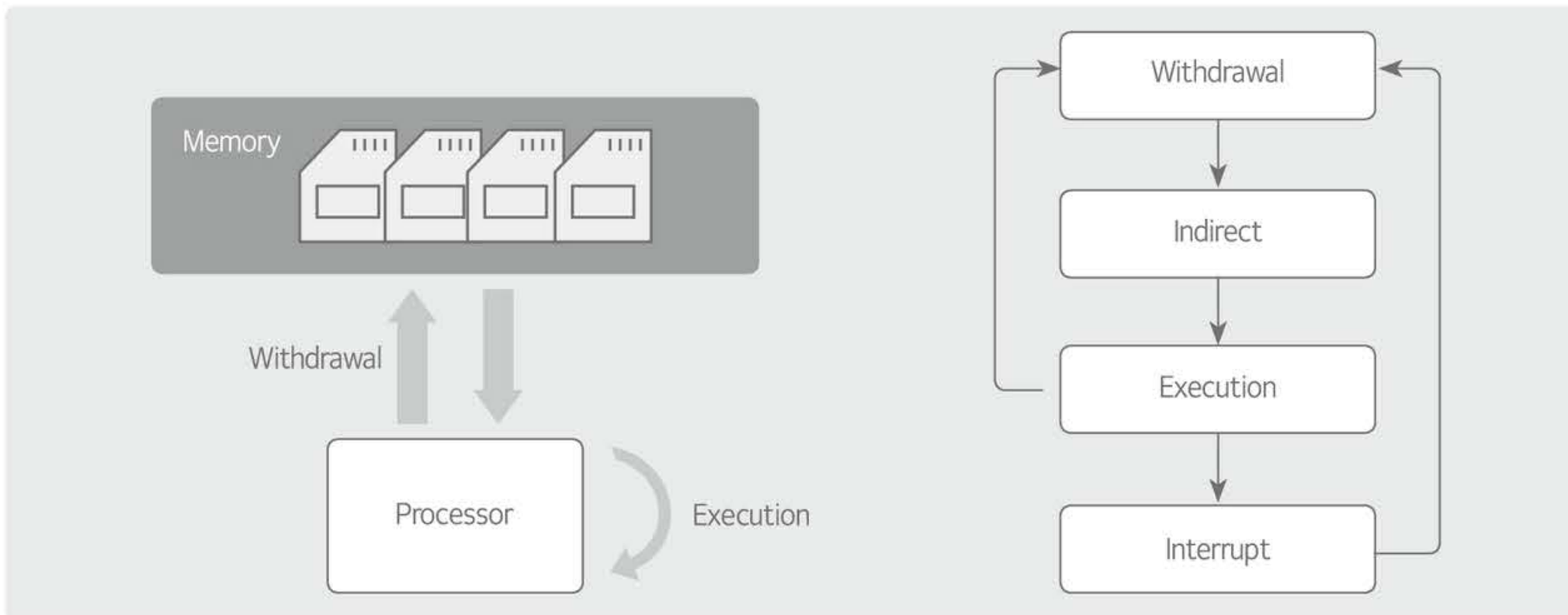
[Figure 33] CPU Structure

D) Instruction cycle

This is the entire process required for the CPU to execute an instruction. The CPU repeats it from the moment the CPU starts executing the program, until the power is turned off or an irrecoverable error occurs to terminate the execution. The instruction cycle from the fetching of the instruction to the completion of the operation consists of a fetch cycle, an indirect cycle, an execution cycle, and an interrupt cycle, as shown in [Figure 34].

2 Arithmetic operation: Addition, subtraction, multiplication, and division

3 Logic operation: AND, OR, NOT, XOR, etc.



[Figure 34] Instruction Cycle Conceptual Diagram

<Table 10> shows the entire instruction cycle. The CPU components needed in each cycle are different.

〈Table 10〉 Relationship between Instruction Cycle Types and CPU Components

Type	Detailed Operation	Related CPU Component
Fetch cycle	• The cycle fetches an instruction from the memory unit location pointed to by the PC.	- Register (PC, MAR, MBR, and IR) - CPU internal bus
Execution cycle	• The CPU decodes the fetched instruction code and performs the needed operation, according to the result.	- Register (AC) - ALU - Control unit
Interruption cycle	• The cycle inspects the interruption request signal, stores the current PC data in the stack, and loads the ISR's beginning address to the PC.	- Register (MBR, PC, MAR, SP)
Indirect cycle	• The cycle is the process of reading the actual data address from the memory unit before the execution cycle begins.	- Register (MAR, IR, MBR) - CPU internal bus

E) Instruction set structure, CISC and RISC

Instruction set, or instruction set architecture (ISA), refers to the machine language instructions that a microprocessor can recognize, understand its function, and execute. The ISA is part of the programming-related computer architecture, and it includes data types, instructions, registers, addressing modes, memory structures, interruptions, exception handling, and external input/output.

The leading ISAs are the complex instruction set computer (CISC) and the Reduced instruction set computer (RISC). <Table 4> shows the concepts and differences in these structures.

- CISC:A complex instruction-type computer that embeds many complex instructions into the hardware in order to process the complex instructions as a single instruction.
- RISC: A reduced instruction-type computer that embeds a few simple instructions into the hardware to process complex instructions as a set of simple instructions.

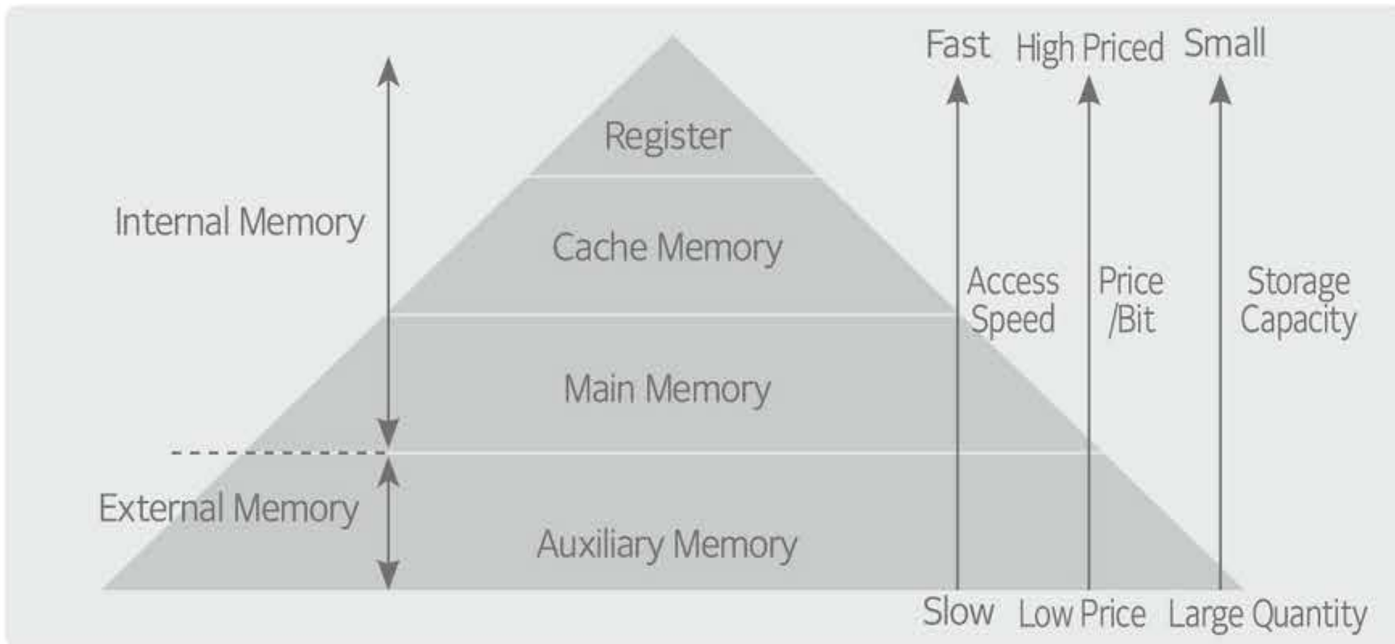
〈Table 11〉 Comparison of CISC and RISC

Type	CISC	RISC
System Configuration	Data Path, Micro Program Control Unit, Cache, Memory	Data Path, Hardwired Control Unit, Data Cache, Command Cache, Memory
Configuration of Number of Commands	1 Byte: OP-CODE 2 Byte: OP-CODE OP-CODE; OP-CODE OPERAND 3 Byte: OP-CODE OP-CODE OPERAND; OP-CODE OPERAND OPERAND 4 Byte: OP-CODE OP-CODE OPERAND OPERAND Since only the needed information is stored as instruction in a variable length, it can reduce wasted code, and the program size decreases accordingly.	OP-CODE OPERAND1 OPERAND2 Fixed to 32 Bits It is easier to apply the pipeline since the instruction size is fixed to 32 bits.
Control system	Micro-program type	Hard-wired type
Compiler	Simple structure	Complex structure
Number of registers	Few	Many
Example	Intel series (x86)	ARM and MIPS

03 Memory

A) Memory unit's hierarchical structure

As shown in [Figure 35], the memory unit at a higher level has a higher price per bit, less capacity, a shorter access time, and a higher access frequency by the CPU.



[Figure 35] Memory Unit's Hierarchical Structure

B) Factors for performance evaluation of memory unit

- Capacity
- Access time
- Cycle time
- Bandwidth of storage unit
- Data transportation
- Cost

C) Type and characteristics of the memory unit

Memory units can be classified according to the use, physical storing method, data retention, and content preservation.

〈Table 12〉 Types of Memory Units

Classification Criteria		Description	Type
Usage	Main memory device	Memory device to temporarily store a program or data that a computer processes	RAM and ROM
	Auxiliary memory device	A memory device that stores data semi-permanently by compensating for the main memory's insufficient storage capacity and using non-volatile characteristics	Magnetic disk, optical disk, etc.
Physical storage method	Magnetic	A device that stores binary data according to the magnetic flux direction by maintaining magnetism	Magnetic tape, hard disk, zip drive, etc.
	Optical	A device that records data on the surface of an aluminum metallic disk using a laser beam	CD (Compact Disc) DVD (Digital Versatile Disc) BDA (Blu-Ray Disk)
	Semiconductor	A device that stores analog data using integrated circuit technology	RAM, ROM, and flash memory
Data aintainable when power is off	Volatile	The memory that loses all data when power is off	RAM-based SSD (Solid State Disk)
	Non-volatile	The memory that preserves stored data even when power is off	ROM, magnetic core, and auxiliary memory device
Access method	Sequential access	To access the required location in the memory space sequentially from the beginning	Magnetic tape
	Direct access	To directly access the required location in the memory space	Disk, flash memory, etc.
Preservation of data	Destructive	The memory in which the stored data are destroyed after reading	Magnetic core
	Non-destructive	The memory that retains the stored data even after reading	Memory devices except for the magnetic core

D) Addressing mode

An address is the location in the main memory where data is stored. Various addressing methods are available to designate instructions, using limited instruction bits and using the memory unit capacity efficiently.

- Direct addressing mode
- Indirect addressing mode
- Implied addressing mode

- Immediate addressing mode
- Displacement addressing mode: Relative addressing mode, indexed addressing mode, and base-register addressing mode

E) Locality

The locality is a tendency in which programs intensively refer to a specific area in the moment, rather than uniformly accessing information in the memory device.

- Temporal locality: Recently accessed programs or data are more likely to be accessed again in the near future.
- Spatial locality: Data stored adjacent to the storage device is more likely to be accessed continuously.
- Sequential locality: Instructions are fetched and executed in the order in which they were stored, unless branched (about 20%).

04 I/O Device

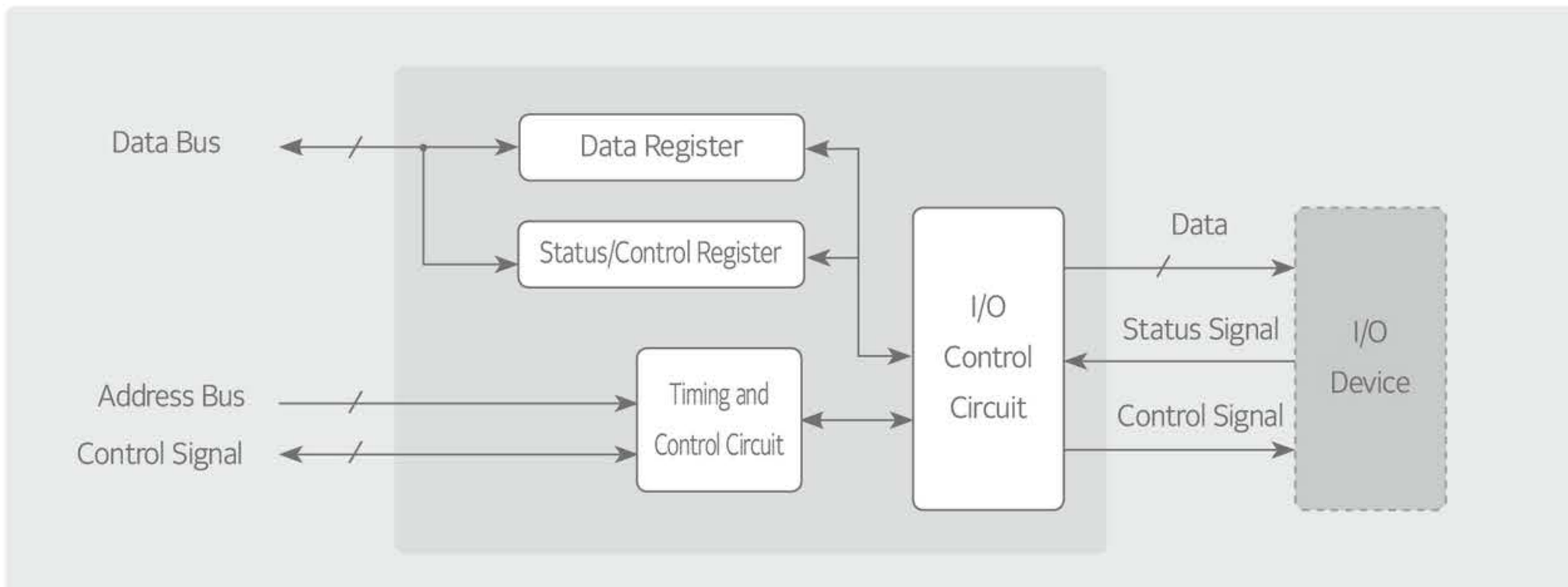
A) Concept of I/O device

The device is necessary to perform an input operation that stores data to be processed by the CPU in the memory unit, as well as an output operation that transfers the processing results from the main memory to an output medium.

B) I/O controller structure and addressing methods

An I/O controller is necessary to process inputs and outputs, as shown in [Figure 36], and it plays the following roles:

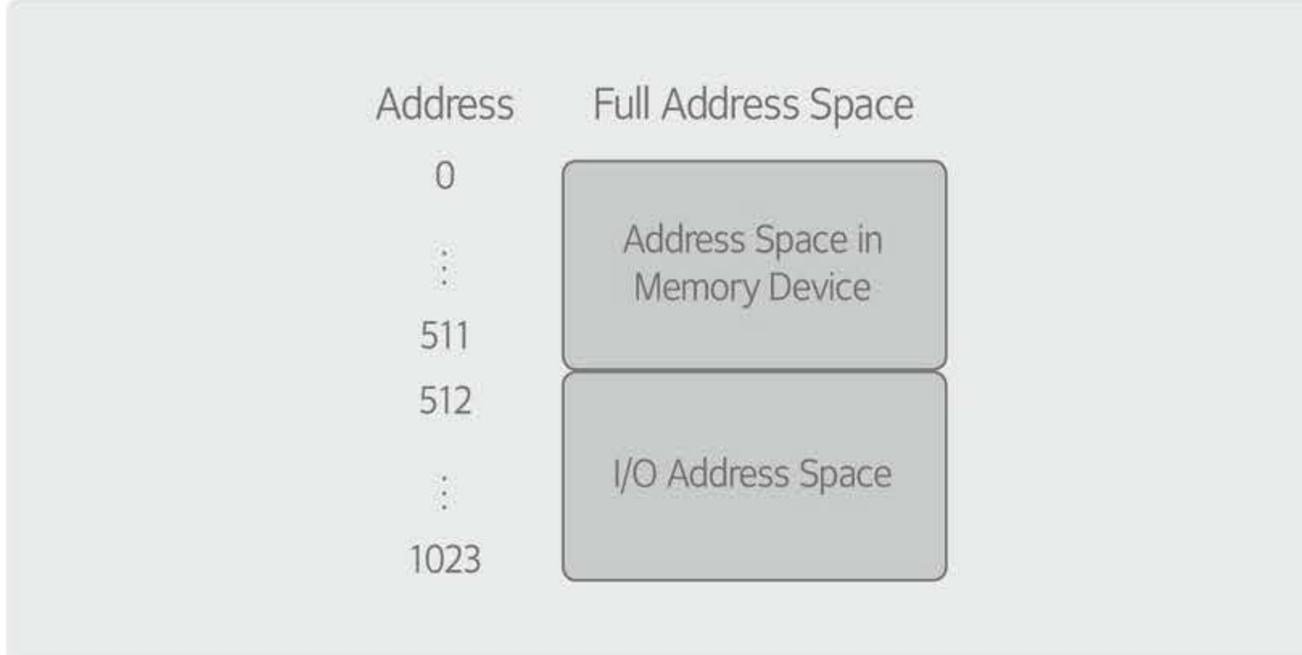
- I/O device control and timing coordination
- Communication with the CPU
- Communication with the I/O device
- Data buffering
- Error detection



[Figure 36] I/O Controller Structure

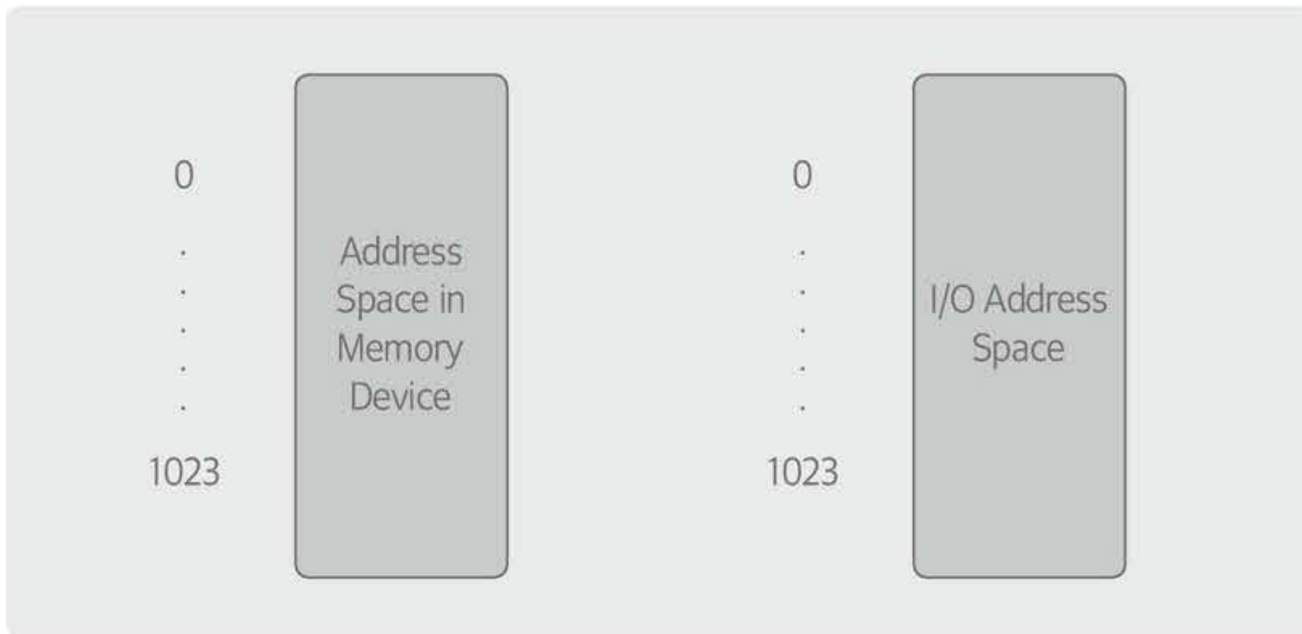
Each device needs two addresses (a status/control register address and a data register address) for I/O control, and the same two addresses are required for each device. It is divided into the memory mapped I/O and the I/O mapped I/O, depending on how the addresses are allocated.

- Memory mapped I/O: It is a method of allocating a part of the address area in the memory to the register addresses in the I/O controller, as shown in [Figure 37]. It has the advantage of easy programming, but the disadvantage of reducing the available memory space.



[Figure 37] Allocation of Memory Mapped I/O Address

- I/O mapped I/O: It is a method of allocating the I/O device address space separately from the memory address space, as shown in [Figure 38]. It has the advantage of not reducing the available memory address space, but the disadvantage of making it difficult to program.



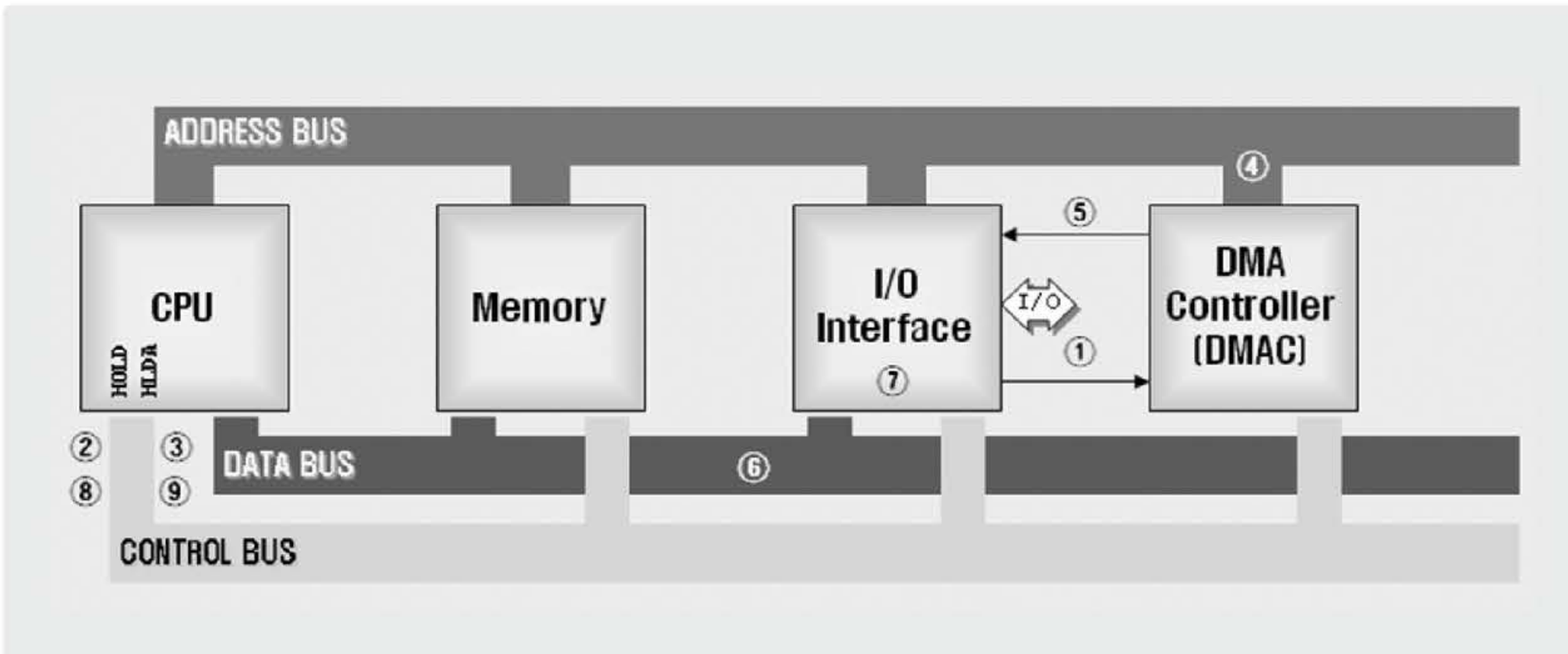
[Figure 38] Address Allocation by I/O Mapped I/O

C) DMA

① Concept of DMA

DMA is a method of the I/O device directly accessing the memory without the assistance of the CPU. The DMA controller controls the bus, and the I/O device and memory transfer information directly. [Figure 39] shows the system structure that includes the DMA controller.

With DMA, high-speed I/O devices can increase system efficiency by minimizing the interruption overhead that reduces the CPU's actual processing time.



[Figure 39] System Structure Including DMA

② DMA operation sequence

① The I/O interface requests a DMA service from the DMA controller.
② The bus request is transferred to the CPU's HOLD pin, so that the DMA acquires the control of the bus (Active High).
③ The bus grant is returned from the CPU's HLDA (Hold Acknowledge) pin, to the DMAC (Active High).
④ The DMAC loads the contents of the address register into the address bus.
⑤ The DMAC transfers the DMA acknowledgment to the I/O interface in order to load the data into the data bus.
⑥ The address bus transfers the data bytes to the identified memory location.
⑦ The I/O interface maintains the data transfer (Latch).
⑧ The DMAC returns the bus grant as the bus request is dropped, and the HOLD becomes a low state.
⑨ The bus grant is dropped, and HLDA becomes a low state.
⑩ Afterward, the address register increases by increments pf 1, and the byte count decreases by decrements of 1.
⑪ The process returns to Step ① if the byte count is not 0, and otherwise stops.

③ DMA operation mode

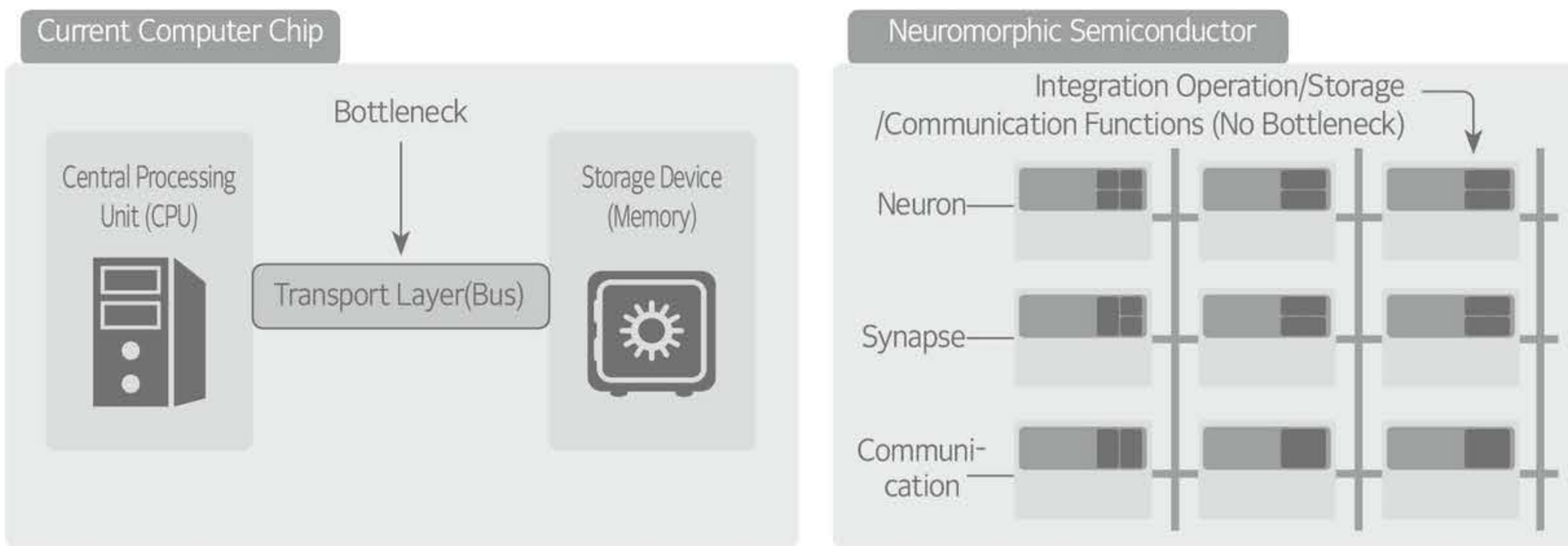
Type	Description
Cycle stealing	- The fast CPU gives the slow DMA the priority of using the bus to enable fast inputs and outputs when the DMA controller and the CPU intend to use the bus at the same time. - It is applied when transferring data of about one word during a DMA operation.
Burst mode	- It is applied when transferring data in blocks during a DMA operation. - It continuously transfers blocks formed by multiple memory words. - It is applicable to high-speed I/O devices. When a DMA interface acquires the bus grant, it monopolizes the bus cycle until the data transfer is complete.

05 Latest Technologies and Trends

A) Neuromorphic chip

Neuromorphic chip, a core technology for neuromorphic computing, is a new semiconductor type that processes information in a way that is similar to human thinking, by implementing brain behavior in silicon as much as possible.

- There are several cores in a semiconductor chip, and electronic devices, such as transistors and memory, are embedded into the core.
- Some of the devices in the core play the role of neurons and neural cells in the brain, and the memory chip plays a synaptic function that links neurons.
- Since the core, which acts as an artificial neuron, is configured in parallel like the human brain, it can process a large amount of data with little power and increase the ability to learn and operate like the human brain.



Existing Semiconductor		Neuromorphic Semiconductor
Cell (Storage/Operation) and Bandwidth (Connection)	Structure	• Neuron (Nerve Function) and Synapse (Signal Transmission)
Storage and operation	Strength	• Sensing image and sound and recognizing patterns
Each semiconductor performing only a specific function	Function	• Storage and operation processed together
Serial (Once I/O at a time)	Data Processing Method	• Parallel (Various data I/Os simultaneously)

[Figure 40] Comparison of Conventional Semiconductor and Neuromorphic Chip

Neuromorphic computers do not only operate in a preprogrammed way, like conventional semiconductor-based computers, but they can develop their processing power by detecting the surrounding context and learning autonomously. Since devices equipped with the neuromorphic chip can increase learning, operation capability, and power efficiency, they can apply intelligence to all computing, ranging from IoT, smartphones, robots, automobiles, cloud computing, and supercomputing.

However, there is still a lack of understanding of how human neural circuits process data at the system level, and neuromorphic computers still remain at a level that simulates single neurons. There are many problems to resolve

from the technical aspect in order for them to have the same level of complex and highly integrated structure as the human brain. Research on neuromorphic processors with various materials and structures is ongoing worldwide, but there is no clear leading group with clear superior competitiveness.

B) Quantum computer

A quantum computer is a new conceptual computer that can simultaneously process a large volume of information at a high speed, based on the principle of ultra-high-speed, large-capacity computing technology optimized for specific operations, according to the principle of overlapping and entanglement inherent in quantum mechanics.
Quantum parallel processing, which uses quantum bits (qubits) as a basic unit of information processing, exponentially increases information processing and computation speed. <Table 13> shows the difference from conventional computer structures.

〈Table 13〉 Comparison of Conventional and Quantum Computers

Subject	Existing Computer	Quantum Computer
Operation concept diagram	0 1 Expression of information in 0 or 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 0 0 1 0 0 1 1 1 0 1 0 1 1 0 1 1 0 1 1 0 0 0 0 1 Sequential calculation one at a time Much time required	0 1 Overlapping 0's and 1's Combining and calculating at once 1 0 0 1 1 0 1 1 0 0 1 0 0 Instantaneous calculation
Basic unit	Bit (0 or 1)	Qubit
Operation method	Calculation with a logic table	Calculation by matrix function
External noise	Easy to correct errors	Difficulty correcting errors
n bits of information	Remember only one value out of 0 - 2^n-1	Remembering all values of 2n (overlap)
Operational throughput	One operation of n bit ALU	2^n operations of n qubit ALU
	① 0 0 0 ② 0 0 1 ③ 0 1 0 ④ 0 1 1 ⑤ 1 0 0 ⑥ 1 0 1 ⑦ 1 1 0 ⑧ 1 1 1 Eight information processing in the case of 3 bits (Repeated calculation)	0/1 0/1 0/1 One information processing in the case of 3 qubits (Simultaneous calculation)

A quantum computer can be an analog or digital type. <Table 14> shows the characteristics of each type.

〈Table 14〉 Types of Quantum Computer and Comparison

Subject			Features
Analog type (Quantum ising machine)	Quantum Annealing type		- Low versatility (Combination optimization calculation) - Commercialized by D-Wave Systems in 2011
	Laser network type	Quantum neural network (QNN)	- Low versatility (Combination optimization calculation) - Operated at room temperature and low cost
		…	

Digital type	Quantum gate type	Superconductive qubit type	- Universal (Similar to computers in principle)
		Spin qubit type	
		...	

Subject		Quantum Annealing type	Laser network type	Quantum gate type
Features	Application range	Specialized • Combination optimization/ sampling (AI)	Specialized • Combination optimization/ sampling (AI)	Universal • Factorization (Decryption), quantum simulation (chemistry), etc.
	Technical features	• Calculation principle: Hamilton insulation change • Hardware noise withstanding: High • Operation temperature: Cryogenic (10mK)※ • Error correction method: Not finalized	• Calculation principle: Quantum phase transition • Hardware noise withstanding: High • Operation temperature: Room temperature (300K) • Error correction method: Not finalized • Others: Excellent insulation performance	• Calculation principle: Unitary rotation of the status vector • Hardware noise withstanding: Weak • Operation temperature: Cryogenic (10mK) • Error correction method: Finalized
	Implementation status	About 2,000 qubits • Superconductive	About 2,048 qubits • Photon	About 20 qubits (50 qubits for prototype) • Superconductive, ion, photon, quantum dots, etc.
Commer-cialization status	Commercialized time	• May 2011	• To be determined	• To be commercialized in 2020-21
	Developing enterprises/ organizations	• D-Wave Systems, Google, US national project IARPA QEO8) • Operational test by NASA, Denso, and others	• Innovative R&D program (ImPACT) by Japanese Cabinet Office: NTT, National Institute of Informatics (NII), and The University of Tokyo • NII-Stanford University	• and others • Twelve global companies and universities including Samsung Electronics participating in IBM's fundamental research consortium (December 2017)
	Recent status	• D-Wave 2000Q launched (January 2017) • Kicked off a 5-year project by the US IAPRA (QEO) (April 2017) • Quantum Anneater Ver. 2 being developed by Google	• Japan's ImPACT disclosed a system to experience in a quantum neural network in the cloud beginning on November 27, 2017.	• IBM is providing IBM Q in the cloud having processed more than one million cases for 54,000 people. • Google disclosed a 72-qubit quantum process in March 2018.

V. Data Processing Technology

▶▶▶ Recent trends and issues

Technologies that have recently become issues in system architecture are cloud computing, AI, and big data. The learning subject related to these is data processing technology. The subject closely related to AI is the parallel processing system. The subject related to big data is storage technology. The subjects related to cloud computing are storage, high availability storage devices, and graphics.

▶▶▶ Learning objective

1. To be able to explain the parallel processing technology and operational principle.
2. To be able to explain the storage technology and operational principle.
3. To be able to explain the image and image compression technology, and graphics processing technology.

▶▶▶ Keywords

Flynn's taxonomy, parallel processing technology, parallel program technology, disk scheduling, SAN, NAS, DAS, LTO, VTL, RAID, GPGPU, video compression standards

✚ Preview for practical business "Intel vs. NVIDIA vs. Google, who will win the AI market? - Targeting based on parallel processing capability and market domination, Google looking to improve power efficiency using TPU", Monthly Electronic Parts News

Competition in parallel processing technology is intensifying as AI has emerged as a core technology of the 4th Industrial Revolution. While existing programs were implemented by a CPU-based serial (sequential) processing method, GPU-based programs use parallel processing technology, which simultaneously processes multiple instructions and is used to implement deep learning of AI. Google is increasing the parallel processing capability by adding its own processor, called TPU.

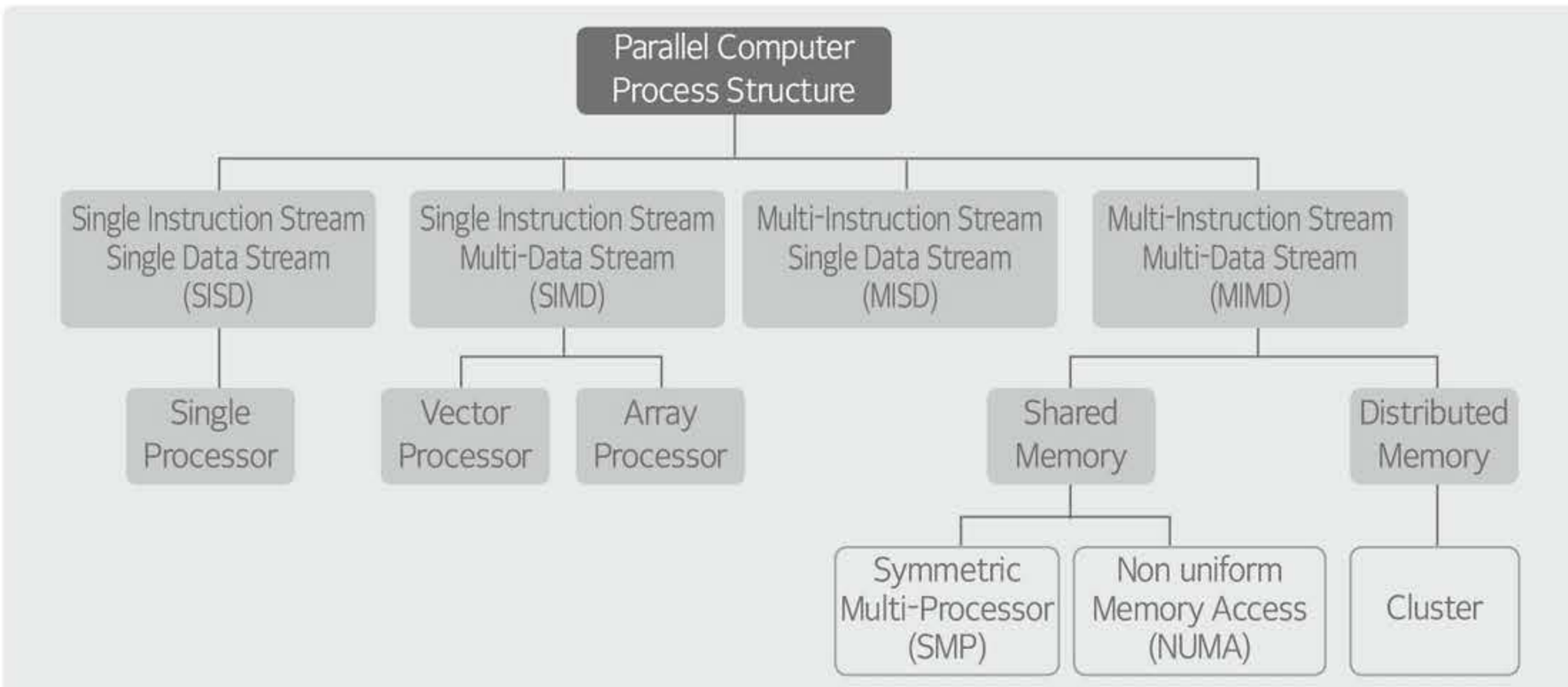
[Source: Monthly Electronic Components News]

01 Parallel Processing System

A) Concept of the parallel processing system

Parallel processing refers to one or more independent operating systems managing multiple processors and performing multiple tasks. There are separate programming languages and syntax for parallel processing. Parallel processing is very fast and can share the memory unit. Since several processors operate it, the impact of failed hardware parts on the entire system is small. It is widely used in the field of AI, consisting of a lot of data and repetitive complex operations, military equipment that is needed to obtain accurate results in a short time, and searching services that extract results from a wide range of data in a short time. Flynn's taxonomy and classification by memory structure are leading examples of parallel processing system classification.

B) Flynn's classification of parallel processing systems



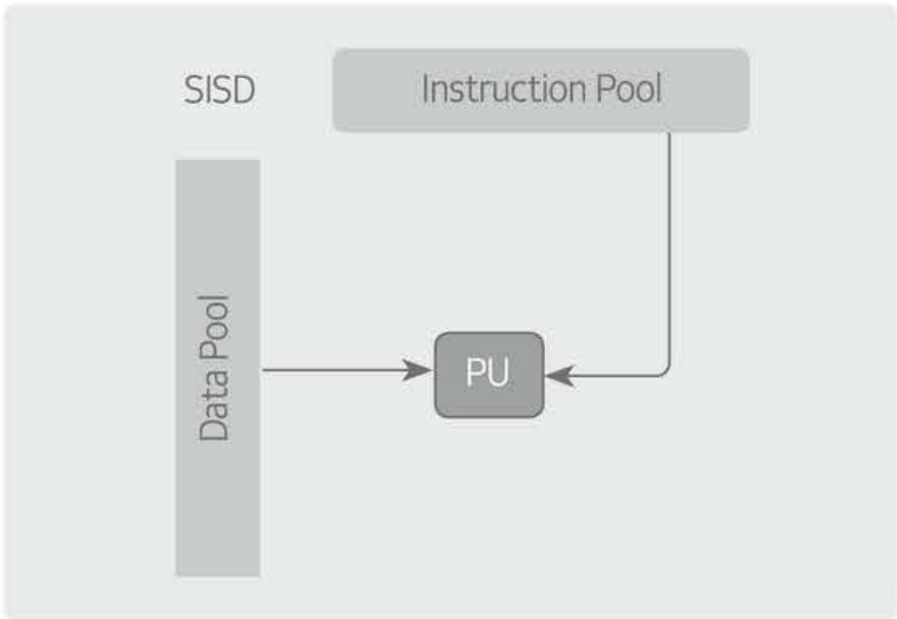
[Figure 41] Structure of Parallel Computer Processor

① Single instruction stream - single data stream, Single Instruction Stream Single Data Stream (SISD)

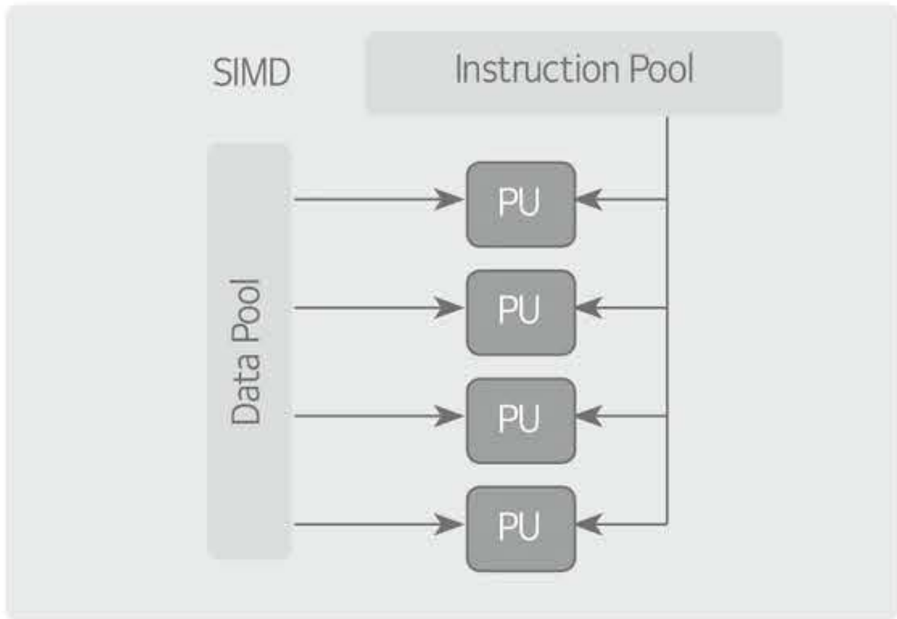
SISD is a single processor system that sequentially processes an instruction and data, one at a time. It is the conventional computer architecture that follows von Neumann's concept. The controller interprets an instruction and operates the processor in order to run the instruction while fetching a piece of data from the memory unit and processing it. The performance may deteriorate, since it requires reading the instruction each time it acquires the processing data. The performance may be improved by using simultaneous processing techniques, such as pipelining and superscalar.

② Single instruction stream - multiple data stream, Single Instruction Stream Multiple Data Stream (SIMD)

The structure of processing multiple data with an instruction to simultaneously perform the same operation on multiple data. It is also called an array processor, as it enables synchronous parallel processing. Intel's processor, made in the SIMD structure, is the Pentium processor with the MMX instruction set. The name is "multimedia acceleration," which is the code that can quickly process frequently-used floating-point arithmetic operations. The processing speed is high because it can process multiple operations with an instruction.



[Figure 42] SISD Structure



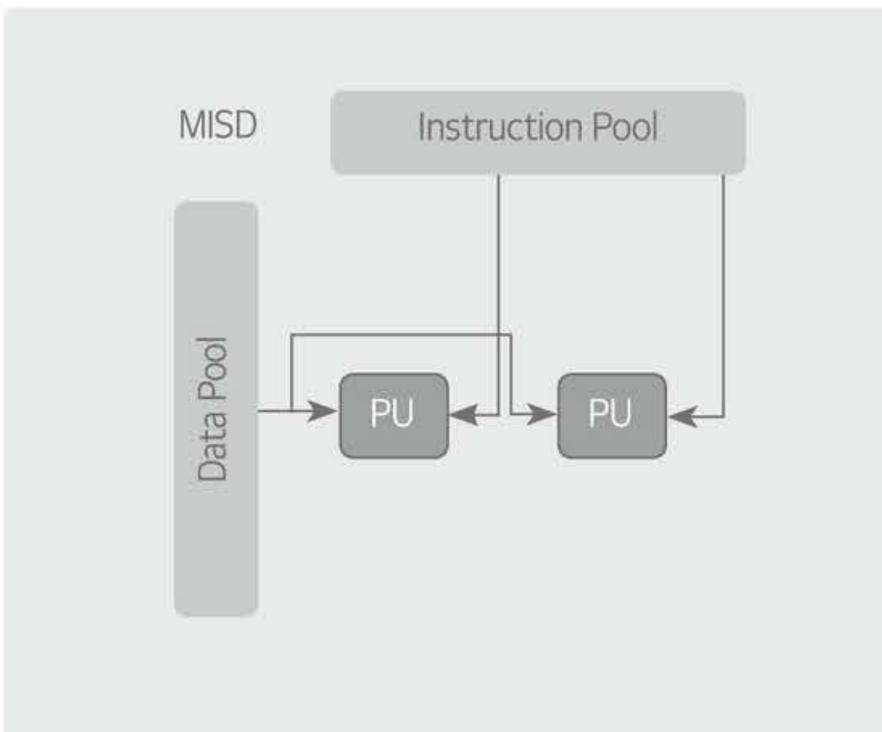
[Figure 43] SIMD Structure

③ Multiple instruction streams - single data stream, Multiple Instruction Stream Single Data Stream (MISD)

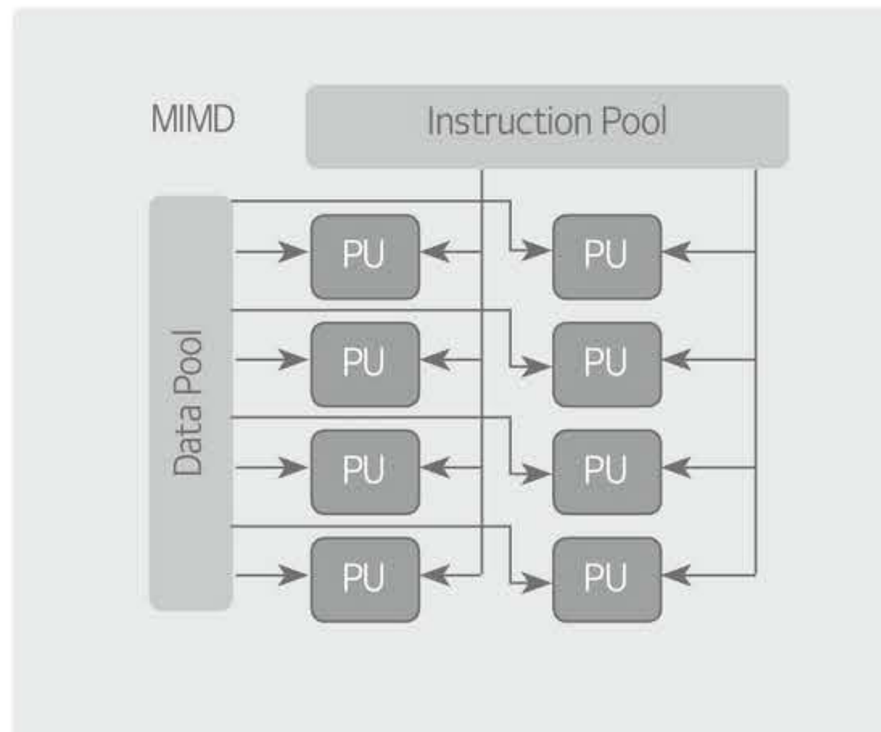
Each processing unit in the MISD parallel computing architecture runs different instructions and processes the same data. The pipeline architecture is an example. It is not a widely used architecture.

④ Multiple instruction streams - multiple data stream, Multiple Instruction Stream Multiple Data Stream (MIMD)

In a MIMD structure, multiple processors process different programs and different data, and most parallel computers fall into this category. It can be classified into a shared memory model and a distributed memory model, depending on how it uses the memory. According to the degree of data interaction, the shared memory model is called a tightly-coupled system, and the distributed memory model is classified as a loosely-coupled system.



[Figure 44] MISD Structure

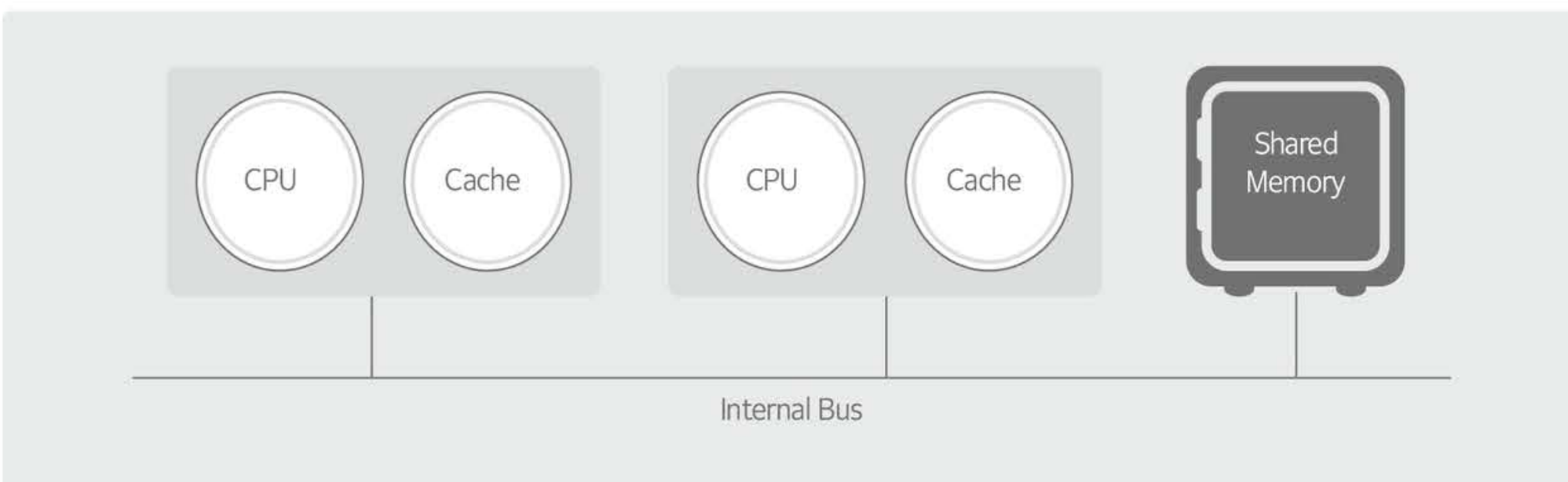


[Figure 45] MIMD Structure

C) Classification of parallel processing systems, according to the memory structure

① Symmetric multiprocessor (SMP)

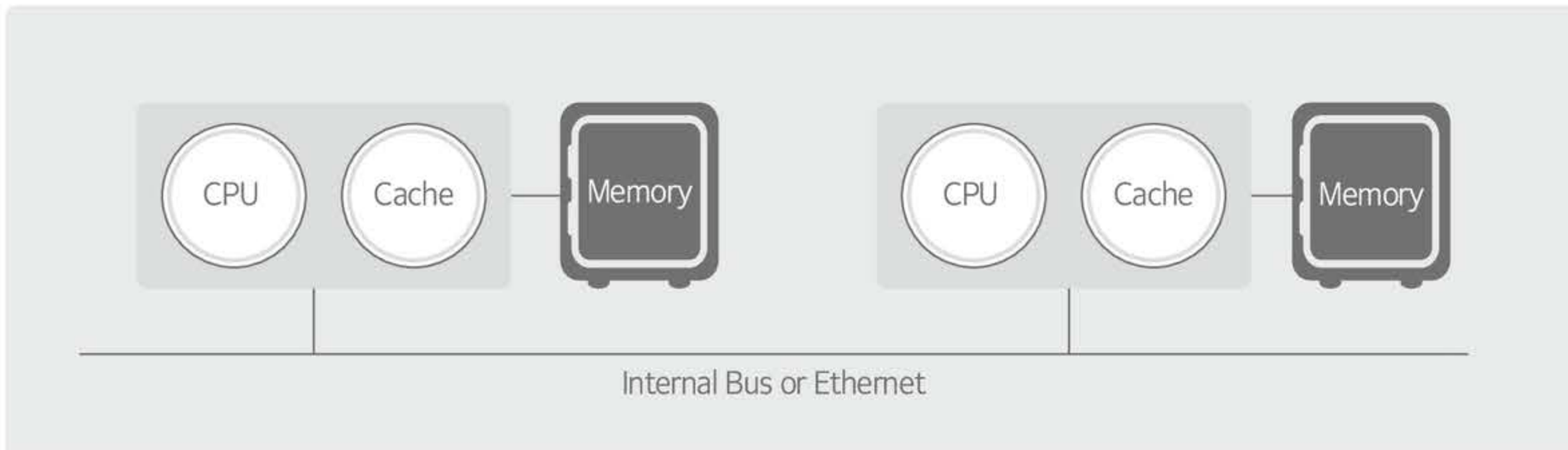
SMP is a tightly-coupled system in which all processors use the main memory as the shared memory. It is easy to program since the data transfer can use shared memory. On the other hand, it has poor scalability and may experience a bus bottleneck, since it uses an internal bus to access the main memory and share the data.



[Figure 46] SMP Structure

② Massive parallel processor (MPP)

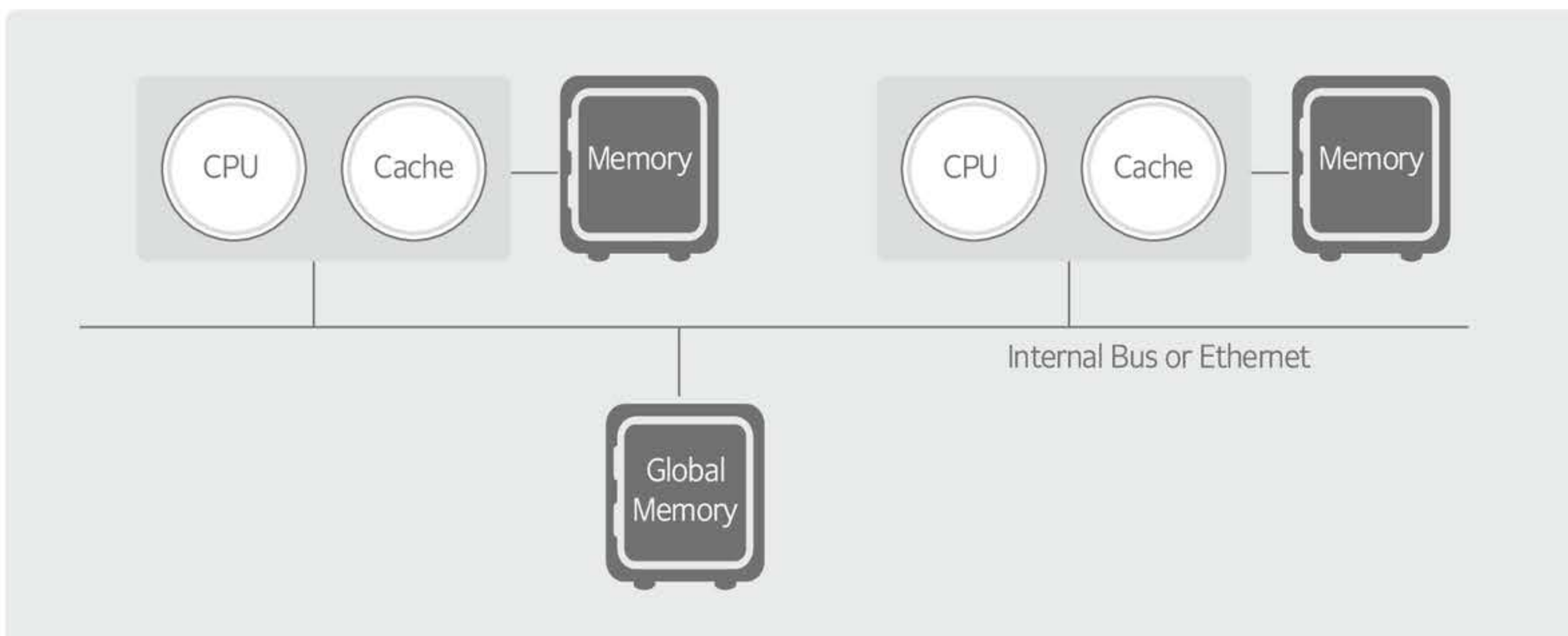
MPP is a distributed memory type in which each processor has an independent memory. The loosely coupled system exchanges data between processors through a network, such as Ethernet. The structure of connecting nodes, like the CPU, memory, and I/O device, each of which has its own system resources, through a network, offers excellent scalability and programming difficulty.



[Figure 47] MMP Structure

③ Non uniform memory access (NUMA)

NUMA is a structure the combines the advantages of the SMP, which is a shared memory structure that makes it easier to develop programs and the MPP structure, which offers excellent scalability. Each processor has a separate local memory and the global memory address space that all processors can access.



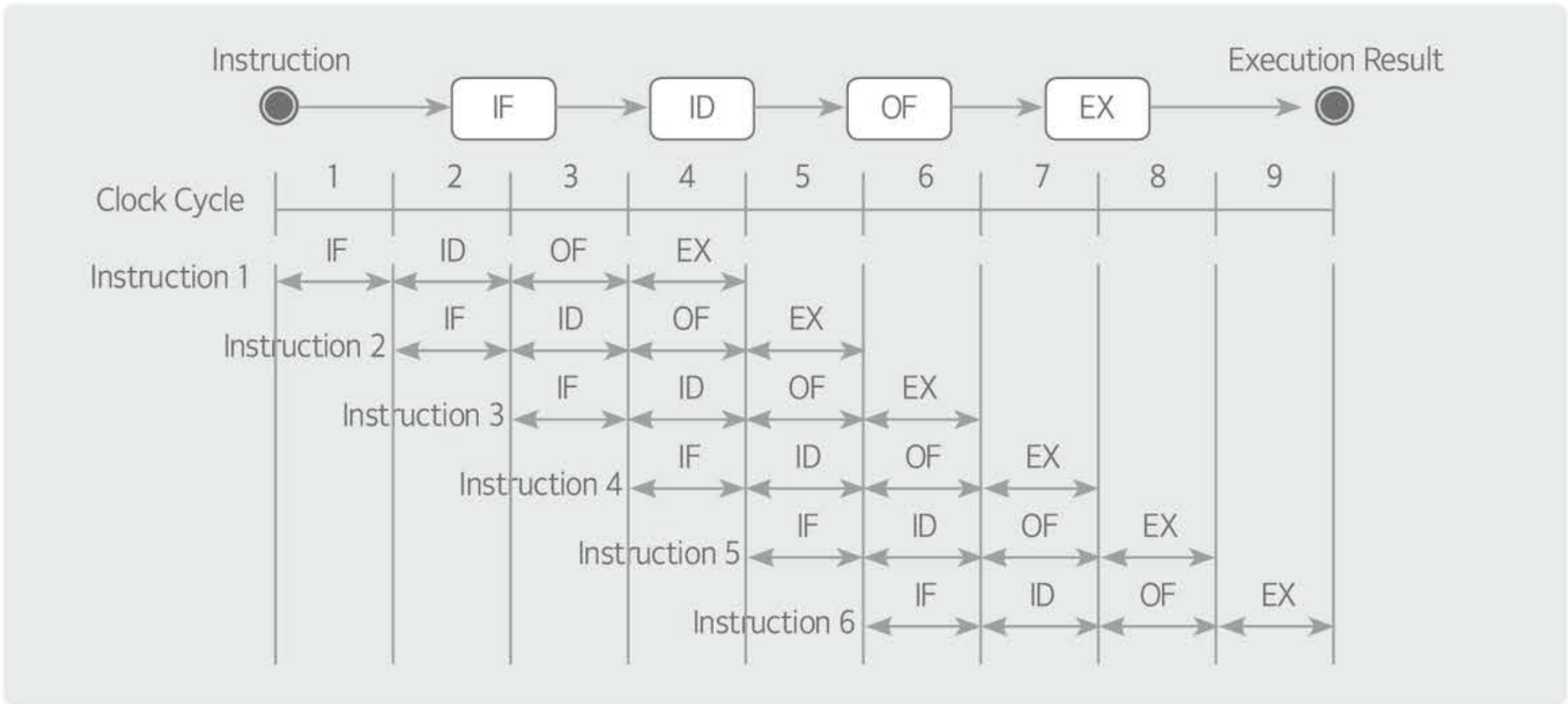
[Figure 48] NUMA Structure

D) Types of parallel processor technology

① Instruction pipelining

The technology improves the CPU performance by dividing an operation into several stages and configuring a hardware unit for processing each stage separately in order to process different instructions simultaneously. In other words, it does not process only one instruction at a time, but it processes multiple instructions simultaneously by

processing another instruction while still processing an original instruction. There are several types of pipelines: a two-stage instruction pipeline, a four-stage instruction pipeline, and a six-stage instruction pipeline. The four-stage instruction pipeline is the most widely used.
Although the instruction set consists of two stages in the four-stage instruction pipeline, the processing time of the stages is divided into 4 in order to prevent efficiency degradation, which occurs because the processing time of the pipeline stages is not the same.
The stages of the four-stage instruction pipeline are instruction fetching (IF), instruction decoding (ID), operand fetching (OF), and execution (EX).

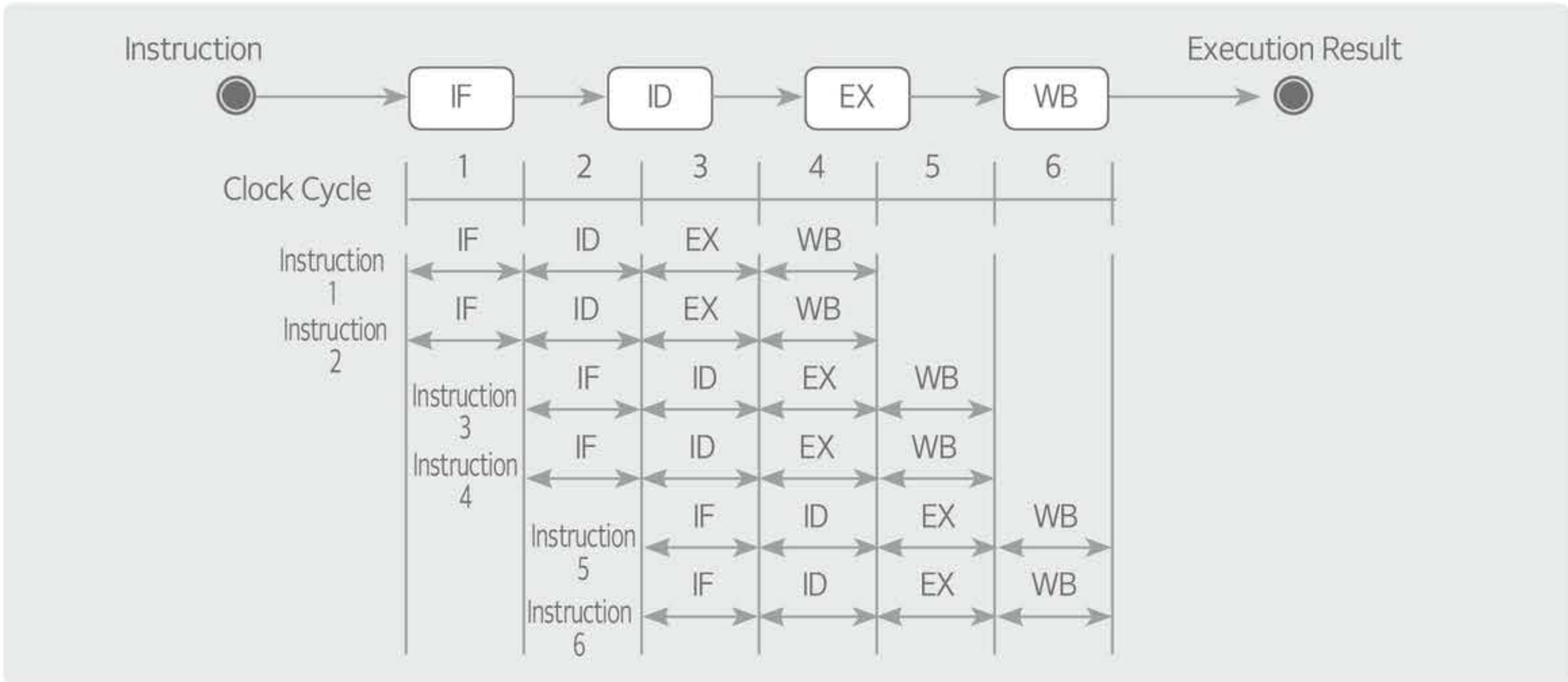


[Figure 49] Pipelining

The IF stage is a process that fetches instructions from the memory unit. It fetches an instruction from the program counter's memory unit address and transfers it to the register. The ID stage interprets the fetched instruction using an instruction decoder. The OF stage fetches an operand that becomes a variable or data, to be used in operation, from the memory unit. The EX stage runs the operation specified by the instruction.

② Superscalar process

The Superscalar process is a structure that includes a number of instruction pipelines to speed up the central processing unit. It can run instructions out of order, since it independently processes instructions using multiple pipelines. The superscalar process has a basic configuration in which several functional units,that are implemented as superscalar pipelines, support the parallel processing of instructions.



[Figure 50] Superscalar Process

③ Pipeline hazard

A pipeline hazard refers to the pipeline speed exceptionally slowing down. Pipeline hazards include the data hazard, the control hazard, and the structural hazard.

Data hazards occur when the next instruction execution has to be delayed until the previous instruction has been completed because of the dependency between instruction operands. Control hazards are generated by branch instructions, like branch and jump,which change the execution order of the instructions. Structural hazards are generated when instructions cannot be processed in parallel in the same clock cycle, due to hardware limitations. In other words, it means the hardware cannot support the combination of instructions in the same clock cycle.

E) Parallel programming technology

① Compiler technology - OpenMP

OpenMP is a compiler directive-based parallel programming API. Here, directive-based means processing only the desired parts in parallel, by adding a directive to a program written sequentially without parallel processing. The execution model of OpenMP is the fork/join model. A program initially operates as a master thread, and when it encounters a directive, it creates threads and independently executes them. When the parallel processing is finished (when the area selected as a directive is finished), they are joined into one master thread in order to run the code sequentially.

② Message passing parallel programming model

The message passing interface is a parallel programming model suitable to a distributed memory system structure. Since MPI is not a memory sharing mode, the nodes share the information by transferring messages over the network. Therefore, the network communication speed is the most important factor for performance, and it is widely used by supercomputers that require a high-speed operation. Parallel programming tools for message passing include High Performance FORTRAN (HPF), Parallel Virtual Machine (PVM), and Message Passing Interface (MPI). MPI has become the standard.

③ Load balancing technologies - AMP, SMP, and BMP

Load balancing adequately distributes jobs to the cores in order to increase the multi-core performance. The multiprocessing models include asymmetric multiprocessing (AMP), symmetric multiprocessing (SMP), and bound multiprocessing (BMP).

- AMP: An OS is executed independently in each processor core.
- SMP: An OS manages all processor cores simultaneously. Application programs can operate in any core.
- BMP model: An OS manages all process cores simultaneously, and an application program can run on a specific core.

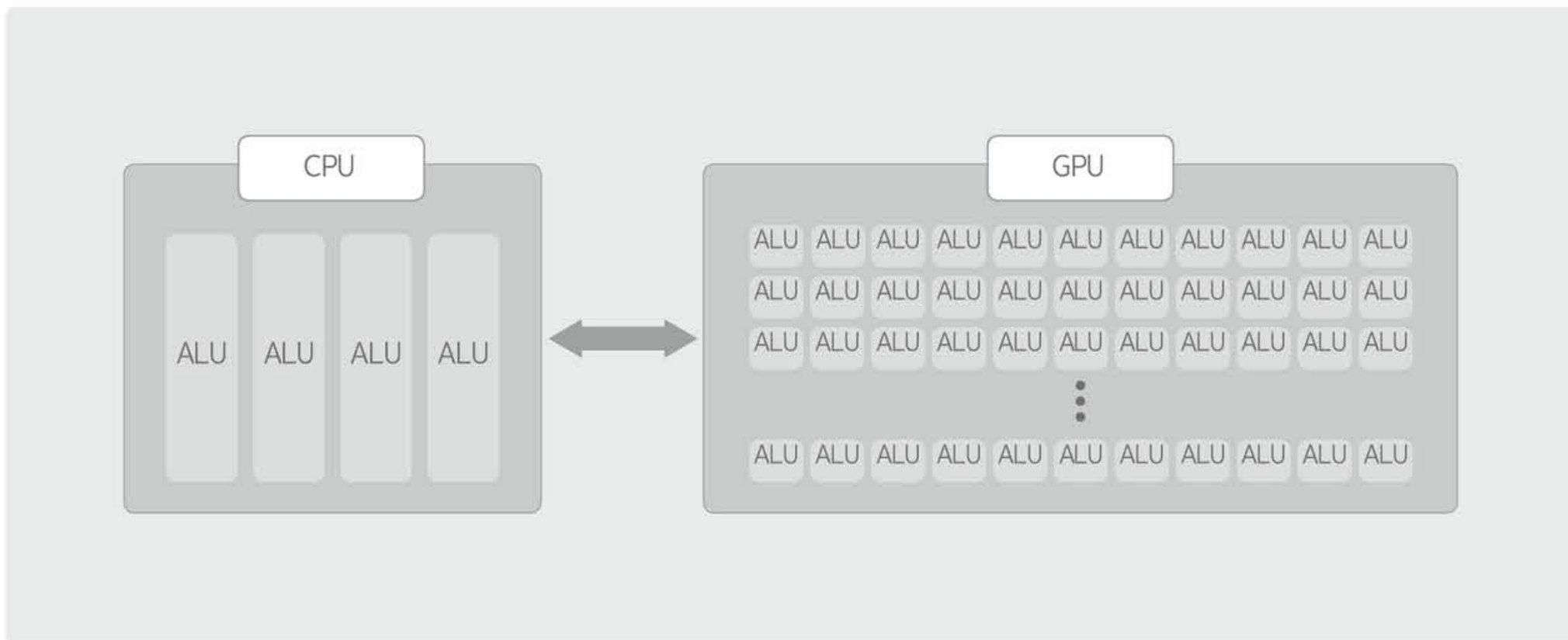
F) Graphic processing technology

① Graphics processing unit (GPU)

The hardware specializes in computer graphics calculation and is mainly used for the rendering of 3D graphics. Since a GPU is configured with thousands of small cores that perform floating-point operations processed in parallel, its performance is superior to a CPU that is configured with a small number of cores. A GPU dedicated to processing large-capacity image data generates results through parallel jobs using multiple cores. Although recent GPUs were mainly used for graphics processing functions, they are evolving into more flexible, programmable GPUs.

② General-purpose GPU (GPGPU)

Based on the fact that a GPU shows high computational performance in matrix and vector operations that are mostly used for graphic rendering, the computing system intends to utilize GPUs in the general computing domain as well. Many models supporting GPGPU programming have appeared. They include CUDA and OpeACC from NVIDIA, OpenCL from Khronos Group, and C++ AMP from Microsoft.



[Figure 51] Difference of CPU and GPU Structures

G) GPU-based parallel programming technology

① Compute Unified Device Architecture (CUDA)

In 2006, NVIDIA introduced CUDA, a tool for GPU development. CUDA is a parallel computing platform and a programming model that can significantly improve computing speed with a large number of GPU cores. It provides

intuitive GPI programming, based on the C language, and it enables quick operation using shared memory. CUDA consists of CUDA Runtime API and Driver API. Runtime API provides user-friendliness by automatically allocating necessary values for settings and others. Driver API, which helps the Runtime API to operate, allows the programs to directly manage memory or devices without using Runtime API. The CUDA is expected to show an excellent performance improvement when applied to performing tasks suitable for parallel processing operations in various fields that require a large amount of computation, such as simulation. Excellent performance can also be expected.

② Open Computing Language (OpenCL)

OpenCL, maintained and managed by Khronos Group, is an open, general-purpose parallel computing framework developed by Apple, AMD, IBM, Intel, and NVIDIA. It is an industry standard programming model for heterogeneous computer systems, consisting of GPUs, CPUs, and other processors. It is a parallel programming model based on data and tasks and has characteristics of the open specification to use GPU, CPU, and DSP. OpenCL mainly consists of the OpenCL compiler and the OpenCL runtime library. Programs running on a computational processor (accelerator) are written in OpenCL C language, and the OpenCL compiler gathers the source code to convert it into a binary code. Moreover, a control program to run or manage programs running on a CPU is necessary. It can be written using C or C++ through the OpenCL runtime library.

③ C++ Accelerated Massive Parallelism (C++ AMP)

C++ AMP was developed by Microsoft in an open programming language for heterogeneous computing, using CPU and GPU. C++ AMP, when added to Visual Studio 2021, can increase the execution speed of C++ codes using GPU. The C++ language-based C++ AMP intends to help developers create general-purpose programs using GPU without a high level of understanding or application capability about DirectX API.

④ Open ACC

Although the concept and usefulness of GPU programming have been widely known with the emergence of platforms like CUDA, it is not easy to write a program that can expect a significant performance improvement by making good use of parallel computing. NVIDIA introduced OpenACC, a programming model based on compiler directives that abstract CUDA. Open ACC is a programming model for higher productivity, since it provides a relatively simple programming environment for developers. It has another advantage of having lower dependence on an OS and platform, since it is a cross-platform model, unlike the above programming models.

02 Storage Technology

A) Concept of storage

Computer systems use a storage unit to access data and run commands. Although a system uses main memory for the main storage unit, it uses auxiliary memory to permanently store and utilize data. The web server, WAS, and the database of information systems also need a storage unit as the permanent auxiliary memory unit. The web server, WAS requires a storage unit to store its OS or the binary files of its application program. Although the data used by an information system is stored and managed through a database, the storage unit is necessary to ensure that data

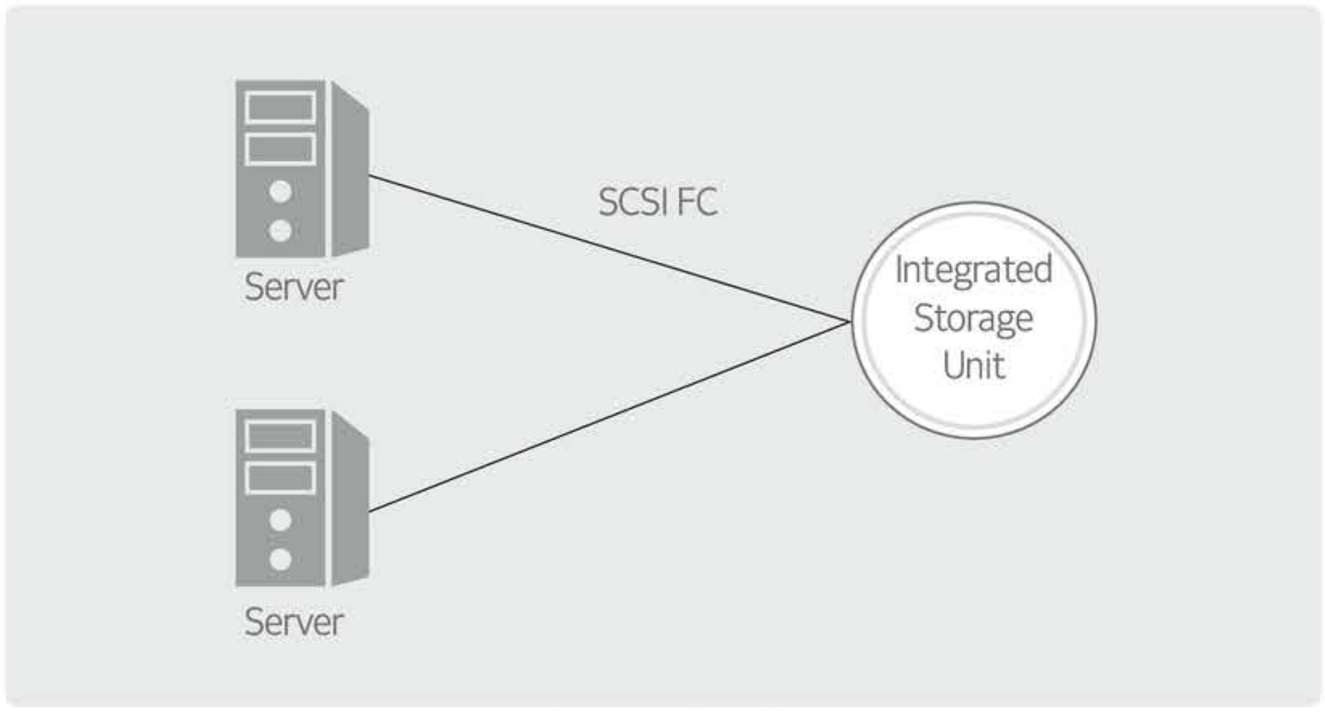
is not corrupted or lost.

B) Connection of storage unit and server

Multimedia services, using a large volume of data, led to computer systems storing an increased volume of data. A large-capacity storage system is necessary, since a single disk cannot support the increasing data capacity. A storage system logically groups multiple disks in order to store large capacity data that a single disk cannot handle. It is classified into DAS, NAS, and SAN, depending on how it is connected to the computer.

① Direct attached storage (DAS)

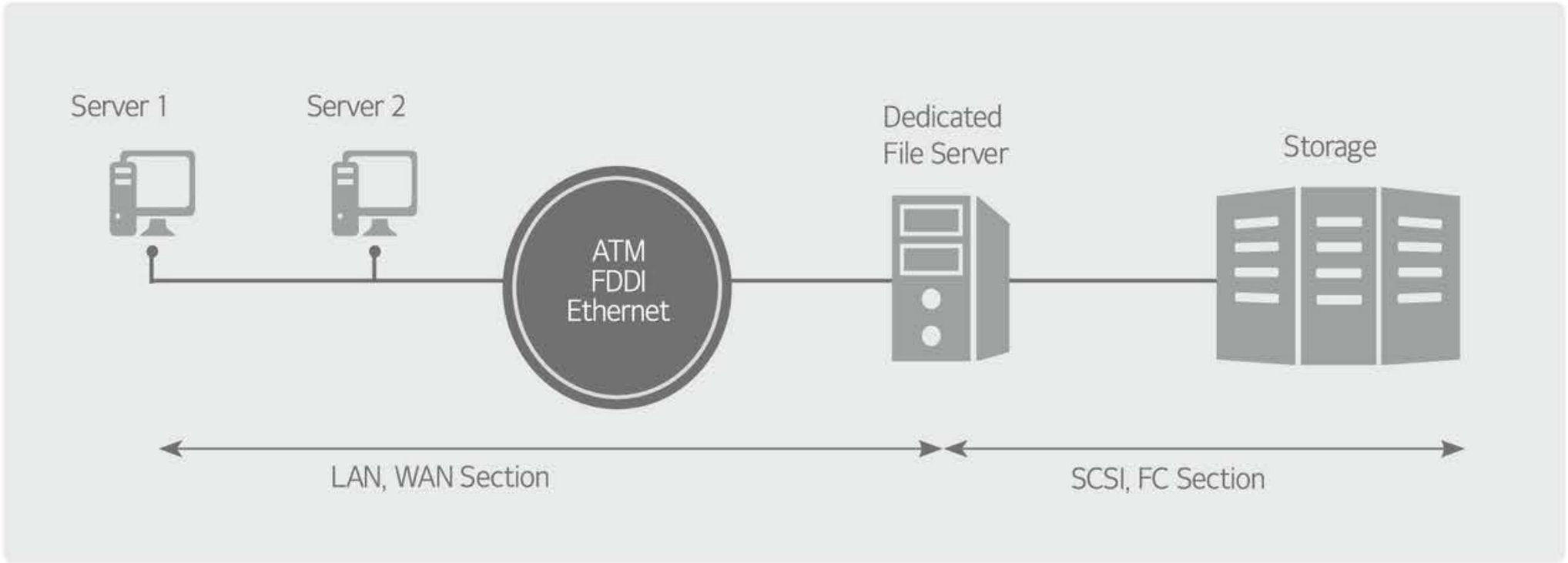
The storage connects a computer system with disks directly through a fiber channel or SCSI cable in order to utilize the storage capacity. It allows the computer system to manage the file system directory. Since the storage is directly connected to the computer system, it works like an external hard disk at high speed. Its advantage is that its configuration is simple, and the cost is low without a separate switch, but there is a limitation in the number of disks that can be connected. Moreover, files in the storage cannot be shared with other computers, since it is directly connected to a computer.



[Figure 52] DAS

② Network attached storage (NAS)

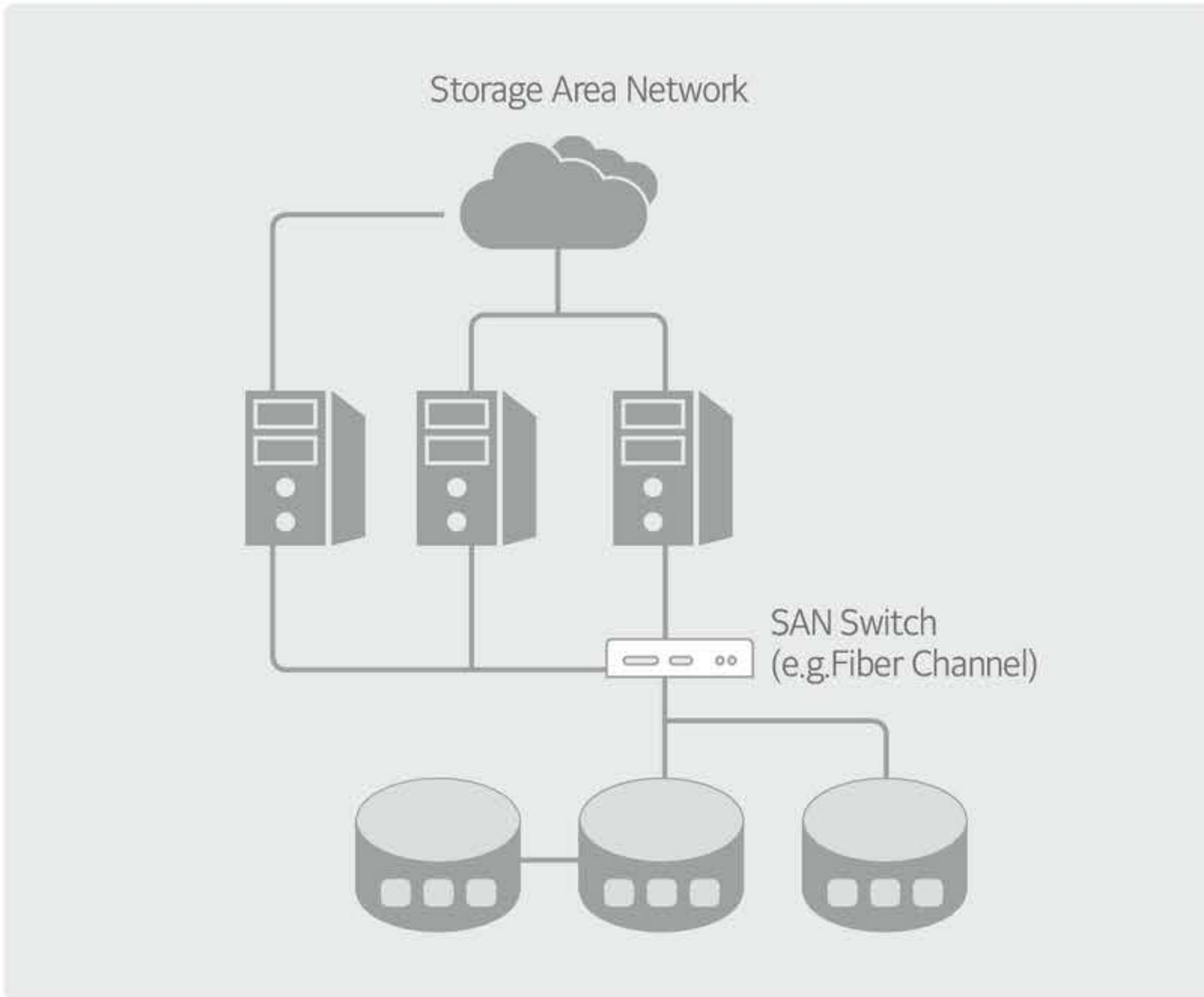
The storage has a separate file system management server (controller) to manage the storage media such as HDD and SSD. It is connected to a computer system through an Ethernet network interface like LAN or WAN. Since it has a separate file system management server, it is easier to manage data, and multiple servers can share the storage. Moreover, multiple computer systems on the same network can use the storage, regardless of the physical location. As the storage is connected over a network, its speed and capacity are limited by the network speed.



[Figure 53] NAS

③ Storage area network (SAN)

A disadvantage of the DAS type is the limitation of the number of disks, and the difficulty of management, while the NAS type has a problem of possible speed delay, depending on the network connection. SAN was developed to overcome the disadvantages. It uses a dedicated fiber channel switch for fast connection, and it enabled the ability to scale up the number of connected servers and storage, with less impact to the connected network load. Using the fiber channel allows the connection to speed up to 8Gbps - 16Gbps. However, it increases costs since it requires the dedicated switch and expensive cables, and it can cause a consistency problem, due to locking, when multiple computer systems share a specific file.



[Figure 54] SAN

C) IP-SAN

① Concept of IP-SAN

This type of SAN uses the gigabit Ethernet Internet protocol (IP), instead of a fiber channel. While the SAN requires a SAN switch and SAN storage disks, it increases interconnectivity, since it is connected using the existing Ethernet network. It can unify the network management and overcome the distance limitation of SAN, since it can use an IP. IP-SAN includes FCIP, iFCP, and iSCSI, while iSCSI is the most widely-used type.

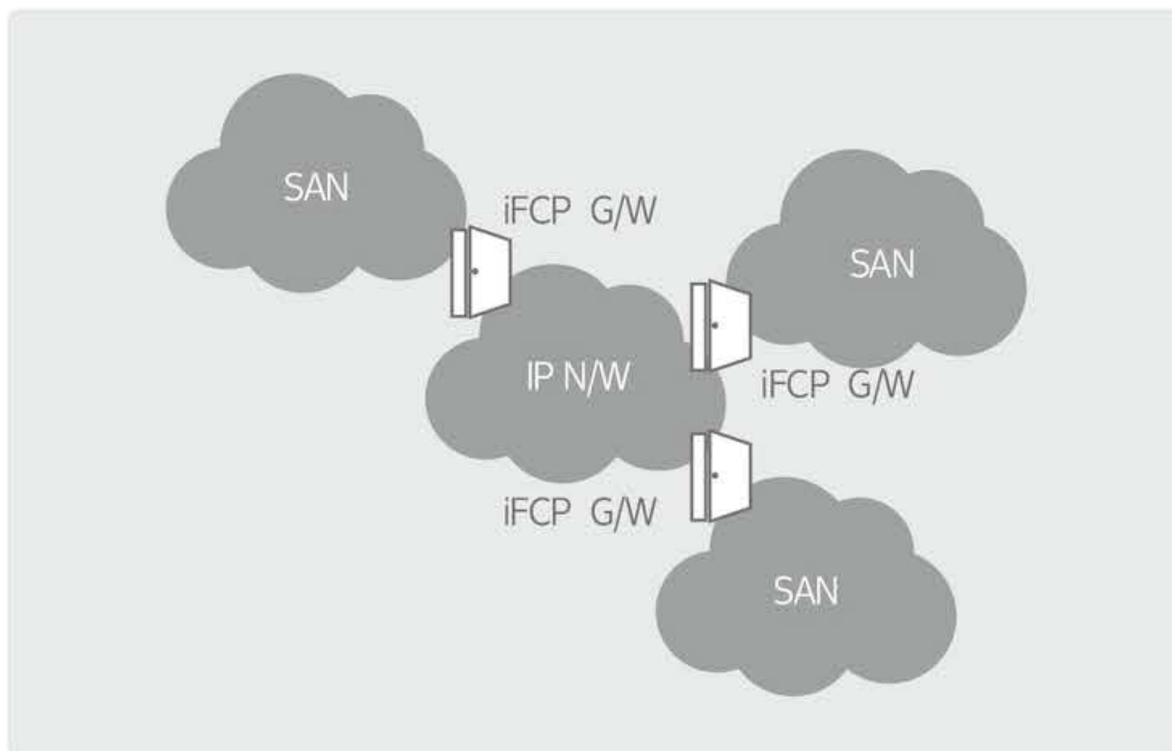
② Fiber Channel over IP (FCIP)

FCIP is used to connect a remote SAN. It encapsulates data to TCP/IP for interconnection when transferring a frame to a remote location. It groups the scattered SANs into a massive fiber channel fabric. A failure of a SAN can affect SANs in other regions included in the fabric.

[Figure 55] FCIP

③ Internet fiber channel protocol (iFCP)

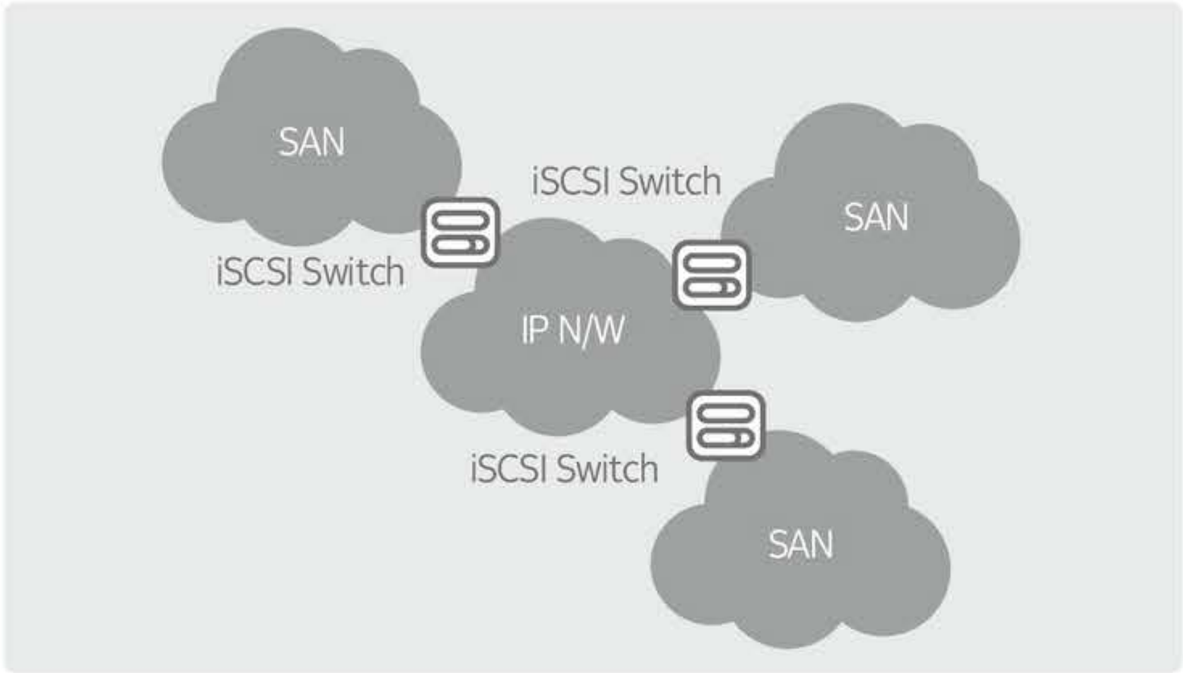
iFCP provides a TCP/IP connection dedicated to regional SAN, using the iFCP gateway. Unlike FCIP, a SAN failure does not affect SAN in other regions, since it does not create a massive fabric. Since it converts protocols using a gateway and can be implemented without changing the infrastructure, it provides highly compatible interconnectivity in both hardware and software.



[Figure 56] iFCP

④ Internet SCSI (iSCSI)

iSCSI encapsulates SCSI commands into IP packets and transfer the I/O block data through TCP/IP. Technologies like IPSec ensure reliability. It reduces the network storage implementation cost, since it can use the existing network environment as is.

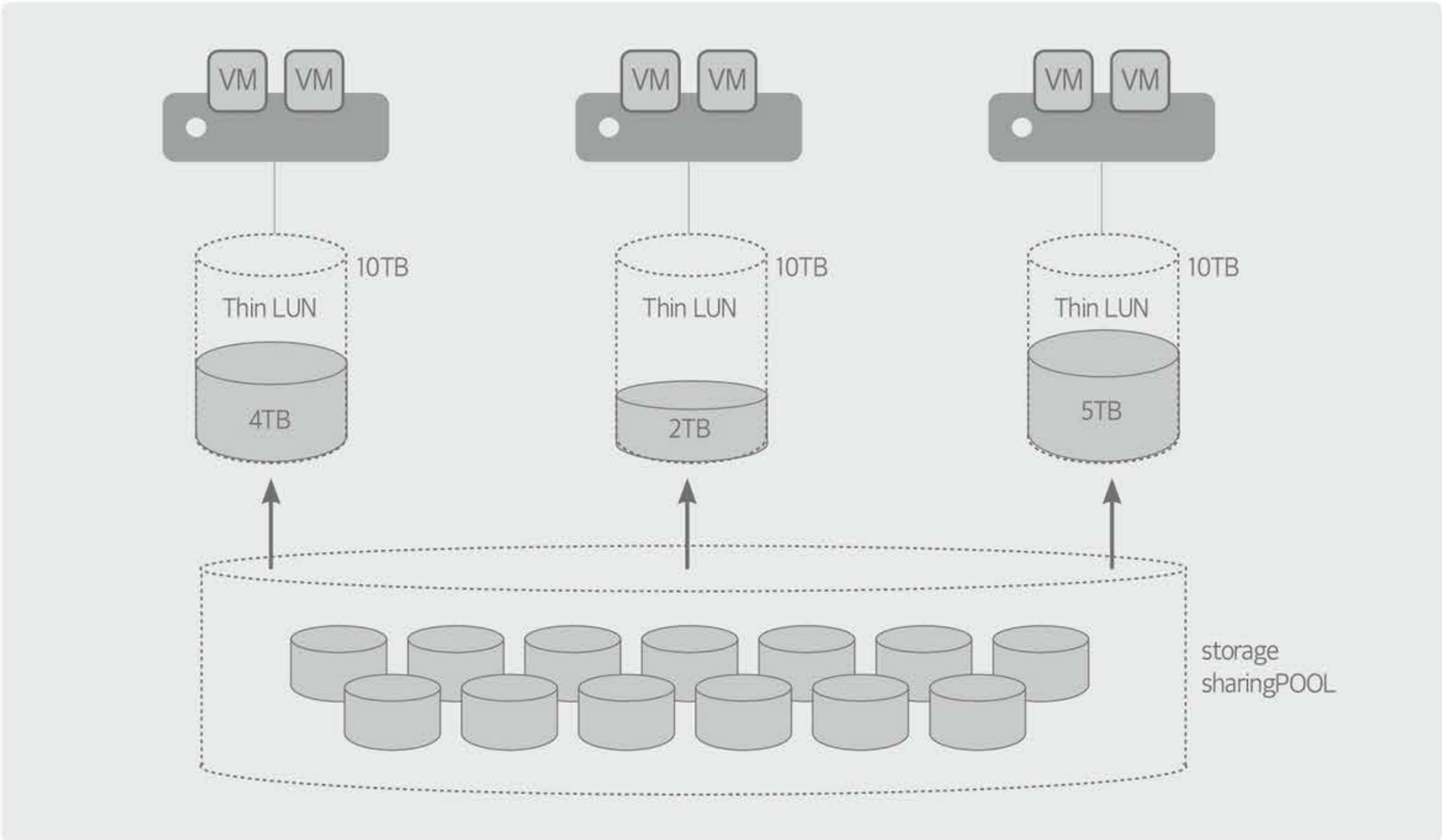


[Figure 57] iSCSI

D) Storage capacity management

① Thin provisioning

The existing fixed-allocation storage technology uses a thick logical unit number (LUN) wasted data storage space. Thin provisioning is a storage virtualization technology that allocates data by mapping actual data space into a thin LUN. It can flexibly expand the disk space required by users in cloud computing, and it offers a higher data utilization rate than the conventional thick LUN type.



[Figure 58] Inexpensive Thin Provisioning

② Data de-duplication

The technology provides a high efficiency of disk space used by removing any duplicated data when saving the data. The removal methods include an inline method, which immediately removes redundant data as it enters the storage system, and an offline method, which removes the redundant data after storing the new data.

E) Storage disk scheduling

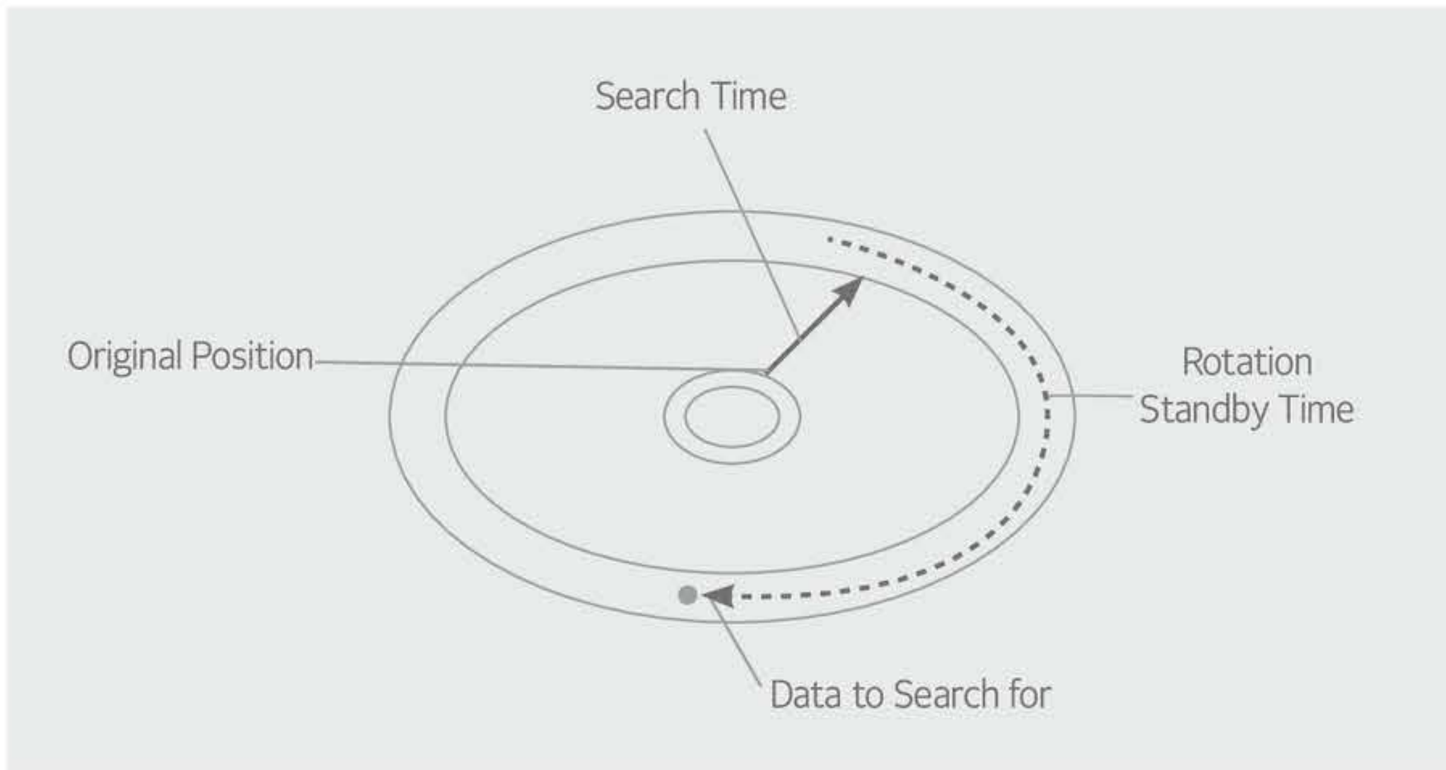
① Disk scheduling

A disk drive that stores data is a device using a rotating magnetic disk. When inputs and outputs are requested to this disk drive, the system performance varies significantly, according to which request is processed first, and the process in moving the head to access the data. Disk scheduling is a technique of efficiently processing I/O requests, when multiple users request them, in order to process different tasks. Using disk scheduling has the following purposes:

- Maximization of I/O requests to service during a unit time
- Maximization throughput per unit time
- Minimization of the mean response time
- Minimization of response time
- Minimization of the variation of response time

② Disk performance measurement indicator

Disk scheduling can be compared with the indicators that measure disk performance. Disk performance measurement indicators include the access time, seeking time, rotational delay or rotational latency, and data transfer time.



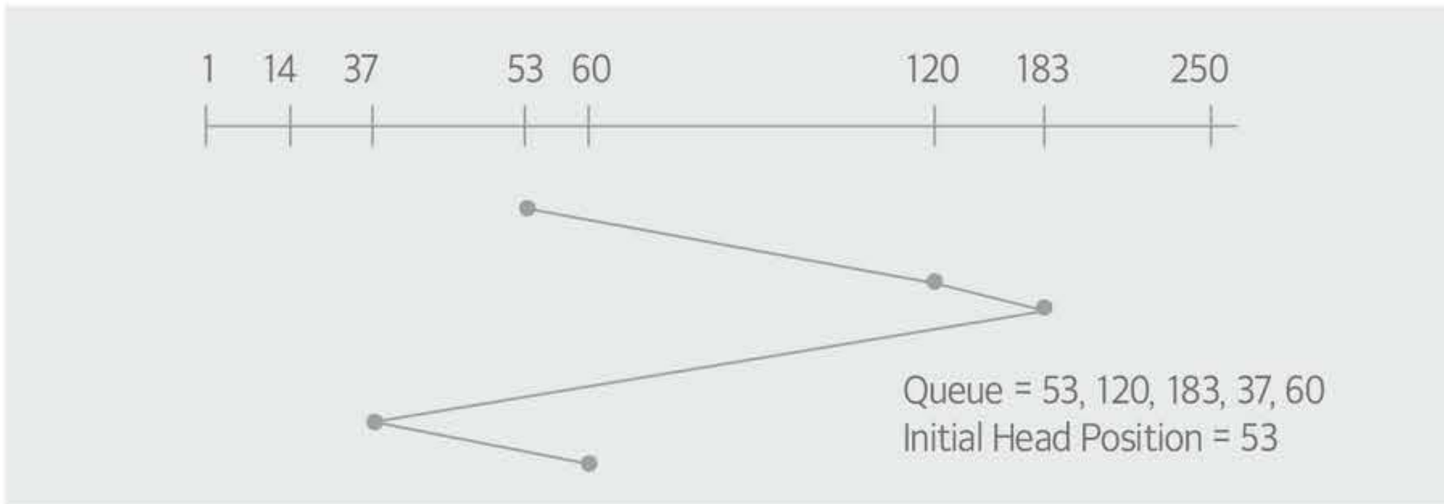
[Figure 59] Disk Performance Measurement Indicators

The seeking time indicates how long it takes to move the head from the current head position, to the track containing the data. The rotational latency indicates how long it takes from the moment the head begins rotating to move to the track containing the data, to the moment it reaches the sector that contains the data.
Seeking time is the time it takes to move the head from the current head position, to the track containing the data. Recent disk drives minimize the rotational latency by having an internal buffer, and by storing all the data in the track

within the buffer. The data transfer time indicates how long it takes to transfer the read data to the main memory. The access time is the sum of the seeking time, the rotational latency, and the data transfer time. This section describes techniques to minimize the access time by minimizing the seeking time and the rotational latency.

③ First come first serve (FCFS) disk scheduling

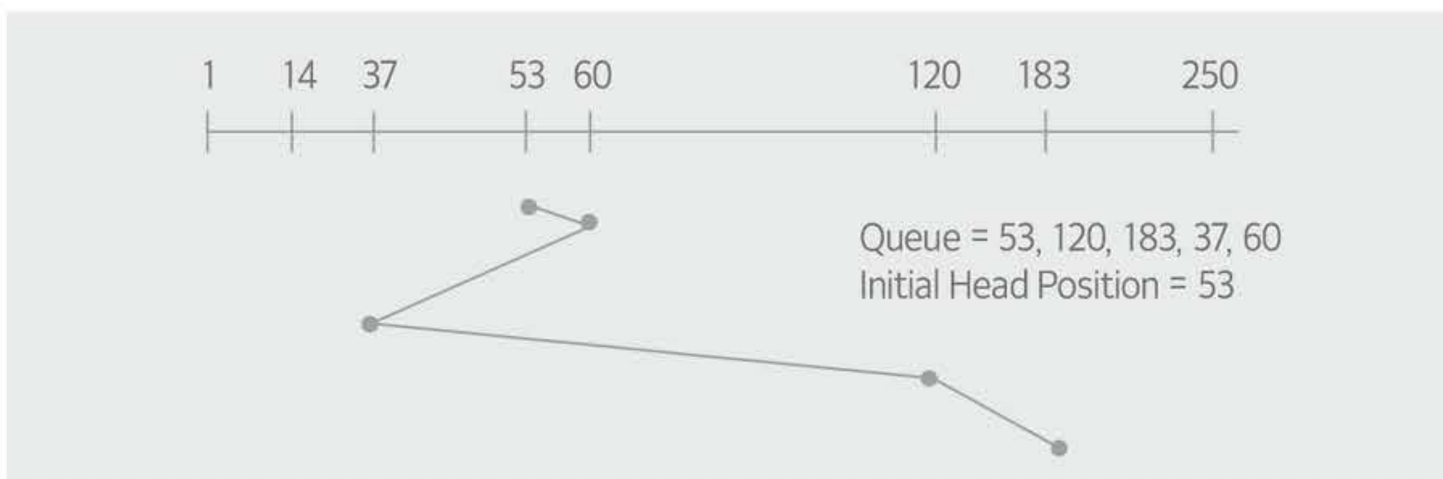
FCFS services the requests in the order they are received. The head position moves in the order of the requested tracks in the disk standby queue. It is easy to program and provides fair service to all requests. However, the head movement may be inefficient, and it can unnecessarily increase the response time. The execution sequence does not change, even when it receives a priority request.



[Figure 60] FCFS Disk Scheduling

④ Shortest seeking time first (SSRF) disk scheduling

The scheduling technique first services the request that is closest to the current head position, among the requested services waiting in the queue. It minimizes head movement because it takes a lot of time to move the head in the disk drive. It maximizes the service response rate per unit time by minimizing the head movement in order to process the queue requests. The response time variation can increase depending on where the requested input or output data is on the disk. It can cause starvation for the requests seeking data that are positioned in the inner or outer cylinder. It offers a higher processing rate and a shorter mean response time than FCFS.

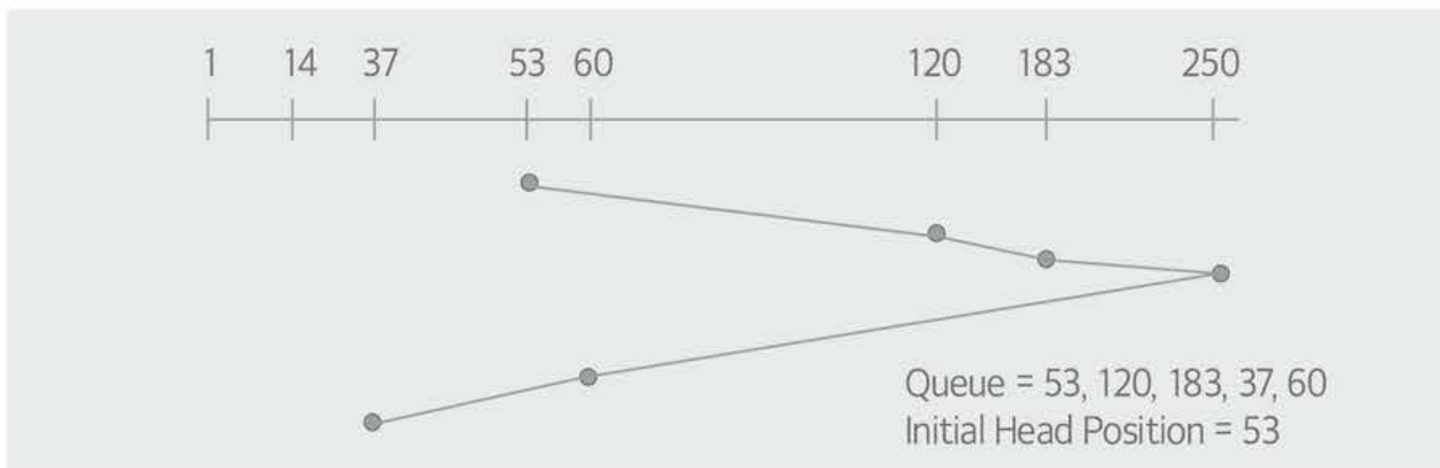


[Figure 61] SSTF Disk Scheduling

⑤ SCAN disk scheduling

The scheduling technique first services the request that has the shortest seeking distance from the current direction

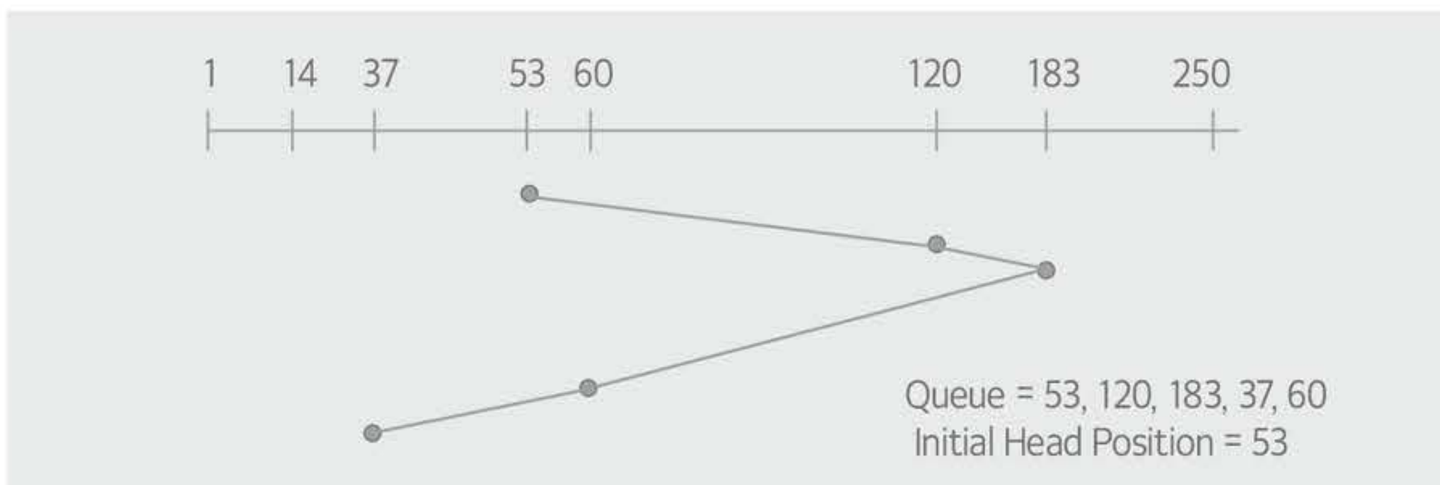
of the moving head. It can reduce the response time variation of SSTF, and it generally offers excellent throughput per unit time and mean response time. On determining the heading direction, the head services all the requests waiting in the request queue in the same direction as the order of the shortest search distance, and it changes the seeking direction after it reaches the outermost or innermost cylinder. It has improved both the throughput per unit time and the response time significantly compared to SSTF, and it solved the starvation.



[Figure 62] SCAN Disk Scheduling

⑥ LOOK disk scheduling

The technique is the same as the SCAN disk scheduling, except that the head changes its direction before reaching the outermost or innermost cylinder. It checks the waiting queue while moving the head, and it does not move further.It changes direction when there are no more requests in the moving direction.



[Figure 63] LOOK Disk Scheduling

⑦ Circular SCAN (C-SCAN) disk scheduling

The SCAN technique moves the head by connecting the inner and outer tracks in a circular model. Like SCAN disk scheduling, the head services the request with the shortest distance in its moving direction first, then it moves to the initial direction to service requests after all the requests in the moving direction are serviced. It services all requests in a predetermined service direction. Since it responds equally to input and output requests by improving the SCAN scheduling, the response time variation is very small, making it easy to predict the response time.

⑧ Circular LOOK (C-LOOK) disk scheduling

Like C-SCAN, it is a LOOK scheduling technique that connects the inner and outer tracks in an annular model in order to make the head move. The head moves to the initial beginning position and services the next requests, if there are

no more requests in the moving direction, even when the head does not reach the end.

03 High Availability Storage

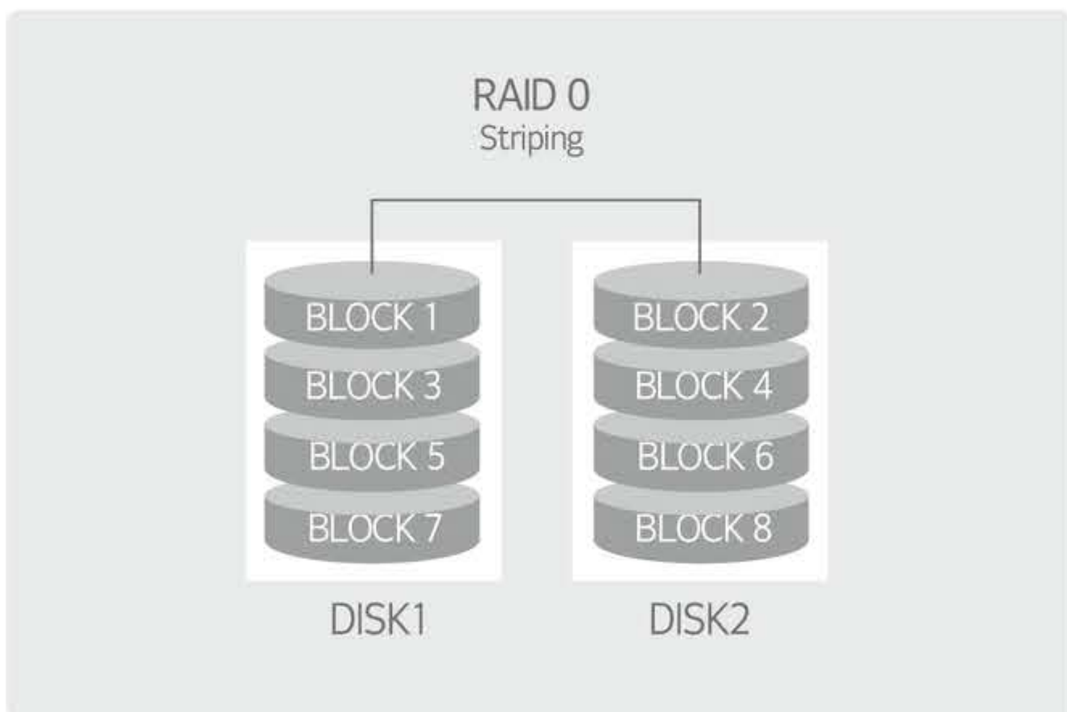
A) Redundant array of independent disks (RAID) technology

Large-capacity storage systems generally have an error controller and backup function to safely store the massive volume of data. RAID technology is a storage technology that minimizes the factors that can cause failure, and it improves access performance by arranging a number of disks, and by creating a separate disk unit by linking them with each other

The main features of RAID are improved availability, increased capacity, and increased speed. Firstly, the improved availability feature provides a hot-swap function to replace a failed disk without shutting down the system, and it recovers the original data to the replaced disk online. Secondly, the increased capacity feature organizes several disks into a large virtual disk, and it recognizes them as large-capacity storage disks. Thirdly, the increased speed improves the overall data transfer rate by partitioning reading and writing data, and by transferring it to multiple disks in parallel. RAID implementation includes RAID-0, 1, 2, 3, 4, 5, 6, and 10. RAID-0, RAID-1, RAID-5, RAID-6, and RAID-10 (1+0 or 0+1) are widely used technologies, while RAID-2, RAID-3, and RAID-4 are rarely used.

① RAID-0 (Striped disk array without fault tolerance)

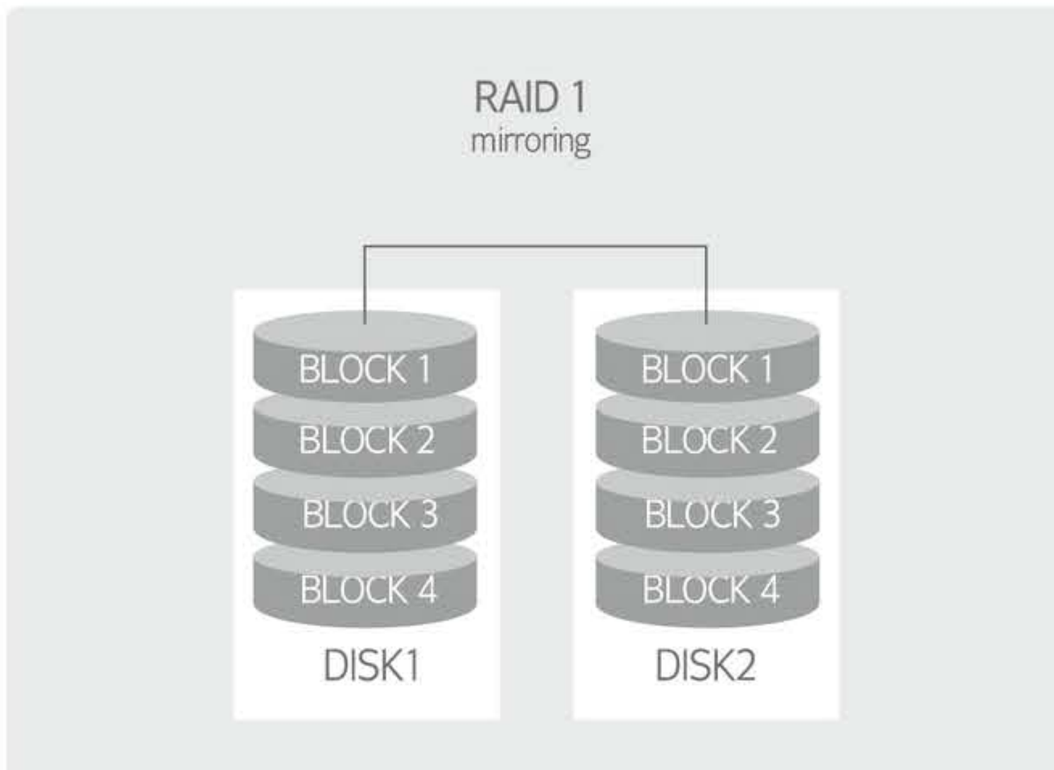
RAID-0 consists of two or more drives and uses disk striping, which stores data by dividing it into pieces of a specific size and saves it on multiple disks at once. Although it increases the I/O speed, it cannot protect data because there is no redundancy. A stripe, which means the size of the stored data, is a unit for dividing data into pieces of a specific size. RAID-0 can reduce the I/O time by enabling parallel processing of I/O requests. However, data is lost if any drive fails. Although the technology is not used much in business environments, RAID-10 uses it by taking its advantages and overcoming its disadvantages.



[Figure 64] RAID-0

② RAID-1 (Mirroring and Duplexing)

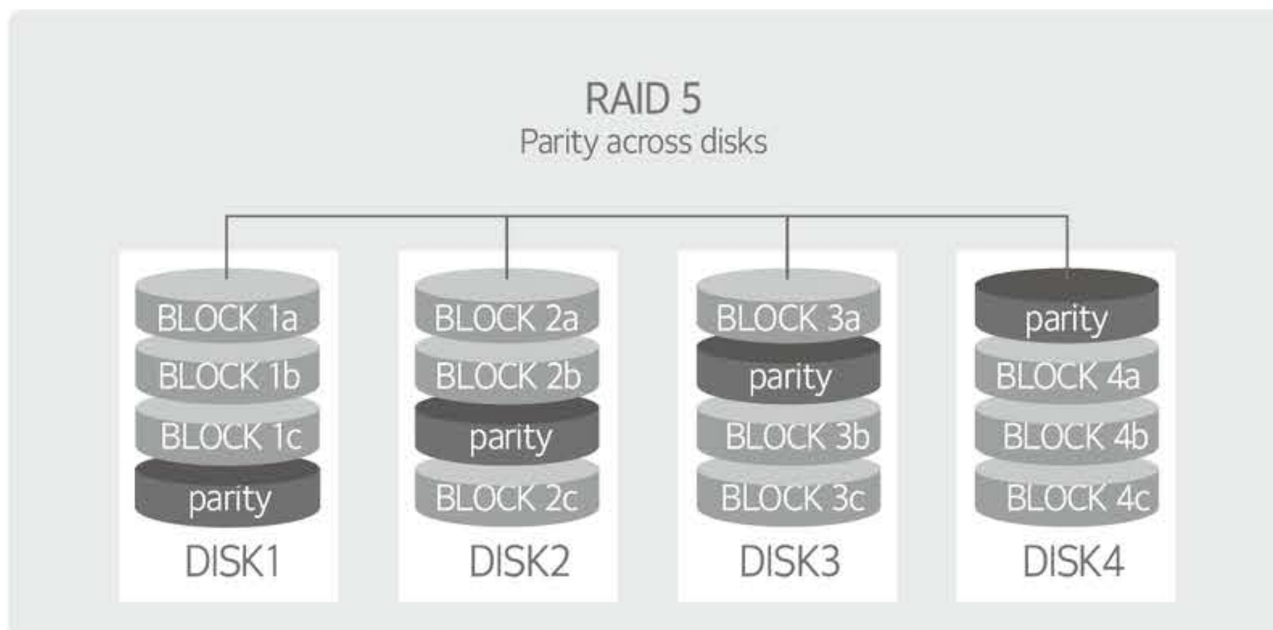
RAID-1 uses a mirroring technique that redundantly stores data on two drives. Since data is stored in redundancy, data can be restored, even if a drive fails. It can improve the reading performance, but the writing performance is somewhat low, since it performs reading operations on a drive with low I/O. Although the system is highly stable, the cost is high because only half of the physical capacity can be used, as the other half is used for redundant storage.



[Figure 65] RAID-1

③ RAID-5

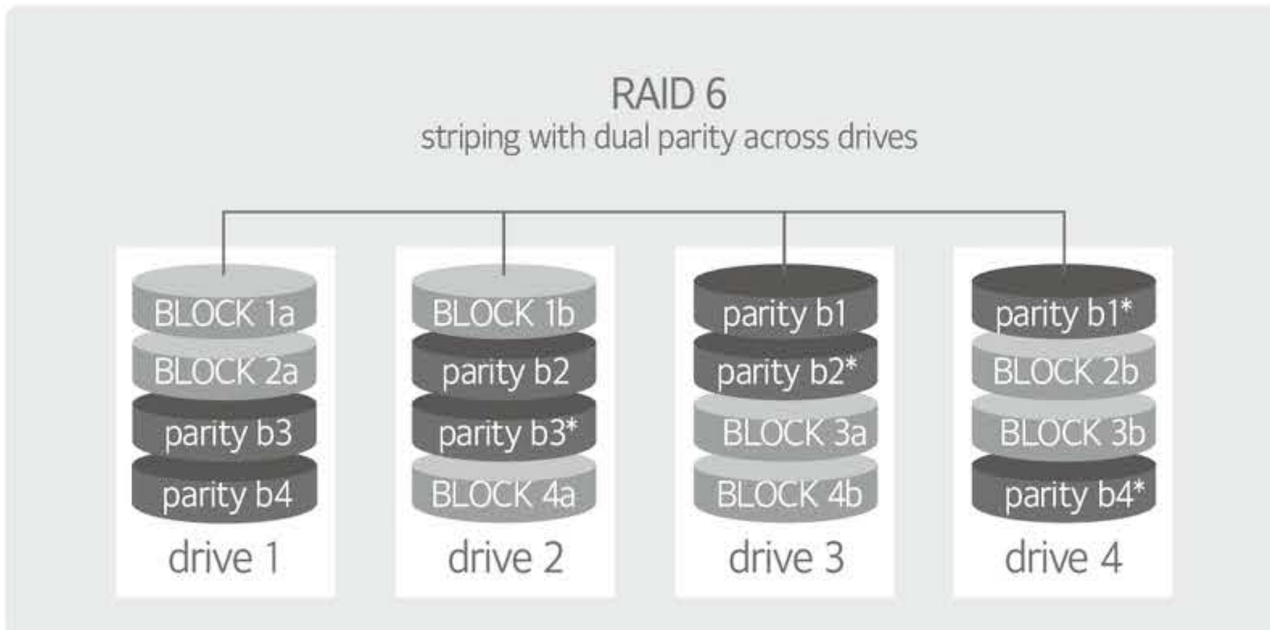
RAID-4 has a separate parity drive and collects and stores parities for data verification and recovery. RAID-5 is an improvement of RAID-4 by distributing the load of the drive that stores the parities. At least three drives are required, and the data and parity are stored by distributing them to different drives. It is highly effective in cost and performance, and recently, most SSDs are configured in RAID-5. If three drives are configured, data cannot be restored if two or more drives fail.



[Figure 66] RAID-5

④ RAID-6 (Stripe set with dual distributed parity)

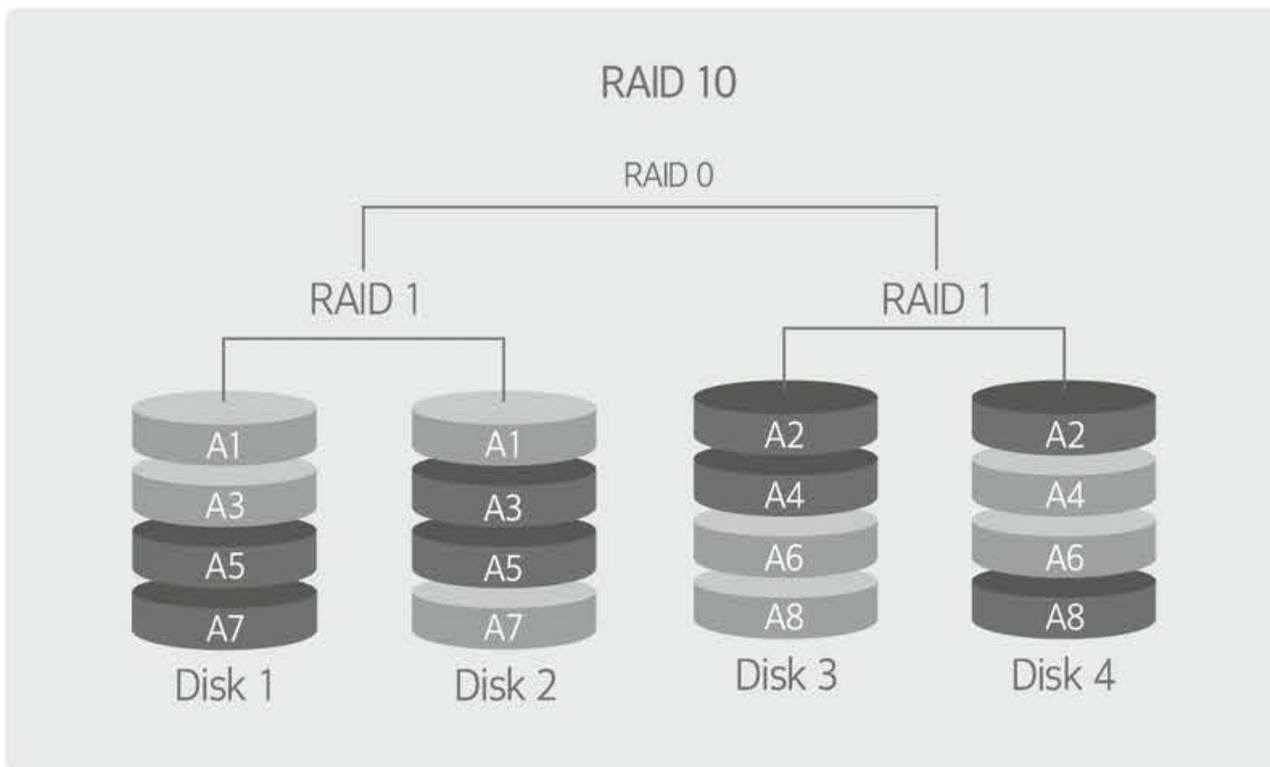
RAID-6 is similar to RAID-5, except that while RAID-5 stores one parity, RAID-6 redundantly stores a parity in two drives. The configuration is more durable than RAID-5 and can store data safely. In a RAID-5 configuration, the rebuild time restores data to the new drive when a drive fails and is replaced. The rebuilding takes a long time in a storage system with a large storage capacity. If another drive fails during this rebuilding time, there is a possibility of data loss. Therefore, RAID-6 is recommended.



[Figure 67] RAID-6

⑤ RAID-10(Striping & Mirroring)

RAID-10 requires at least four drives and is a combination of RAID-0 and RAID-1 to improve I/O speed while providing data stability. It eliminates the concern with stability, which is an important drawback of RAID-0, and it can significantly improve performance, which is the biggest drawback of RAID-1. However, its disadvantage is that the cost is high. RAID 1+0 configures two drives in mirroring first, then RAID-1 drives, configured of two drives, are configured in RAID-0. RAID 0+1 divides the drives to be configured in half, configures them in RAID-0, and reconfigures ½ of the drives and the remaining 1/2 of the drives in RAID-1. The RAID-10 configuration features better stability and generally uses RAID-10 more than RAID-0+1.



[Figure 68] RAID-10

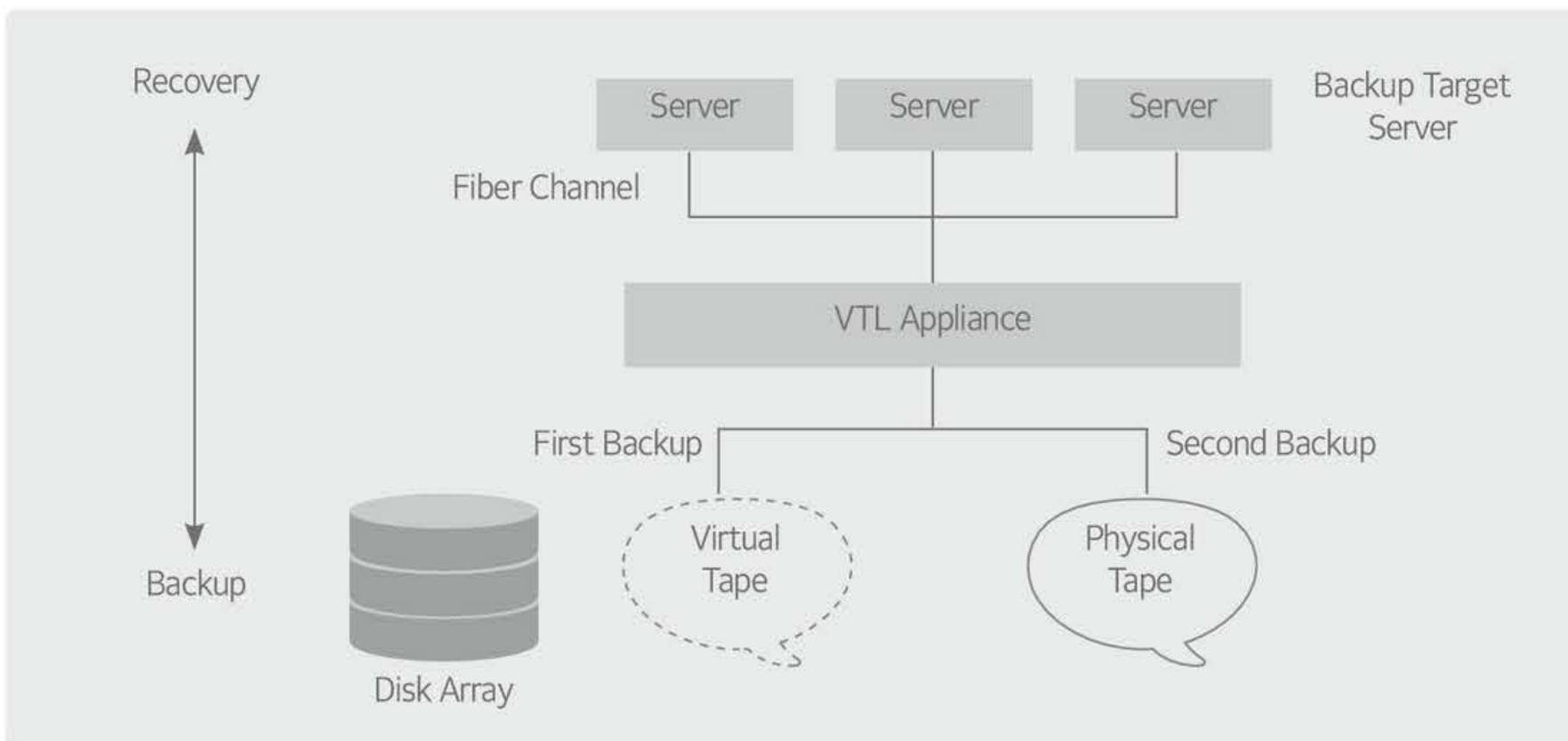
B) Backup storage: LTO and VTL

① Linear tape-open (LTO)

LTO is a standard open tape drive technology that supports high-speed data processing and a large capacity. Although LTO refers to physical devices and technology, they are also used interchangeably. It is also called the tape backup machine, tape media machine, LTO machine, LTO library, and LTO. LTO-1, first released in 2000, provided a maximum speed of 20 MB/s at an uncompressed capacity of 100 GB. The LTO-8 that is currently available, supports the performance of a maximum speed of 427MB/s based on an uncompressed capacity of 12.8TB. The LTO-10 is expected to provide a maximum speed of 1100MB/s at an uncompressed capacity of 48TB.

② Virtual tape library (VTL)

VTL is a backup solution that emulates disk storage and makes it into a virtual tape device to compensate for problems such as limited performance, scalability, and the recovery time of existing tape backup devices. Backup is simplified, and high-speed backup is enabled through tape virtualization.



[Figure 69] VTL Conceptual Diagram

04 Graphic Compression Technology

A) Graphic compression type

Video data compression, which accounts for most of the traffic in a multimedia network, can be divided into lossless compression (reversible compression) and lossy compression (irreversible compression). Lossless compression is also called reversible compression.It refers to a method of restoring a compressed image without information loss from the original data while decompressing. It is characterized by a lower compression rate than lossy compression.

Lossy compression is also called irreversible compression.It refers to a compression method when the compressed data is restored, but it does not match the original data before the compression because some data is lost.

① Lossless compression

Since the compression and decompression algorithms are exactly the opposite of each other, the compression method preserves the original data's integrity, and no part of the data is lost during processing. Leading lossless compression methods include run-length decoding, dictionary decoding, Huffman coding, dictionary coding, and arithmetic coding.

〈Table 15〉 Lossless Compression

Compression method	Description
Run-length encoding (RLE)	• Run-length encoding (RLE) • A method expressing consecutively repeated symbols in terms of the symbol and repetition count (run-length)
Dictionary coding	• Dictionary coding • A method that creates a dictionary while scanning a message, and if an entry character string in the dictionary is found in the message, the code value (index value) of the entry is sent instead of the string
Huffman coding	• Huffman coding • A method that encodes data in a binary pattern and allocates short codes to symbols that occur more frequently and long codes to symbols that occur less frequently
Arithmetic coding	• Arithmetic coding • A method that maps the entire message to a small section in [0,1] and encodes the small section in a binary pattern

② Lossy compression

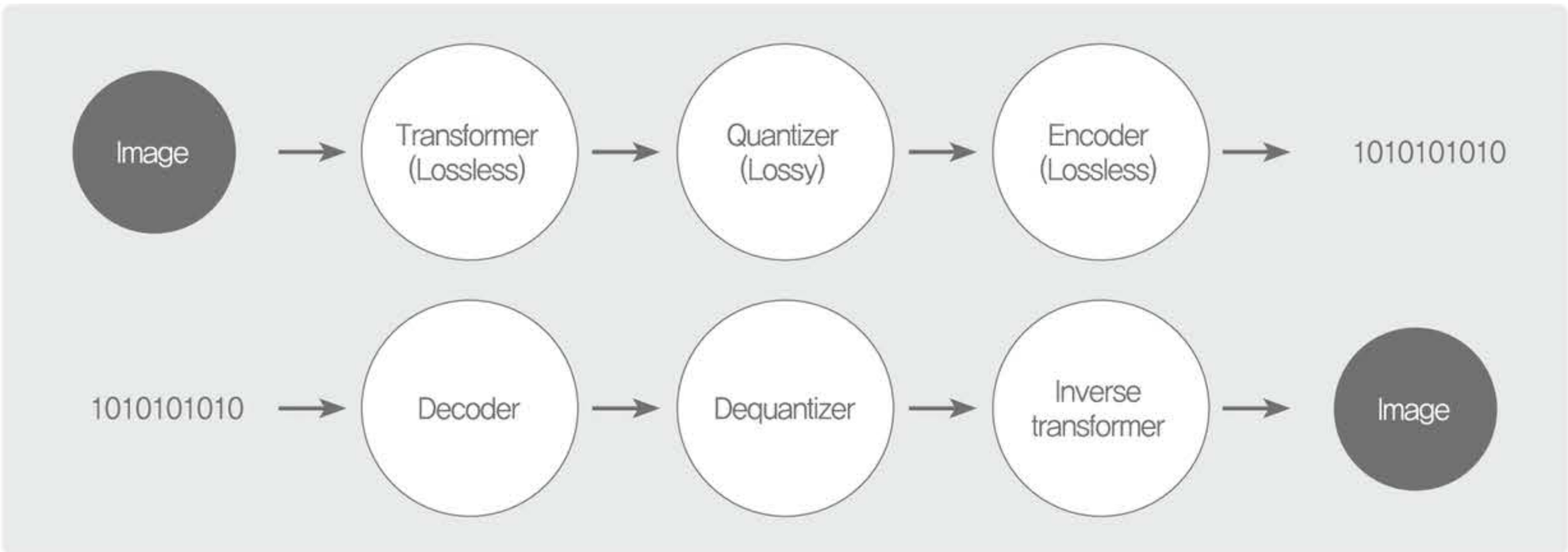
The lossy compression method compromises some accuracy to increase the compression rate, by allowing the loss of redundant or unnecessary data. There are two types of lossy compression methods: prediction coding and transform coding. The prediction coding method is used for digitizing the analog signal. Instead of separately quantizing the PCM (Pulse Code Modulation) samples, it quantizes the difference. The difference values are smaller than the actual sample values, so fewer bits are required. The transform coding method transforms a signal from one domain (mainly a time and space domain) to another domain (mainly a frequency domain), then compresses it.

B) Multimedia data

Multimedia data includes text, image, video, and audio data. The text has the form of plain text and non-linear hypertext. The basic language is Unicode for expressing symbols, and it uses a lossless compression method.

In multimedia, an image is called a still image and refers to a photo, fax page, or a video frame. As shown in [Figure 70], an image is transmitted as binary data through a transformation process, a quantization process, and an encoding process, before being converted back into an image through the reverse process.

In the transformation process, the JPEG uses DCT (Discrete Cosine Transform) in the first stage of compression, and the decompression uses the inverse DCT method. The transformation and inverse transformation apply 8 X 8 blocks. The quantization process creates integers from the real number of the DCT transform output and converts some values to zero. The coding process arranges data in a zigzag order after quantization and before encoder input, then lossless compression is performed using run-length decoding and arithmetic coding [1].



[Figure 70] Image Data Transformation Process

Video data is composed of several frames. Since each frame is an image, a video file requires a high transmission rate. Video data can be compressed with a spatial compression or a temporal compression method. JPEG performs spatial compression of each frame, and each frame is compressed independently. Temporal compression removes redundant frames and configures each frame into I-frames, which are independent frames, B-frames, and P-frames, which reference other I-frames to compress data temporally.

Analog audio signals are digitized using an analog-to-digital converter (ADC), and the ADC consists of two processes: sampling and quantization.

① Video compression standard

The Moving Picture Experts Group (MPEG) is an international standardization organization. The official name of the standard is ISO/IEC JTC1/SC29/WG11. MPEG created the following compression formats and additional standards.

<Table 16> Compression Format and Additional Standard

Type	Key Features	Application
MPEG-1	• Audio and video compression and decompression	• MP3 and video CD
MPEG-2	• Compression and decompression for DTV broadcasting	• Digital TV and DVD
MPEG-4	• Mobile phone video compression and decompression	• IMT2000 and Internet broadcasting
AVC/H.264	• Twice the compression rate, compared to MPEG-2	• HDTV and mobile phone video
MPEG-V	• VR media presentation and control	• 4D movies and VR
MPEG-DASH	• Internet-based video streaming	• Internet image streaming
MMT	• Next-generation media transfer standard to replace the MPEG-2 TS transfer standard	• UHDTV and smart TV
HEVC/H.265	• Twice the compression rate, compared to AVC	• UHDTV and smart TV
3D Audio	• Added dimensions to existing surround audio	• UHDTV and smart TV

② Advanced video coding (AVC) and high efficiency video coding (HVEC)

AVC codec is a video compression technology jointly proposed by the international standard organizations ITU-T and ISO. It uses the name H.264, proposed by ITU-T, or MPEG4 Part10/AVC, proposed by ISO. The standard name for HEVC is ISO/IEC 23008-2 by ISO/IEC

VI. Embedded System

Recent trends and issues

With the advent of the IoT (Internet of Things), represented by technological innovation in various industries, the embedded system market is expected to grow explosively, due to the increased demand in the automobile (or connected car) and medical fields.

A market research firm predicts that the global embedded systems market will grow by 5.6% per year, reaching $2,587 billion by 2023. The reason for this growth is that the demand for embedded systems in the automotive and medical fields is expected to increase rapidly, thanks to the recently emerged IoT.

Embedded systems are widely utilized in information home appliances, information terminals, communication equipment, aviation/military, logistics/finance, vehicle/transportation, office work, industrial/control, medical care, games, and robots. In other words, embedded systems have become an integral part of any industry. Among various applications, the automotive and medical industries are expected to drive the growth of the embedded system market in the future.

However, solving security problems that may arise from linking various embedded systems, through technological innovation, has emerged as a challenge. Embedded systems mostly operate home appliances, transportation, aviation, machinery, medical equipment, automobiles, and communication devices. Recently, all equipment, machines, and vehicles are being connected to an OS, a webserver, a remote control, and smart devices. Among these, medical equipment and automobiles are very important because their security vulnerabilities can have a great impact on public safety.

It is essential to protect embedded systems from unauthorized software and other vulnerabilities, such as theft and privacy protection. Accordingly, continuous efforts are being made to protect the systems from theft and data hackers and to solve them.

Learning objectives

1. To be able to explain system structure and characteristics.
2. To be able to explain embedded OS.

▶▶▶ Keywords

Embedded hardware, Microprocessor, Microcontroller, Low power, Lightweight, Small form factor, RTOS, Embedded Linux Windows CE

✚ Preview for practical business

Company A specializes in developing embedded systems that control industrial equipment and home appliances. Since embedded systems are applicable to a wide range of devices, such as sensors, controllers, and IoT devices, they are the core technology required from the market development stage. An embedded system is a solution built into a product to perform specific tasks. Developers of such systems must have a broad knowledge of software and hardware functions. This chapter describes embedded systems in detail.

01 Overview of Embedded System

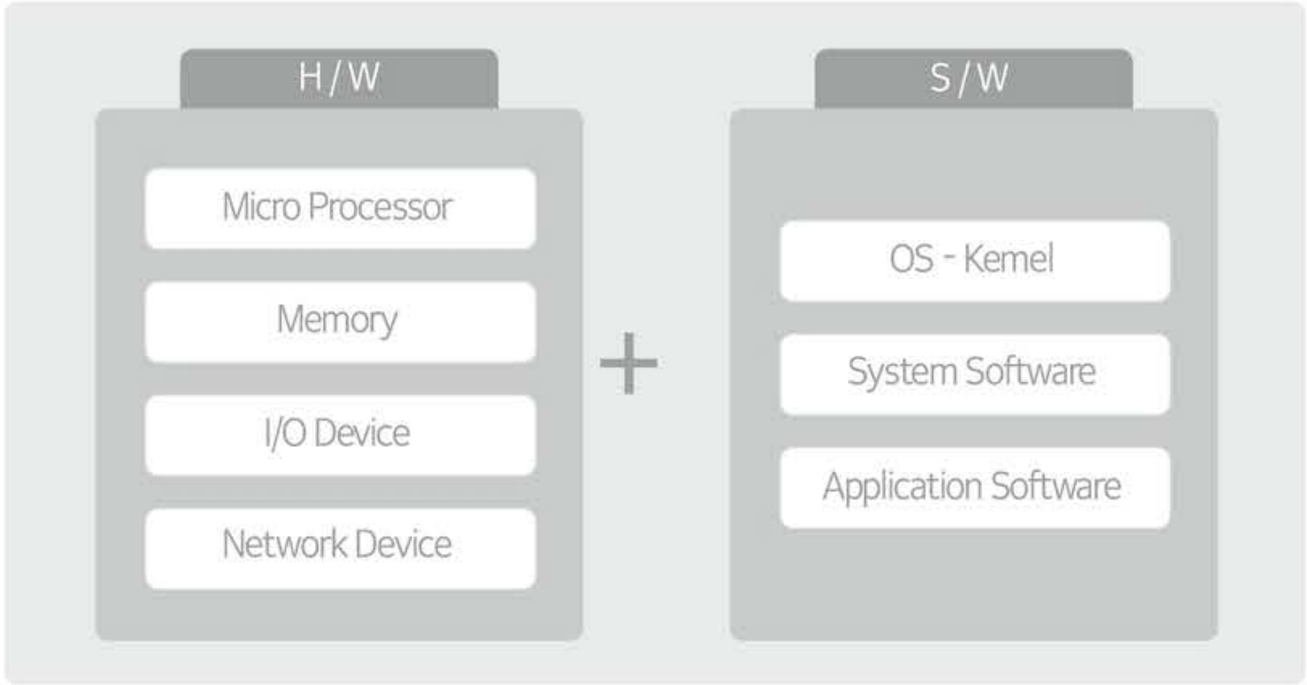
A) Definition of embedded system

An embedded system is a computer system that performs a specific function for controlling a machine or other systems requiring control, and an electronic system in the device.

In more detail, it is a system with a built-in microprocessor and a program that drives the microprocessor to perform a specific function, more than just simple circuits in various electronic devices, home appliances, and control devices used in our daily life.

An embedded system is configured of a set of hardware and software, as shown in [Figure 71].

- Embedded hardware: Embedded hardware consists of a microprocessor/microcontroller, equivalent to a CPU, memory, input/output device, network device, sensor, driver, etc.
- Embedded software: Embedded software consists of an operating system (OS) that directly controls hardware and manages the software, system software, and application software. Unlike conventional software, the main purpose of the embedded software is to provide an appropriate response to the physical world. Therefore, embedded software has the characteristics of the physical world, such as time, energy, and permanence.



[Figure 71] Embedded System Configuration

Embedded systems control many devices used in everyday life. As shown in <Table 17>, embedded systems are used in a wide range of areas, from portable digital watches and health bands, to large fixed facilities, such as traffic lights, factory control devices, and nuclear power plant control systems.

〈Table 17〉 Examples of Embedded System

Type	Example
Offices	Telephone, fax, copying machine, scanner, and cutter
Games	Arcade and console game machines
Industry/control	Automation systems, industrial accident diagnosis systems, and fire systems
Vehicles/railways/transportation	Engine controls, navigators, ITS controls, and subway electrical controls
Logistics/finance	POS, ATMs, currency exchange indicators, and card readers
Medicine	Diabetes meters, ECG meters, and MRI systems
Wearable	Smartwatches and health bands
Digital advertising	Signage systems and digital information displays (DID[4])
Aviation/shipbuilding	Control systems, signal systems, and information display systems
Robots/drones	Industrial controls, education, and aviation filming drones

B) Characteristics of embedded system

Since an embedded system performs a specific mission, engineers can optimize the design in order to reduce the size and production cost and to increase reliability and performance.

① Low cost

Many embedded systems are mass-produced in units of millions and the reduction of production costs is one of the crucial factors. Some embedded systems that do not require high processing performance or resources can have a relatively slow processor and small memory in order to reduce costs.

② Low performance

Many embedded system parts have low performance. Here, being slow does not simply refer to a slow clock speed. In

4 DID: Digital Information Display

general, an embedded system's overall structure is intentionally simplified to reduce cost, compared to the hardware of general-purpose computer systems. For example, embedded systems often use peripherals controlled by a serial bus that is 1024 times slower than the conventional peripheral interfaces that are used in personal computers.

③ Resource constraints

Programs on embedded systems usually operate with real-time constraints on limited hardware resources. There is often no disk drive, operating system, keyboard, or screen on the system. It may not have a file system, or it may use a flash drive as a storage medium. Even if there is a user interface, it could be a small keypad or LCD.

④ Stability

Embedded systems are designed to run reliably, without error, for many years. Therefore, firmware goes through a more careful development and testing process than the software used in personal computers. Most embedded systems avoid the use of parts damaged by mechanical actions, such as disk drives, switches, and buttons. Instead, they use chip materials that are relatively free from physical damage, such as flash memory.

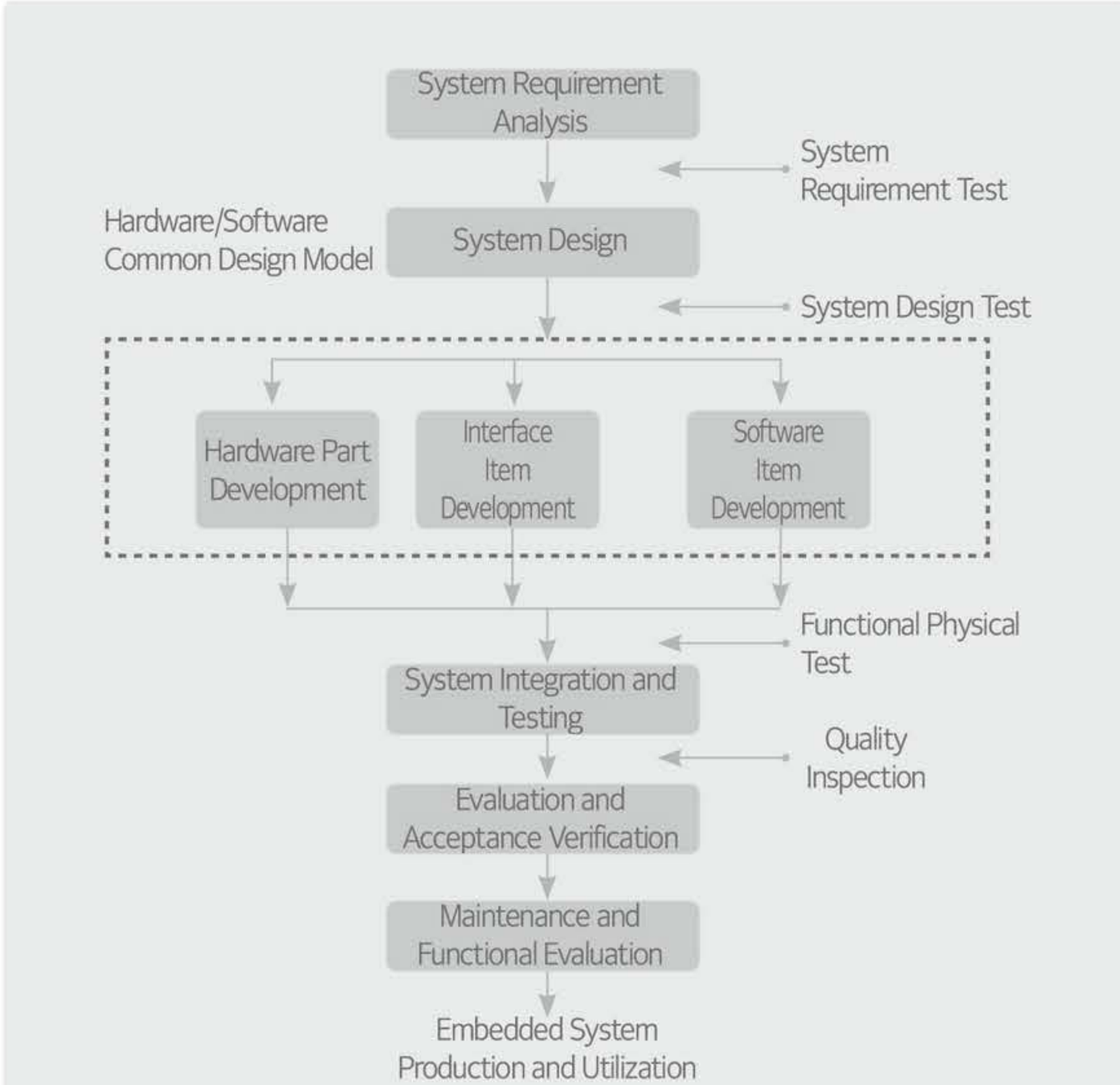
⑤ Resilience

Embedded systems may be applied in places where it is difficult for humans to immediately and directly control, such as oil boreholes and outer space. Therefore, an embedded system must be able to restart itself, even in the worst-case scenario. An electronic part, called a watchdog timer,[5] enables such an emergency recovery.

C) Embedded system development process

An embedded system is developed by the procedure, shown in [Figure 72]. Unlike conventional information system development, it includes a hardware development process.

5 Watchdog Timer (WDT): An electronic timer used to detect and recover computer failure. Computers in normal operation restart the watchdog timer periodically to prevent a timeout.



[Figure 72] Embedded System Development Procedure

<Table 18> below shows details of the above development procedure.

〈Table 18〉 Detailed Tasks of Embedded System Development

Stage	Task	Products
Product planning	- Commercial feasibility determination (Price, technology, and marketing review) → Determining rough product specifications	Detailed specification and timing chart
Part selection	- Hardware team: Selects hardware parts (CPU, memory, and other peripherals) → Selecting hardware parts that meet requirements at the lowest cost - Software team: Selects the OS. Checks if the product under development requires an OS and selects one, according to characteristics.	Hardware and OS specifications
Hardware circuit generation and software structure design	- Hardware team: Generates the circuit diagram (Disposal of all produced products if there is a hardware error → Led by skilled personnel). - Software team: Designs module-to-module interfaces, and checks required libraries.	Hardware circuit diagram
Code design for each artwork module	- Hardware team: Performs the artwork using the circuit diagram. - Artwork: The task of generating data to connect actual parts on the circuit diagram (Actual part size, connector shape, part location selection, and wiring) - Software team: Designs the code of each module, while considering the interface.	Codes artwork results.

Making the PWB (printed wiring board) and integrating modules	- Hardware team: Manufactures the PWB using the artwork. - PWB: A circuit board is made by melting other parts with chemicals and the necessary parts using a copper disk. A machine uses artwork data to draw the circuit on the board and make the PWB. - Software team: Integrates modules. (For example, embedded Linux software integrates the bootloader, Linux kernel, and apps.)	PWB
Part mounting and hardware testing	- Hardware team: Mounts parts like the CPU and memory on the PWB. - Software team: Tests if the actual hardware operates as intended. (Since it is not realistic to test all designed and completed codes, a test program is made to check the normal operation of hardware parts on the PCB, each CPU port, the memory unit, and I/O devices.)	Test results
Integrated testing and debugging	- Installation of software on hardware (Hardware power supply → Bootloader execution→ Bootloader OS execution → OS performs intended action, according to the system input value.)	
Mass-production testing	- Solving the identified problem until the mass production - Establishing countermeasures to unexpected problems, such as the increasing part defect rate and part termination	
Quality inspection	- Conducted after functional stabilization (six months before the mass-production testing) - Conducted in various environments (high temperature/high humidity and sub-zero environments) for about six months → For worldwide export	
Specification inspections, such as EMI and safety	- EMI: Inspection of the interference of the electromagnetic wave on other electronic parts (Change of channel when using a mobile phone) - Safety: Product stability test - The designer includes the EMI Team data in the debugging circuit during the integrated testing and debugging for EMI debugging.	EMI and safety results
Product shipment	- Finished product shipment	

02 Embedded Hardware

A) Microprocessor, the key to embedded systems

An embedded system can handle more functions and performance because the processor provides the performance required to drive the device. Early microprocessor units (MPUs) were made in a single chip, in which registers data processing, arithmetic logic unit (ALU) for mathematical operation, and flags for storing result values were integrated into a chip. Although CPU and MPU were separated according to use and purpose in the past, the boundaries now are somewhat ambiguous. Since then, microcontrollers, with a built-in processor, memory, and control interface on a single chip, have emerged to be used for real-time operating systems or automatic controls.

It is not far-fetched to consider that the MPU is included in all the electronic devices that are open to user intervention in its use or configuration. It is currently used in smartphones, smart bands, and tablet PCs that are used

in daily life. It is also used in mechanical keyboards that provide functions, such as various shortcut keys and infinite key inputs. It is also in a TV or air conditioner,as well as in the remote control that controls the TV and air conditioner. Moreover, it is in game consoles, car trip computers, and smart keys.

While Intel and AMD are everything when it comes to CPU for computers, many companies make MPUs. Many companies, like Broadcom, Mediatek, Samsung Electronics, Apple, Nvidia, Qualcomm, and Texas Instruments, make MPUs used in electronic devices.

As shown in <Table 19>, various CPU architectures, from 8-bit to 64-bit, such as ARM, MIPS, ColdFire/68K, PowerPC, x86, PIC microcontroller, and 8051, are used for embedded systems. That is what differentiates the embedded system market from the personal computer market.

Moreover, one of the latest trends of embedded systems is the System on Chip (SoC) concept that integrates all ICs for a specific field into a single CPU chip in order to eliminate the need to mount separate peripheral chips on the board.

〈Table 19〉 Embedded Processor Types

Processor	Manufacturer	Type
8051 train	Atmel, Cirrus, Intel, TI, etc.	8051
PowerPC series	Freescale	MPC860,850,8260
ARM series	ARM	ARM7, ARM9, ARM10, ARM11, and Cortex
Intel series	Intel	strongARM, Xscale (PXA255), and Bulverde (PXA270)
Samsung series	Samsung Korea	S3C2410, S3C2440, etc.
MIPS series	MIPS	MIPS32, MIPS63, etc.

B) Technical trends of embedded hardware

① Expanded connectivity

Studies of applying connectivity technologies, such as WiFi, LTE, BLE, and ZigBee, used by mobile devices and servers/PCs, to embedded systems, have increased with the convergence of the IT industry.Traditional industries and interoperability between devices, such as IoT and M2M sensor networks, have also become important.

② Low-power 32-bit processor

Processors for embedded systems were mostly converted from 8 bits to 32 bits. As such, the number of embedded systems using an OS has rapidly increased, and the portion of general-purpose OS, like Linux, has significantlyincreased.

③ Polarization of lightweight/general-purpose OS

Embedded operating systems, supporting wearables and IoT devices, require lighter weight and lower power consumption, while operating systems applied to conventional embedded devices include middleware, related to high functionality, and stability and convenience functions like 3D acceleration GUI and multimedia. As a result, technology polarization is accelerating.

03 Embedded Software

A) Definition of embedded software and technology application

Embedded software refers to software installed in embedded systems. As the application fields of embedded systems diversify, embedded software is becoming increasingly complex to handle various functions.

As shown in <Table 20>, the main technology fields of embedded software can be classified into basic embedded applications, embedded middleware, embedded system software, embedded system development tools, and embedded software platforms.

〈Table 20〉 Technology Fields and Examples of Embedded Software

Technology Application	Example
Basic embedded application	Multimedia player, MAP viewer, browser, PIMS, game, mobile shop, and CNS (GPS, GIS)
Embedded middleware	- Distributed middleware, such as CORBA, COM XML, and TMO - Java middleware, such as JVM and J2ME - Control middleware, such as Jini, UPnP, and Havi - Multimedia middleware, such as streaming and codec - Communication middleware, related to WLAN and WPAN
Embedded system software	Embedded OS, device driver, wired/wireless communication protocol and multimedia protocol support library, and embedded DBMS
Embedded system development tool	Design tool, test tool, IDE, target system resetting tool, simulator, cross compiler, real-time remote monitor, remote monitor, and device driver toolkit
Embedded software platform	MS .net compact framework, Sun ONE, Brew, and WIPI

B) Characteristics of embedded software

Embedded software is embedded in a physical product and continuously interacts with the physical environment. In other words, it must continuously receive and process signals from the physical environment. In some cases, it is necessary to receive the user's requirements and respond to them, by considering changes in the physical environment. There are also cases, such as computers and smartphones, where it only responds to user requirements. Therefore, embedded software has characteristics, such as reactivity, timeliness, concurrency, heterogeneity, liveness, resource and environmental constraints, and determinism. <Table 21> below shows features of each characteristic.

〈Table 21〉 Characteristics of Embedded Software

Technology Application	Example
Reactivity	Embedded software should continuously interact with physical environmental changes, by converting input data into output data at a specific speed.
Timeliness	The time required to convert input data into output data should be within the very short allowable range.
Concurrency	Embedded software should be able to process input data, generated by the concurrent physical environmental changes.
Heterogeneity	Embedded software should effectively process data from hardware or software.

Liveness	Embedded software should never stop its operation, even after unexpected events.
Resource and environmental constraints	Embedded software should meet the resource (CPU processing speed, memory size, and user interface) and environmental (power supply and protocol compatibility) constraints.

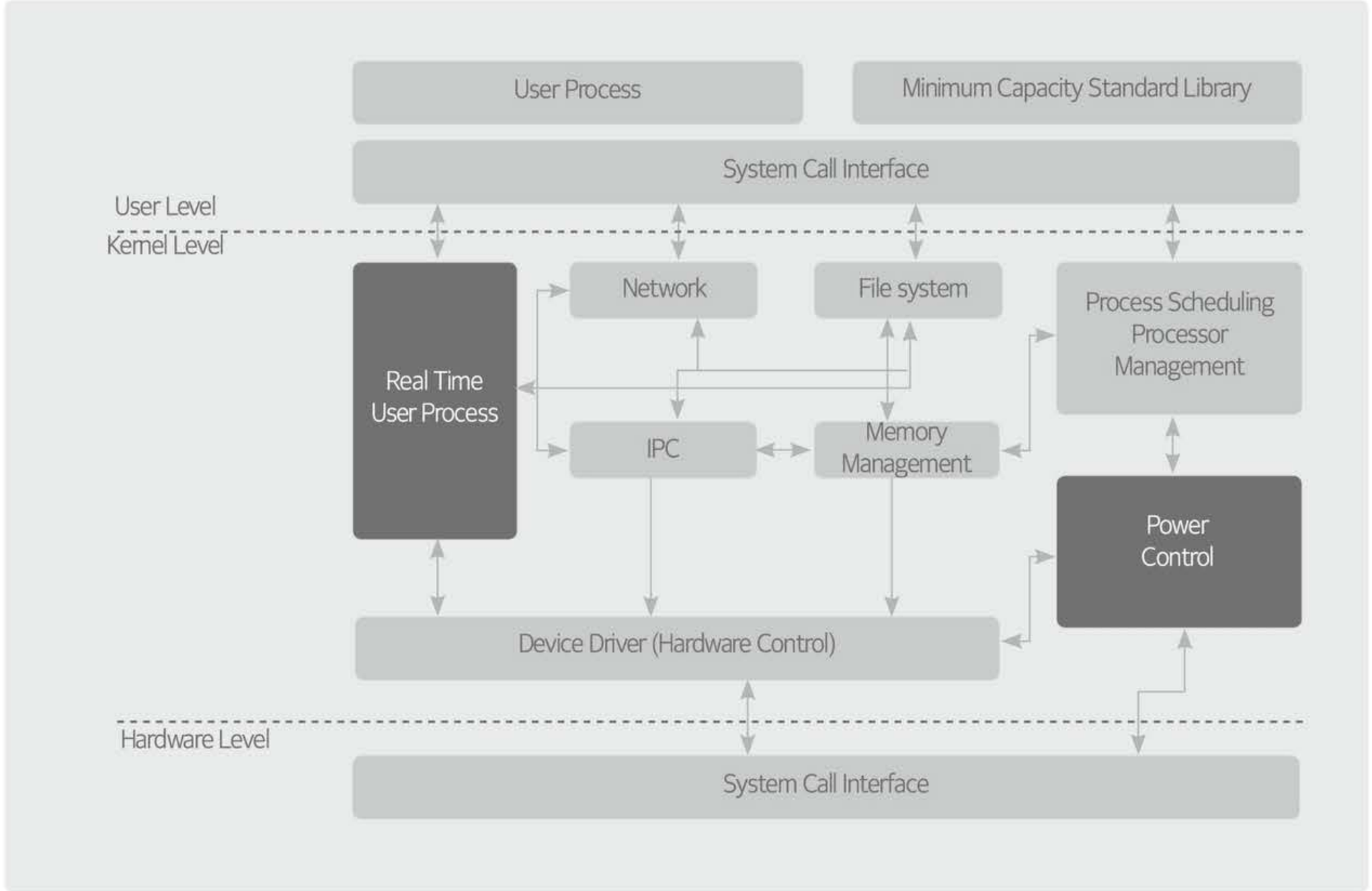
04 Embedded OS

A) Concept of embedded OS

An embedded OS is an operating system that is optimized to execute the application programs running on a computer system that perform specific functions for controlling a machine or other systems. It is optimized for a specific system that has hardware with a built-in microprocessor, and it performs set purposes. In the early embedded systems, the hardware was directly controlled using the assembly language, and it operated without a separate OS. However, as the embedded system applications have diversified, the need to use an OS has increased.

B) Embedded OS structure and applied technology

As shown in [Figure 73], an embedded OS generally has a structure to support the stable application program execution with a low-specification processor, a limited memory and storage device, and poor operating environment conditions.



[Figure 73] Embedded OS Structure

Technologies applied to embedded OS can be characterized by real-time, small/lightweight, and low power consumption, as shown in <Table 22>.

〈Table 22〉 Embedded OS Application

Application	Description
Real-time multitasking technology	Real-time synchronization mechanism and real-time resource management function to improve the guarantee and predictability of real-time services, along with real-time scheduling
Lightweight kernel technology	Basic kernel support technology to provide a solution suitable for a system targeting a specific application with limited resources, according to the characteristics of the embedded system
Power management technology	Power management system for embedded systems, including portable information home appliances using small batteries in a ubiquitous environment
Tiny kernel technology	Basic support technology to support ultra-small/ultra-low-resource embedded systems, such as sensor networks in ubiquitous environments

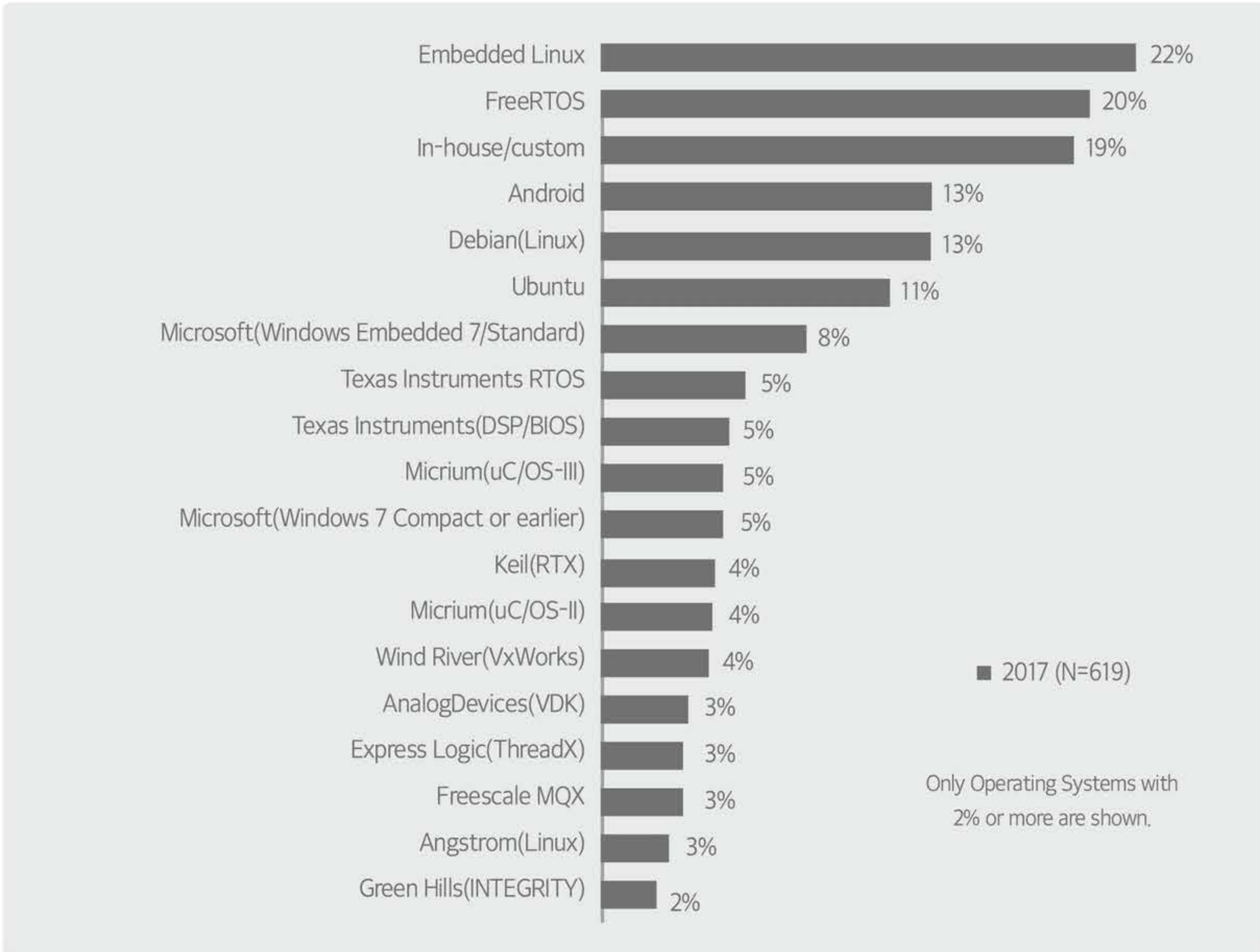
C) Types and trends of embedded OS

Embedded software runs in a wide range of devices and supports many operating systems. However, it is largely divided into real-time operating systems that must satisfy conditions to operate within a set time, and operating systems that do not.

- Real-time OS (RTOS): Mainly focuses on controlling application programs or hardware affected by a limited amount of time. The leading operating systems include VxWorks, VRTX, QNX, and pSOS.
- Non real-time OS: Mainly focuses on development convenience, scalability, and diverse operating environments. The leading operating systems include Linux series, embedded Window XP, and Windows CE.

Traditionally, an embedded OS was used because systems, such as satellites and missile controllers, that must have flawless protection against errors needed to have an OS optimized for real-time systems. In that case, the systems used RTOS, instead of a general-purpose OS.

However, one of the latest trends in embedded OS technology is that the portion of Linux and free RTOS is steadily increasing, when compared to traditional embedded RTOS, such as VxWorks, pSOS, QNX, and VRTX, which have led the existing information home appliance market and other embedded application industries (see [Figure 74]). Linux is installed in about 65% of all embedded devices and is still growing very fast, at an annual average of 17%. Meanwhile, the UK's FreeRTOS is rapidly expanding its base. The market share is around 50%, which makes it the de-facto standard. In addition to FreeRTOS, ARM's mbed, which was recently acquired by Softbank in Japan, is spreading rapidly. Of course, commercial products are still leading in markets that require a high degree of safety, such as in automobiles, aircrafts, and railways.



[Figure 74] Status of Embedded OS Use (2017)

VII. Information System Implementation

▶▶▶ Recent trends and issues

With the advent of the information age, the social demand to construct information system infrastructure is increasing, and various information systems are being implemented for business efficiency and customer service improvement. These information systems generally consist of hardware, software, and human resources. It is necessary to understand the concept, architecture, and structure of the information to implement a successful information system.

Recent information systems tend to adopt a large number of web-based structures. In that regard, theinformation system projects of the planners (public institutions) and implementers (systems integrators and hardware manufacturers) need to procure hardware resources that suit the project scope in order to improve the efficiency of investment in information systems.

Information system infrastructure is generally implemented by purchasing servers, storage, and network devices separately from each vendor, then integrating them for the project. Moreover, they also purchase software separately, and thus need to test and verify if it works properly with the hardware.

The complex structures and procedures of constructing information system infrastructure have inevitably resulted in inefficiency in construction time and system management. The converged infrastructure that integrates the hardware,such as storage, servers, and networks, with the virtualization solution and the management software into a product, and validates it in advance for supply to customers, has been introduced to compensate for it. Moreover, the hyper-converged infrastructure that complements the high investment cost has also appeared. These efforts to increase the efficiency of system implementation are ongoing.

▶▶▶ Learning objectives

1. To be able to explain information system structures.
2. To be able to calculate information system capacity.
3. To be able to explain the latest OS technologies and trends.

▶▶▶ Keywords

Server, storage, network equipment, webserver, WAS server, DB server, hardware capacity calculation, OLTP capacity calculation, web/WAS capacity calculation, memory capacity calculation, disk capacity calculation, storage capacity calculation, converged infrastructure, hyper-converged infrastructure

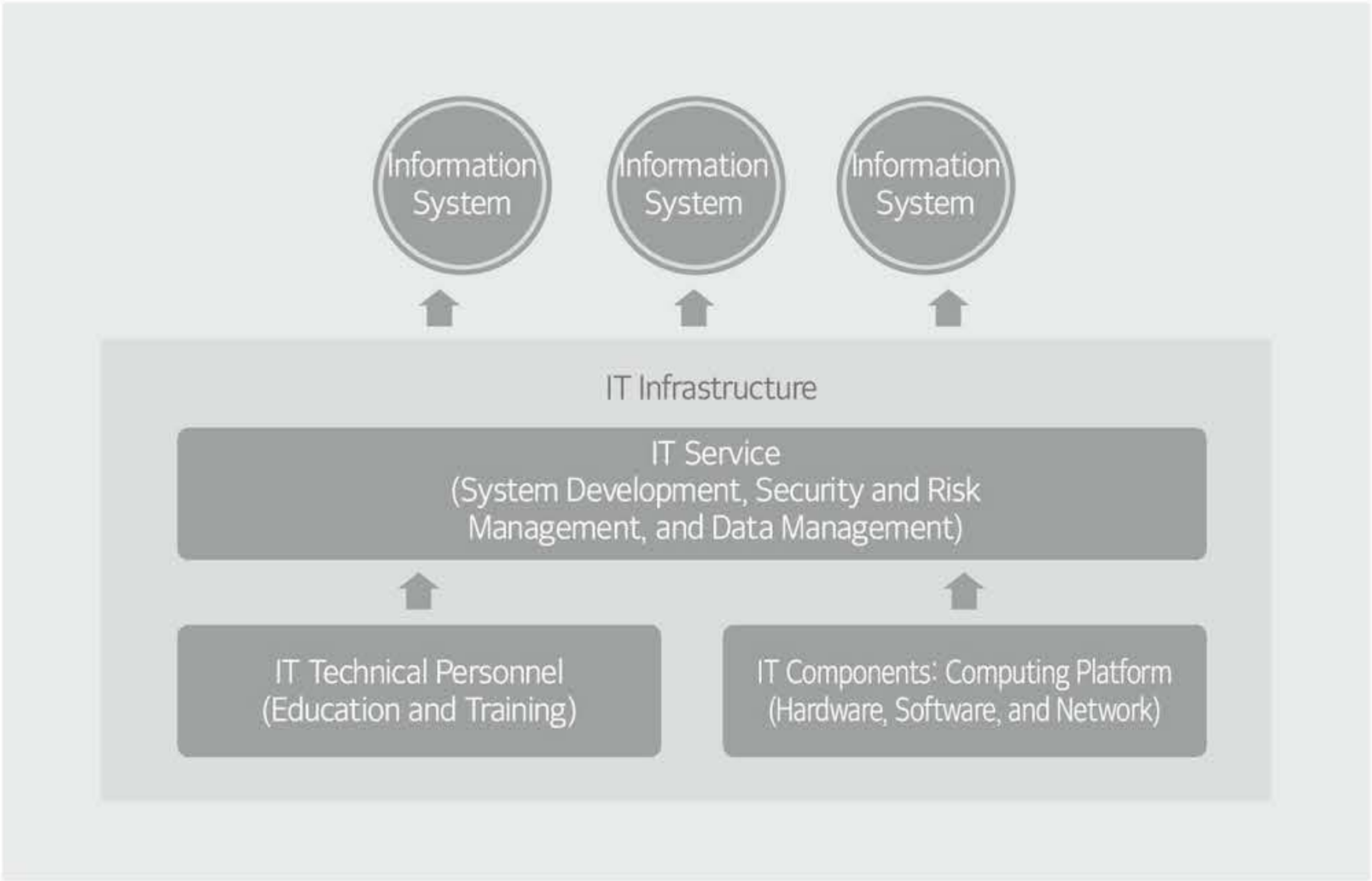
✚ Preview for practical business

Company A has recently launched a new system, but recognized that its hardware capacity is about 30% insufficient during a service. An investigation found that the concurrent user calculation of some services was inaccurate, leading to incorrect hardware capacity selection. The current infrastructure is expected to face inadequacy within three to four months if the company operates the system as is. Therefore, Company A's Planning Team decided to allocate the additional budget to expand the infrastructure in two months. This case shows the practical importance of calculating the information system capacity, according to its structural design and system structure. This chapter describes information structure and size calculation in detail..

01 Information System Structure

A) Concept of information system

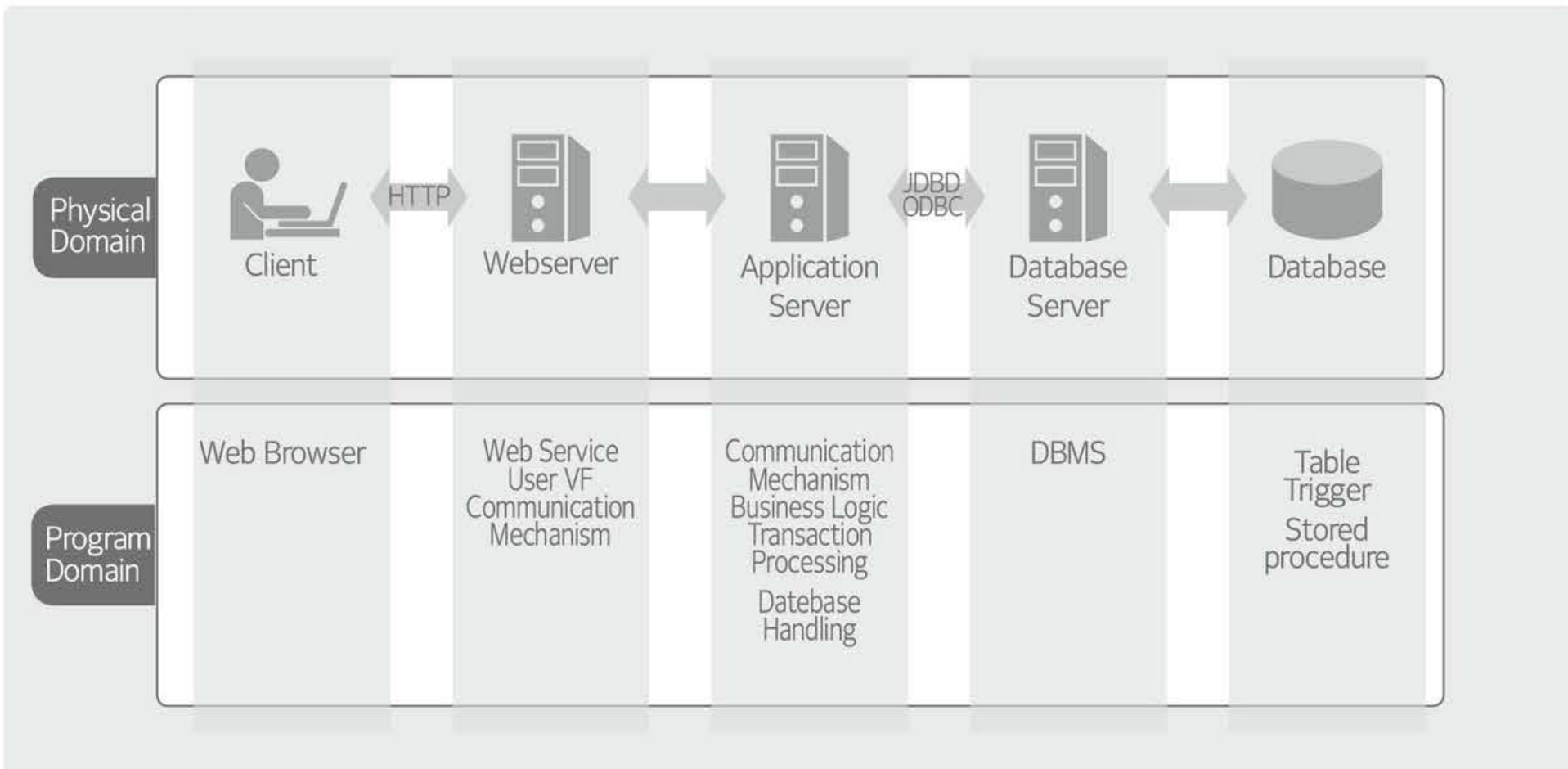
An information system uses information technology (IT) to perform business procedures, such as providing necessary information, or business processing required by enterprises. It creates a system that enables IT personnel to provide IT services by using IT components, such as hardware and software, to perform organizational business procedures. As shown in [Figure 75], the information system provides services by using the technology resources of the IT infrastructure. The IT infrastructure includes computing platforms, such as networks, hardware, software, IT personnel, and IT services.



[Figure 75] Information System and IT-Based Structure

B) Information system structure

There are several types of structures for an information system. A web system architecture and a client-server system architecture are leading examples. The most recent information systems are generally composed of a 3-tier web system architecture, and these information systems can be divided into physical areas (hardware) and program areas (software), as shown in [Figure 76], from a structural viewpoint.



[Figure 76] 3-Tier Web System Architecture

① Information system hardware structure (Physical domain)

A server is the most fundamental component required when implementing an information system, and devices,such as storage and backup disks, are added and configured as needed. A server is a hardware platform that runs server application programs and is classified in various ways, according to performance and type as follows.

• Classification according to performance

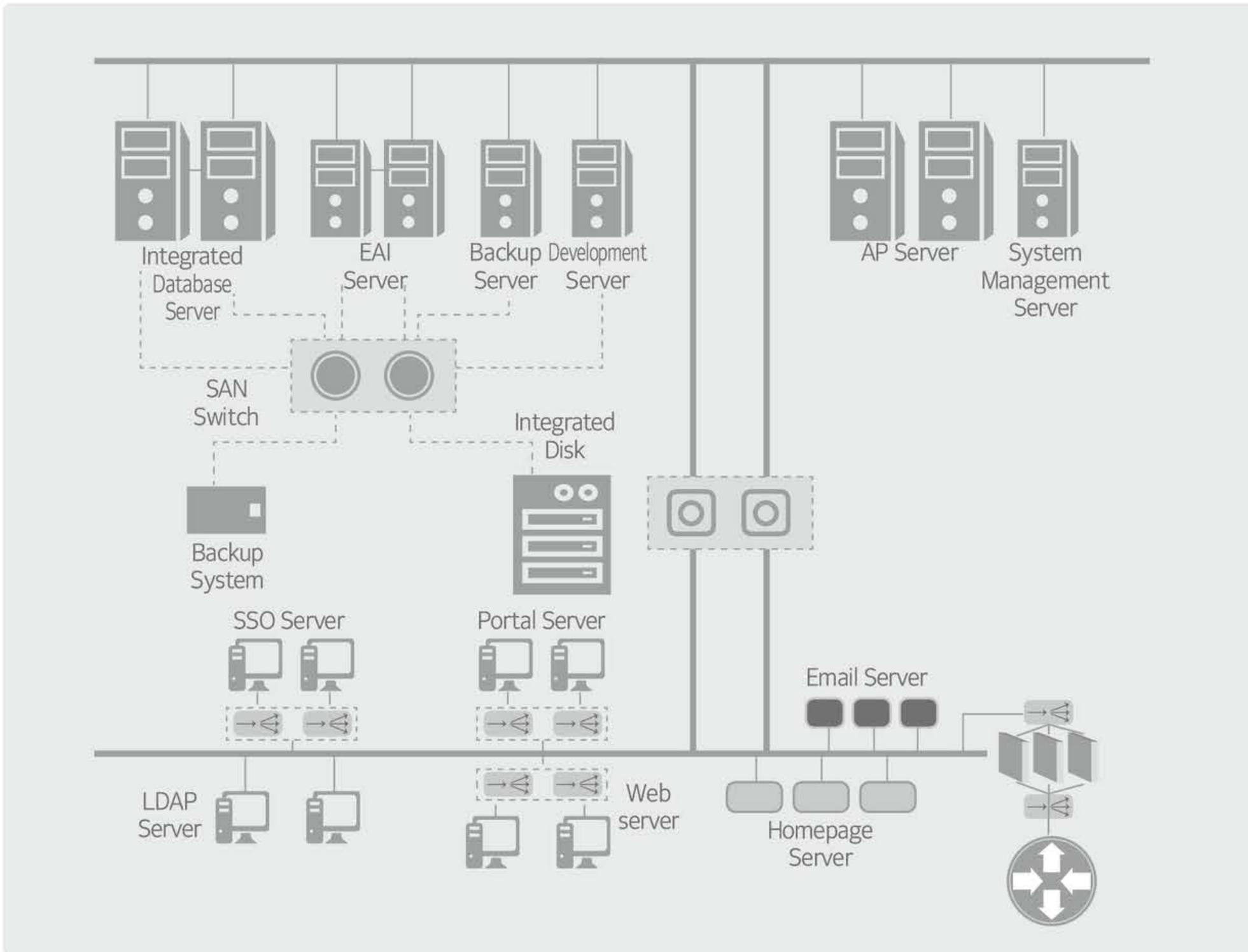
Classification	Description
Entry server	An entry server, typically priced in millions of won, is used by a webserver or application server. It usually refers to a server that can mount one or two CPUs per socket.
Middle-range server	A middle range server, typically priced in tens of millions of won, is used by a database server or enterprise server. Servers that can mount four or more CPUs per socket, and are not considered high-end servers, are called the middle-range server.
High-end server	A high-end server is a very expensive server, typically priced in hundreds of millions or billions of won. It is mostly used by a database server or enterprise server. It usually refers to a server that can mount tens of CPUs per socket.

• Classification according to the configuration

Classification	Description
Rack-mount server	A rack-mount server fits into a rack, which is a steel frame for storing servers or network equipment. It is used in places where multiple server/network equipment is placed, such as in a data center or server room. Rack-mount server sizes are standardized. Standard rack-mount servers are generally 1U, 2U, and 4U sized installed in a 19-inch rack. Power, network, and SAN switch are separately connected.
Tower server	A tower server is installed in an in-house server room, an office, or a store. It is similar to a general PC. When a tower server is installed in a space where people reside or work, such as an office or a store, a low-noise server, developed for office installation, may be a good option.
Blade server	A blade server maximizes capacity while minimizing the size, by plugging several CPU blades into a frame. The server and network are reduced and integrated into a box, and the operating system is pre-installed. The biggest advantage of blade servers is their small size. A blade server can be more densely integrated than a rack-mount server in the same space.

Storage is classified into enterprise or mid-range storage, according to processing capacity. It provides a large storage space, high performance, and high availability, compared to general disk arrays.

[Figure 77] below shows an example of an information system hardware configuration, using a server and various devices. A system architecture or technical architecture design describes its details.

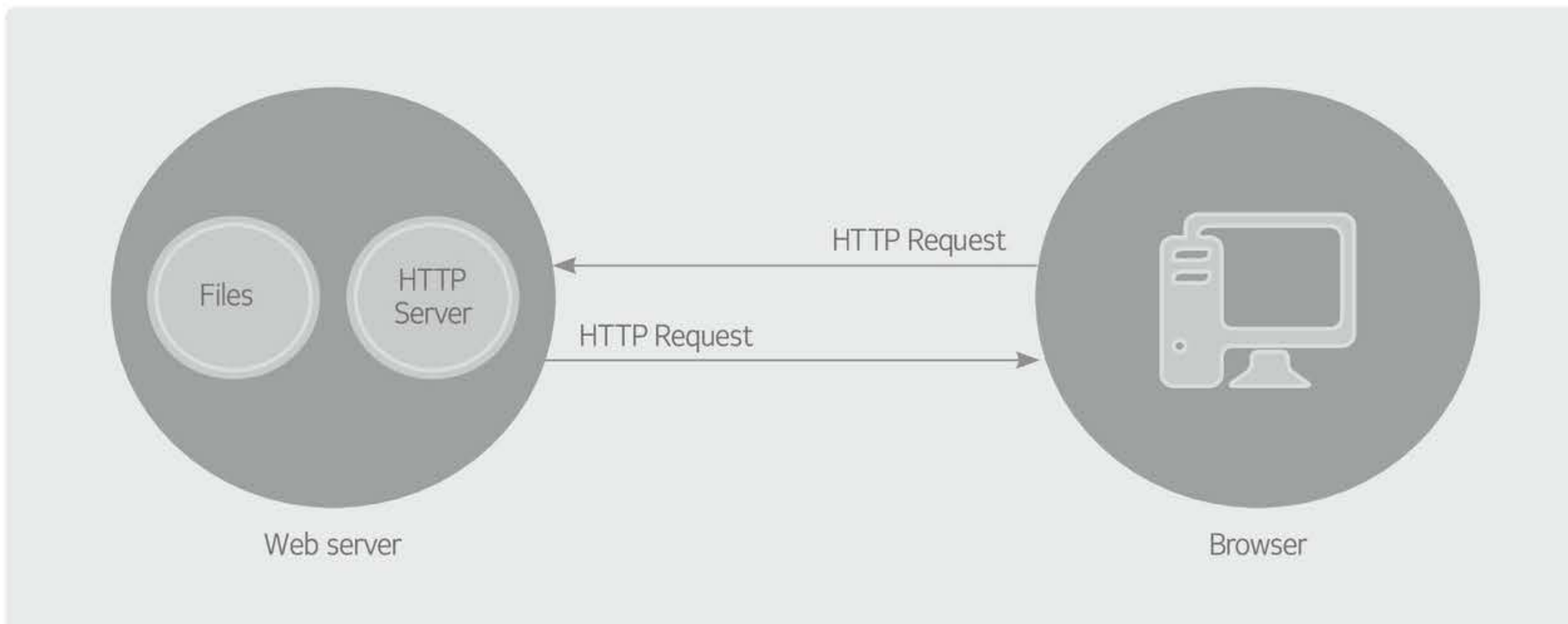


[Figure 77] Information System Hardware Configuration (Example)

② Information system software structure (Program domain)

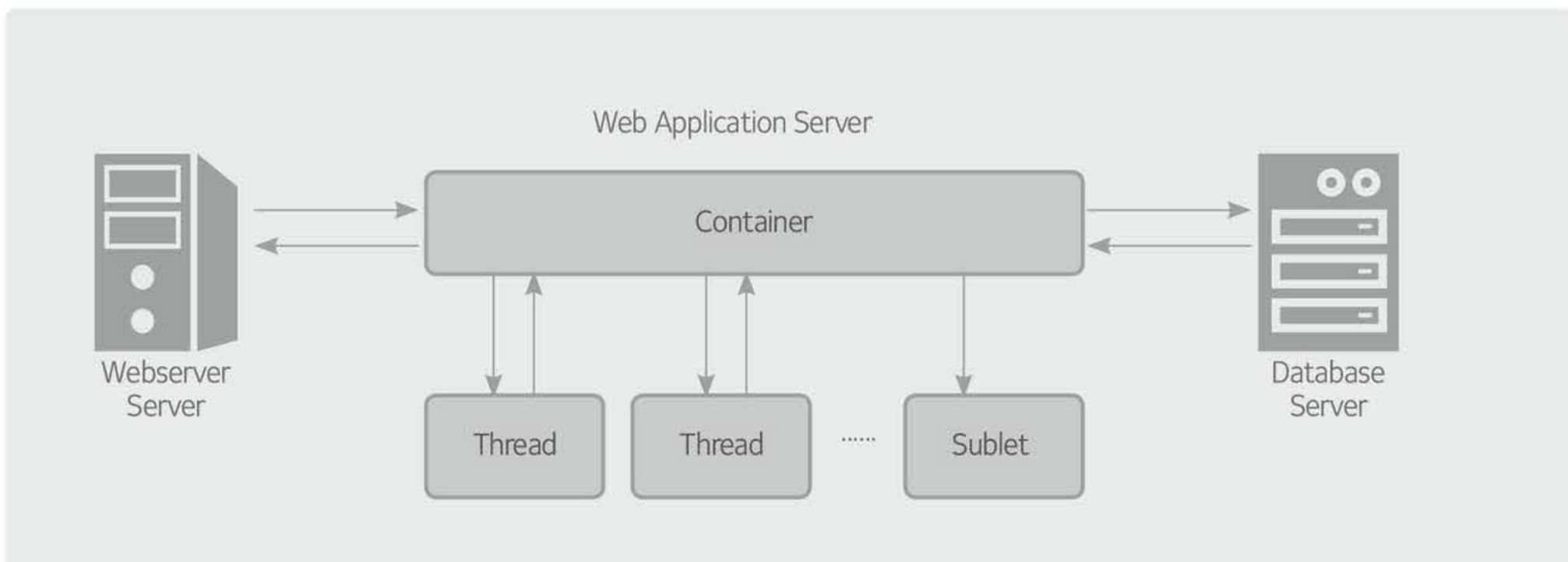
- OS: An OS is defined as system software that operates and manages an application platform and that provides an interface between an application program and the platform. The OS for the server includes Linux, UNIX (BSD, Solaris, etc.), Windows, and macOS. The OS for the server market is quite different from the desktop market, which is dominated by Windows. Linux and UNIX are the most widely used OS, followed by Windows. UNIX servers have replaced most mainframes, and they were followed by Linux, which is a UNIX-like OS.

- Webserver: A webserver provides media content, such as HTML and text, images, and video, to web clients, like web browsers. It provides data through the HTTP protocol to enable a web browser to present information. Leading webserver products include Apache, Nginx, and IIS (see [Figure 78]).



[Figure 78] Webserver

- Web application server (WAS): A WAS is a middleware that enables the stable transaction processing of web client requests and helps develop the distributed systems. It is an engine for database queries or business logic processing, which supports component development and use, application development, web, distributed objects, security, IED, system management, and interface to legacy systems. Leading WAS products include Tomcat, Jetty, JEUS, JBoss, WebLogic, WebSphere, and Node.js (see [Figure 79]).

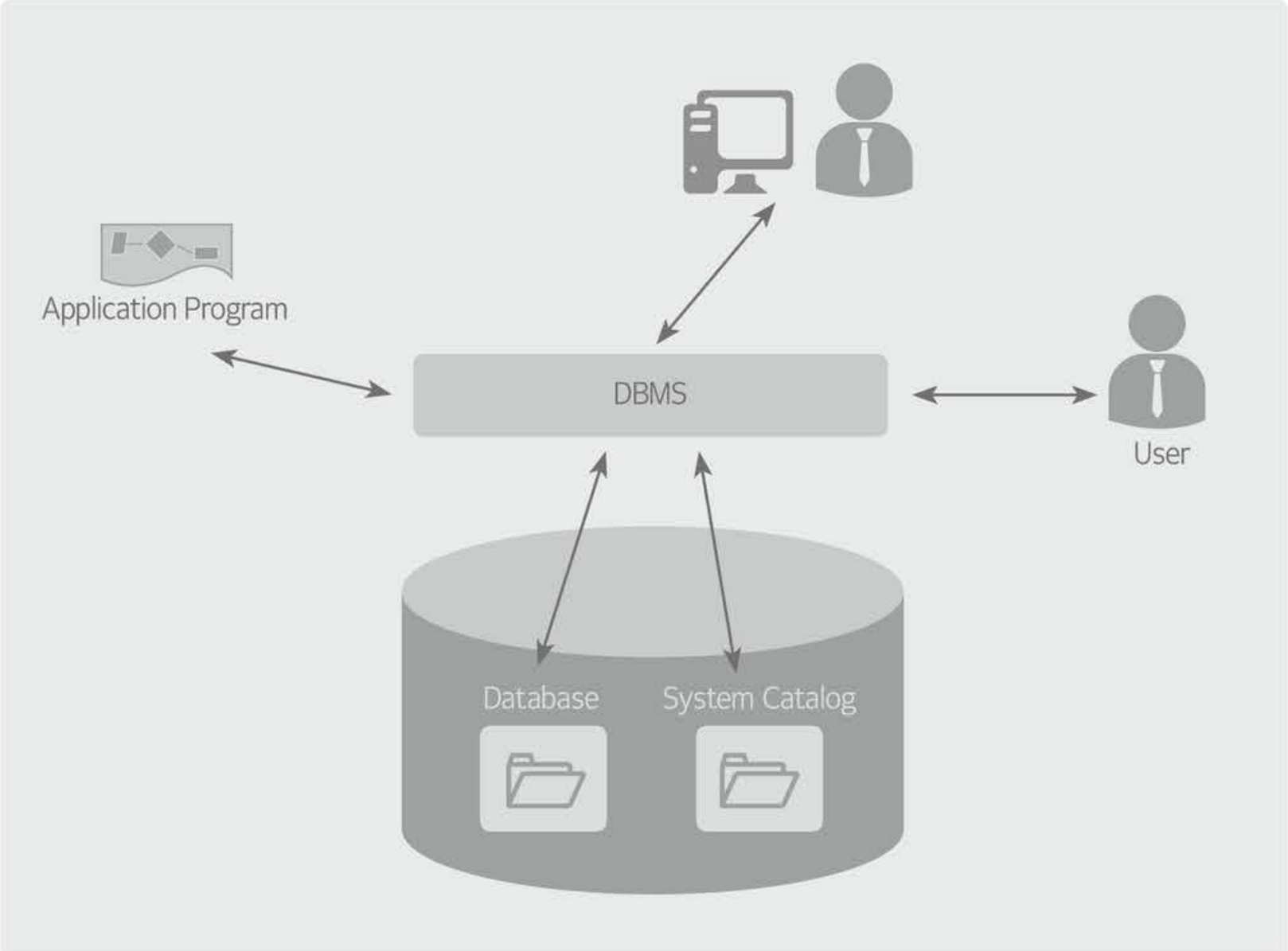


[Figure 79] WAS

- Transaction processing (TP) monitor: A TP-monitor is transaction management middleware that supports the distributed transaction processing and monitors transactions, which are the minimum processing units between active sessions and systems in various protocols, in order to maintain them consistently. Most TP-monitor products comply with the DTP (distributed transaction process) model specified by X/Open, the UNIX environment standardization organization. The leading TP-monitor products include Tmax and Tuxedo.

- DB server (Database management system, DBMS): A DB server provides the database type, database management

technology, and query language for the access to and modification of structural data. It is large-scale storage of consolidated data with minimum duplication and is used by multiple users concurrently. It provides independence between application programs and data and efficient access to data through queries. The system catalog is the tool for database administrators (DBAs) and a system table that lists the definitions or statements of all data objects stored in the database. The leading DBMS products include Oracle, MS-SQL, PostgreSQL, Maria DB, and Cubrid (see [Figure 80]).



[Figure 80] DB Server

02 Information System Capacity Calculation

When implementing an information system, many hardware suppliers or system integrators use the items and rates for sizing calculation, based on their experience, leading to inaccurately calculated sizes.
Moreover, since the hardware size must be calculated according to the nature of the business, load increase rate, and use frequency by the users, it is difficult to determine the appropriateness of the hardware size in the information system implementation projects.
For that reason, hardware sizing has usually been dependent on the project contractor or hardware supplier. It has gained relatively low interest, compared to software development, although the share of hardware in the information system implementation project is very large in the total project cost. Therefore, there is no way to correct it, even if a

hardware component is over or underestimated by a project contractor or a hardware supplier. As such, this chapter reflects the government guidelines for calculating hardware size.

A) Concept of size calculation

System sizing is to estimate or calculate a system's size using a mathematical methodology, based on actual work and applications. It can be defined as a lower-level concept of capacity planning or capacity management to determine performance requirements and performance based on the system architecture application, as shown in [Figure 81]. Sizing is temporarily performed for a specific system or server, and it is also called size estimation because it has a predictive nature.

Subject	Definition	Viewpoint	Time Factor
Capacity management	To generate current and future capacity plans to meet business requirements and balancing cost and balance cost and capacity	Organization	Continuous
Capacity plan	A plan to determine the performance requirements and performance needed for the system based on the outlined system architecture and application work	Organization and system	Continuous
Size calculation	To convert the given basic capacity and performance requirements into system requirements	System	Temporary

[Figure 81] Comparison of Size Calculation Concepts

Sizing is a matter of prediction, and the calculation method significantly affects the accuracy of the calculation. Hardware size calculation methods are generally classified into the mathematical calculation, reference, and simulation methods, as shown in [Figure 82]. It is necessary to apply an appropriate sizing method to get accurate sizing. This guideline uses a mathematical calculation method. However, it is necessary to use all three methods appropriately for size calculation, in order to increase the accuracy of the calculation results and objectivity.

Subject	Concept	Strength	Weakness
Mathematical method	To calculate capacity based on factors for sizing, such as the number of users, and apply the correction factor	The basis for sizing can be clearly presented, and the calculation is simpler than other methods.	If the correction factor is incorrect, the calculation result can far off from the desired value, and it is difficult to provide accurate evidence for the correction factor.
Reference method	To calculate the similar size by comparing the approximate size based on the reference data according to the workload (number of users and DB size)	Relatively safe size calculation since it is possible to compare with the existing business system	The evidence is weak since the result is by comparison instead of calculation.
Simulation method	To model the workload for the target job and calculate the size by simulating it	Relatively accurate values can be obtained	Requires relatively much time and cost

[Figure 82] Hardware Size Estimation

B) Size calculation targets

- Generally, the target for sizing includes not only hardware, but also software or network. As shown in [Figure 83],

the size calculation target in the capacity calculation guidelines is hardware, which is defined as servers, not PCs or other peripherals. While there are several hardware configuration areas, the four areas - CPU, memory, disk, and storage - are the most important in terms of system price and performance, and they are defined by sizing areas.

- CPU: Calculates the CPU size to process the necessary tasks and selects the server model that shows adequate performance.
- Memory: Calculate memory usage by the system software and applications programs for each server, based on the server configuration, according to the CPU size.
- Disk: Calculates disk usage by the OS, system software, database, and database archive/backup for each server, based on the server configuration, according to the CPU size.
- Storage: Calculates the size of the needed storage, based on the server size, according to the CPU.

			System Type	
			OLTP	Web/WAS
Server	CPU		○	○
	Memory		○	
	Disk	System	○	
		Data	○	
Storage			○	○

[Figure 83] Size Calculation Target

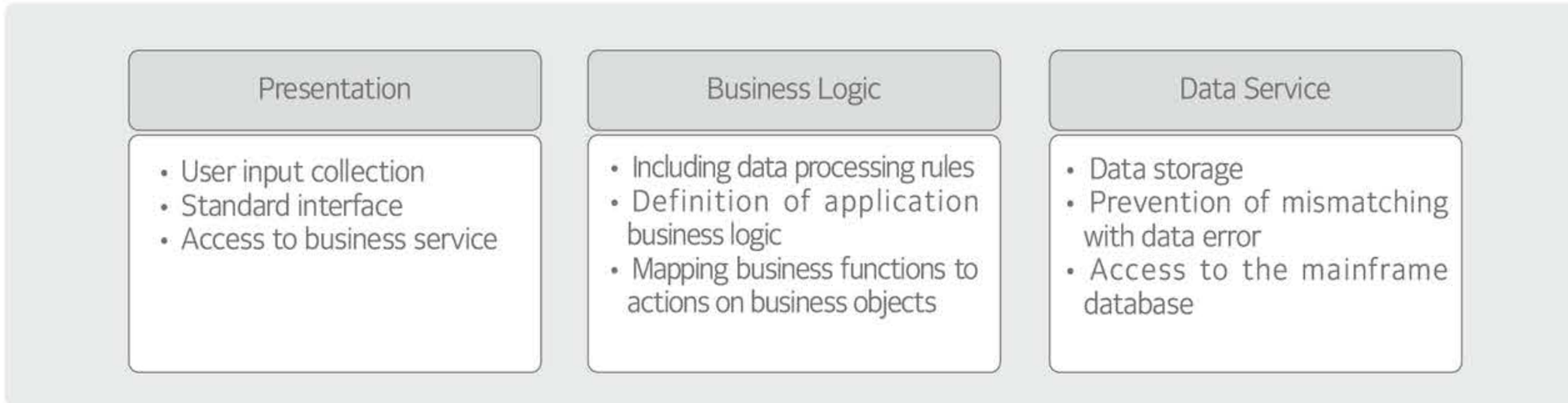
Then apply the performance standard for each size calculation target, as shown in [Figure 84].

Subject	CPU			Storage
	OLTP or OLTP & Batch Application Server	Webserver	WAS	
Performance measurement	tpmC	ops		IOPS
Reference performance standard	TPC-C	SPECWeb2009	SPECjbb2005	SPC-1

[Figure 84] Performance Standard Applicable for CPU and Storage

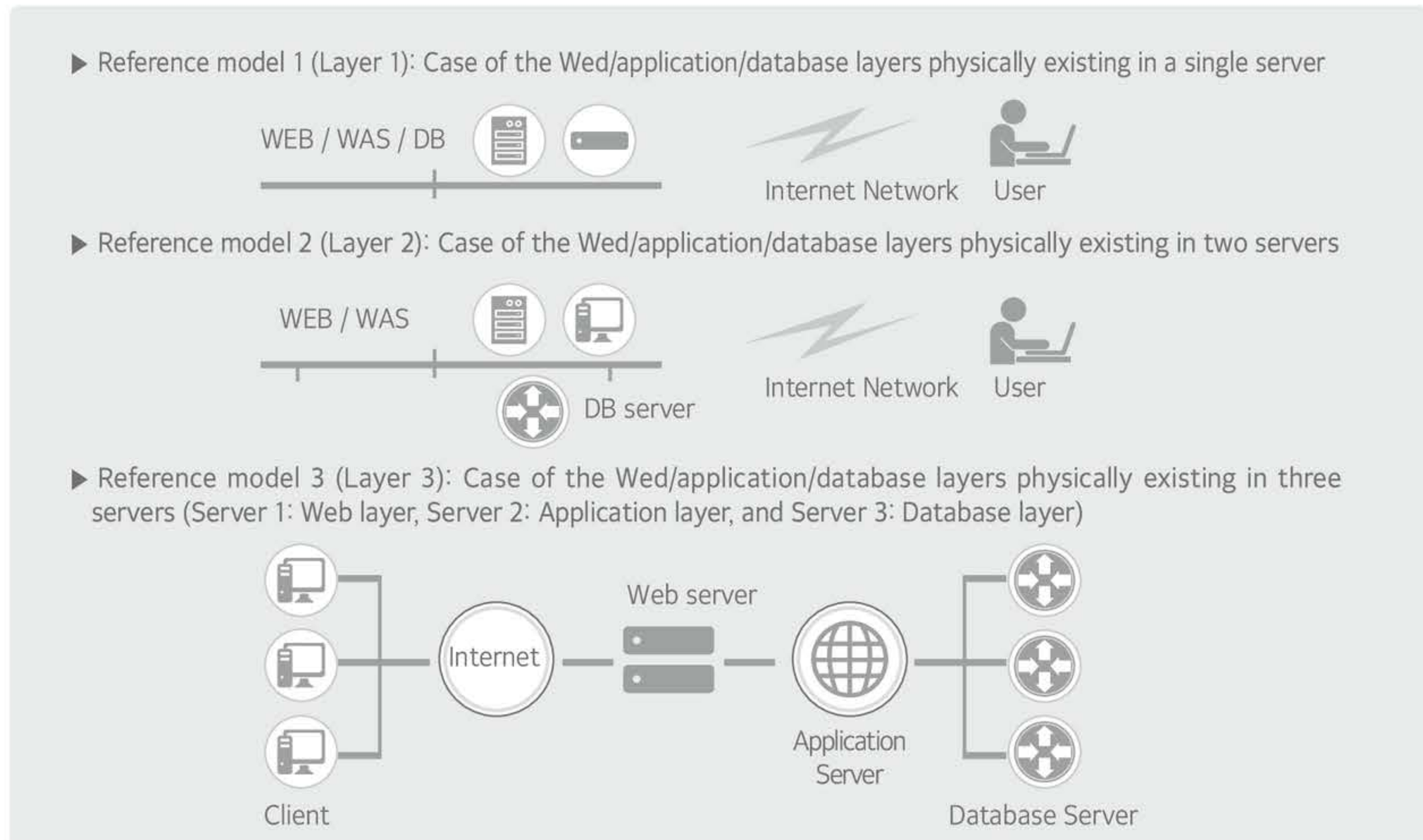
C) Reference architecture for size calculation

Since the size is calculated for each server, the person who calculates the size needs to understand the conceptual model of the overall system architecture that is to be implemented. Most information systems nowadays have adopted the 3-tier architecture, as shown in [Figure 85].



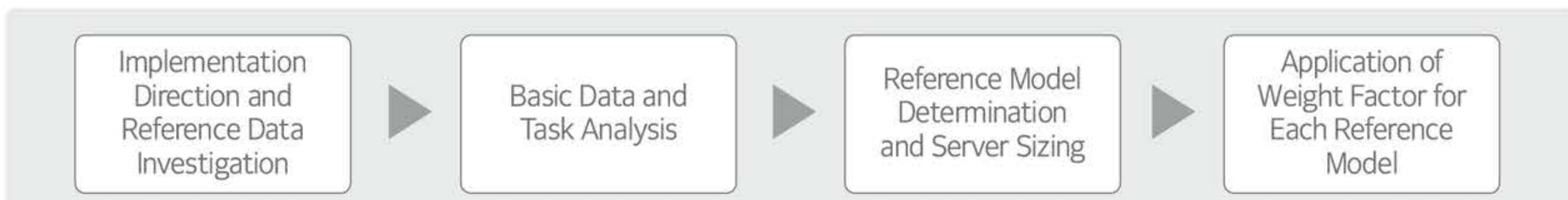
[Figure 85] 3-Tier Architecture

The government standard suggests three types of reference architectures, as shown in [Figure 86], to establish the standard for size calculation and presents the size calculation method for each layer in the architecture for each architecture type.



[Figure 86] Reference Model for 3-Tier Physical Server Configuration

D) Size calculation procedure



① System implementation direction and basic data survey

- Basic data on the tasks and system to be implemented are necessary to ensure the hardware capacity calculation accuracy. To get accurate information, cooperation with the users is crucial, and the basic task analysis and implementation direction are established through consultation with them.

② Basic data and task analysis

- If necessary, analyze the newly added workload,as well as the relevance and complexity of each work, and determine each task's expected load, through the basic data and task analysis results.
- Calculate the reference load by calculating the sum of the expected load for each task.
- Validate the basic data and task analysis.

③ Reference model determination and server size calculation

- Select the reference model suitable for the architecture type of the calculation target system.

Architecture Type	Calculation Condition	Calculation Criteria
Reference model 1	Processing of all web, application, and database layers in a single server	Server 1: OLTP
Reference model 2	The web and application layers processed in server 1 and the database layer processed in server 2 or the web layer processed in serve 1 and the application and database layers processed in server 2	Server 1: Web/WAS Server 2: OLTP
Reference model 3	The web layer processed in server 1, the application layer processed in server 2, and the database layer processed in server 3	Server 1: Web/WAS Server 2: Web/WAS Server 3: OLTP

- Select various correction coefficients, based on the task analysis data investigated in Step 2.
- Calculate the size for each component of the server in the reference model.

④ Application of weight factor for each reference model

After the capacity calculation of each item of the server in Step 3, Step 4 applies the weight factor for each server, according to the architecture type.

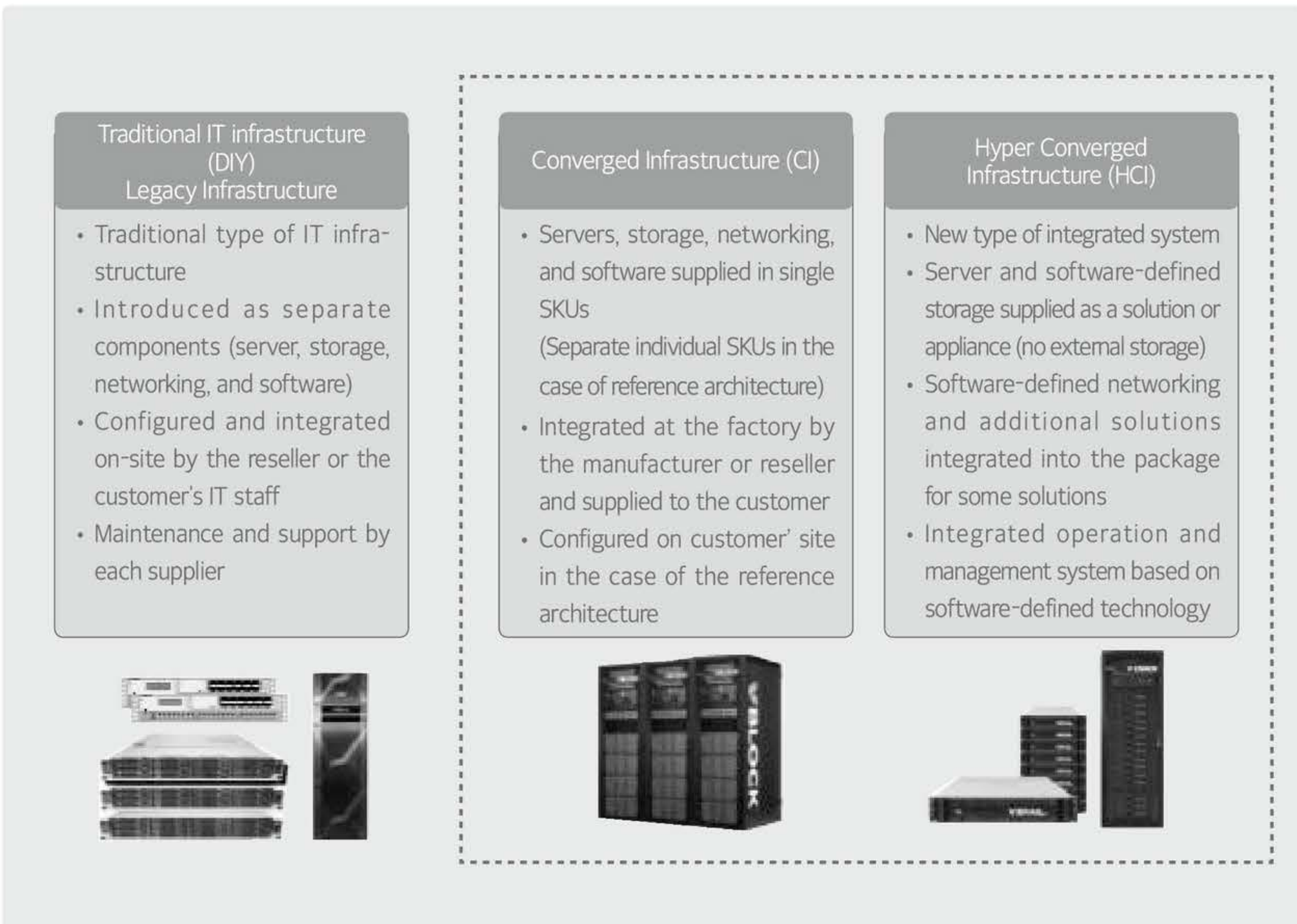
E) Size calculation method for each hardware component

This section describes how to select the size for each server component. As mentioned in the section for the procedure for size calculation, determine the server, based on the architecture type, and calculate the CPU, memory, disk, and storage sizes for each server.

Refer to "TTA Guidelines on Calculating Information System Hardware" for how to calculate the detailed size of each element.

03 Latest Technology and Trends of Information Systems

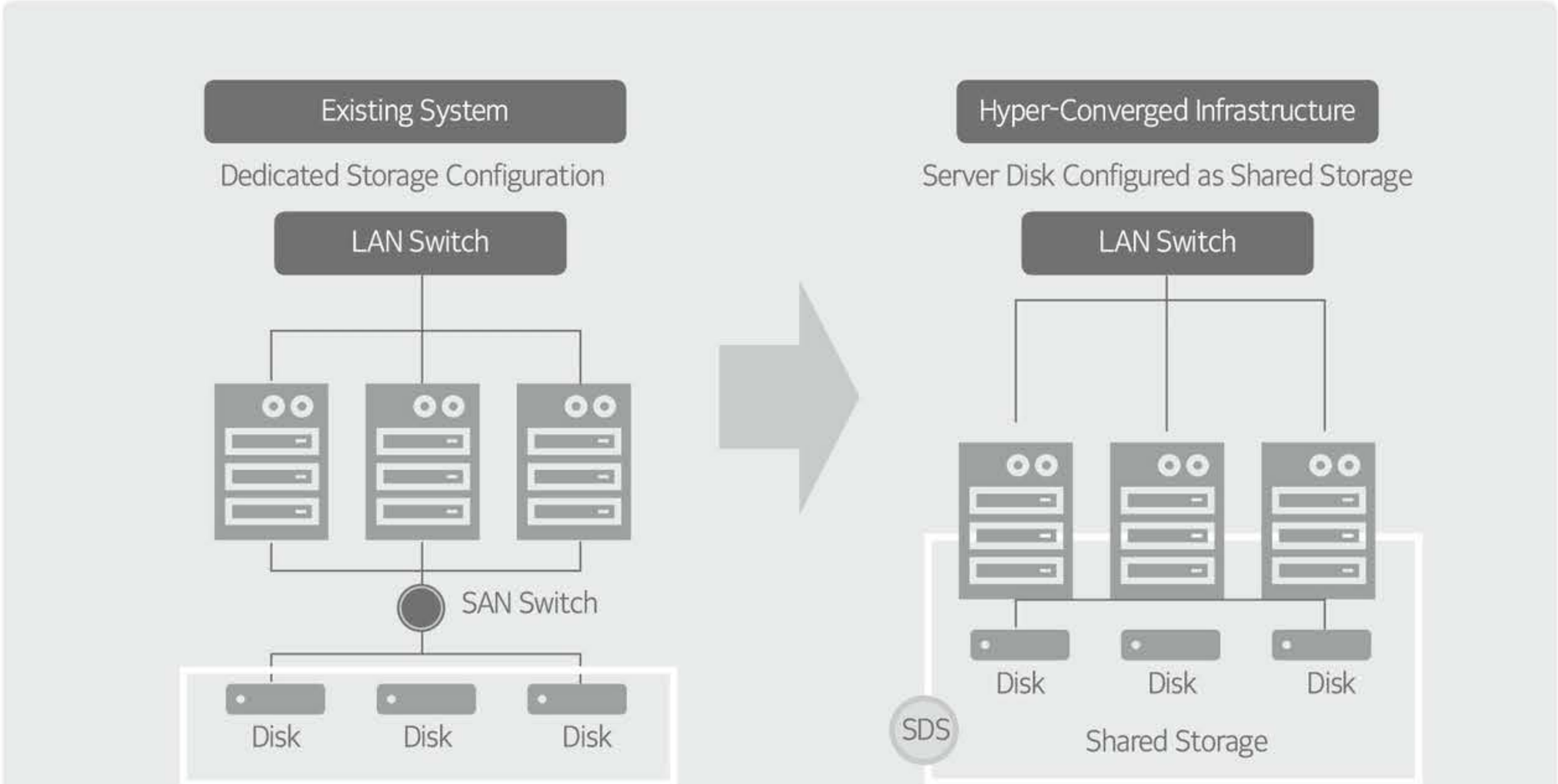
Information system infrastructure is generally implemented by separately purchasing servers, storage, and network devices from each vendor, then integrating them for the project. Moreover, they also separately purchase software, thus they need to test and verify if it works properly with the hardware. However, such a complex structure and procedure inevitably have led to inefficiency from the implementation time and system management aspect.
The converged infrastructure is a concept introduced to solve the problem. It integrates the hardware, like storage, server, and network, with the virtualization solution and the management software, supplied by each vendor, into a product and validates it in advance for supply to customers (see [Figure 87]).



[Figure 87] Conventional Infrastructure vs. Converged Infrastructure

However, despite the advantage that the converged infrastructure enabled quick service implementation, through the integration and validation in advance, it was not actually well received in the market,due to its poor performance compared to input resources. Since the product integrates the server, the network, and the external storage, the cost was exceptionally high, and the system had a problem with scalability. In particular, the high-performance external storage was a severe burden from the cost and structural aspect.
The hyper converged infrastructure (HCI) is a concept that overcomes the limitations of CI by eliminating the

expensive external storage. As shown in [Figure 88], HCI groups the direct attached storage (DAS), which is located under a general-purpose x86 server, with the software-defined storage (SDS) technology through the IP-based network and uses it as a storage pool, similar to a storage area network (SAN). The architecture eliminates the structural complexity and reduces costs most of all by eliminating external storage.



[Figure 88] Converged Infrastructure vs. Hyper Converged Infrastructure

Ⅷ. Fault Response Technology

Learning objectives

1. To be able to explain the HA/FT technology and operational principle.
2. To be able to explain the DR configuration and utilization.

Keywords

SPOF, Hot-standby, Mutual takeover, Concurrent access, Mirror site, Hot site, Warm site, Cold site, RPO, RTO, Fault-tolerant technique

✚ Preview for practical business **Information system failure at Bank A lasting until late in the afternoon. "Blocked wagesfer", The Hankyore.**

Information systems have the role of not only supporting business, but also creating business. A failed bank information system can have an enormous impact. A fire at OO Card's IDC center in 2014 is a similar case. Enterprises must improve the availability of information system services, through a high availability (HA) configuration or disaster recovery (DR) center in order to ensure that their information system safely and continuously provides service. Most major information systems are basically configured with redundancy. An important banking system has a spare system, in addition to redundancy, to ensure that the reserve system can continue to provide information services at the time of disasters, such as an earthquake or a fire. The DR center is a place that provides such a spare system. DR centers are configured in different ways, depending on the importance of the information system. When a failure or disaster occurs, the time required to recover in order to provide normal service is called the recovery time objective (RTO). Moreover, the tolerable level of data loss in a failure or disaster is called a recovery point objective (RPO). The DR center is classified into a mirror, hot, warm, or cold site, based on RPO and RTO. This section describes techniques for providing service continuity of information systems.

01 High Availability (HA)

A) Concept of HA

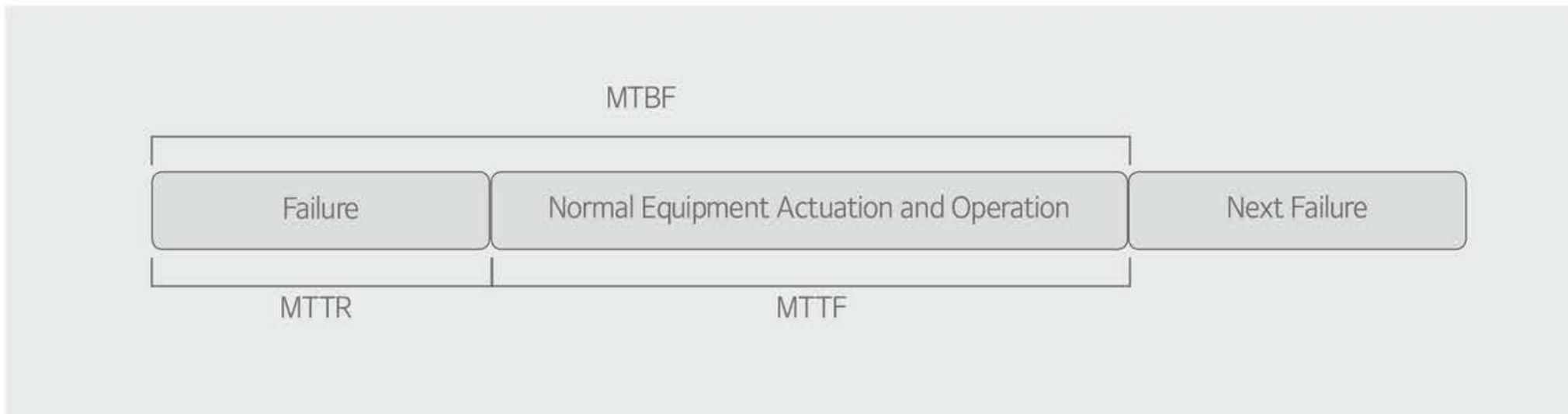
In information systems, HA refers to an uninterrupted service. However, it is not easy to make a system,that consists of numerous parts and software, 100% available. There must be no failure, or services must be available to users in a usable state, even at the time of failure, in order to guarantee uninterrupted availability. In general, the system fails at some point. Therefore, the way to guarantee service continuity is to prepare for failure, rather than preventing failure. HA prepares for failure by configuring two or more systems using clustering.

A typical HA criterion targets 99.999% (called 5 nines), representing less than 5 minutes and 15 seconds of unplanned service downtime per year. The scale of the HA level is expressed by the number of nine, and the formula for calculating the availability is the mean time between failures (MTBF), divided by the sum of the MTBF and the mean time to repair (MTTR).

$$A = \frac{MTBF}{MTBF + MTTR}$$

- A = Availability
- MTBF: Mean Time between System Failures
- MTTR: Time to Restore

[Figure 89] Availability Calculation Formula

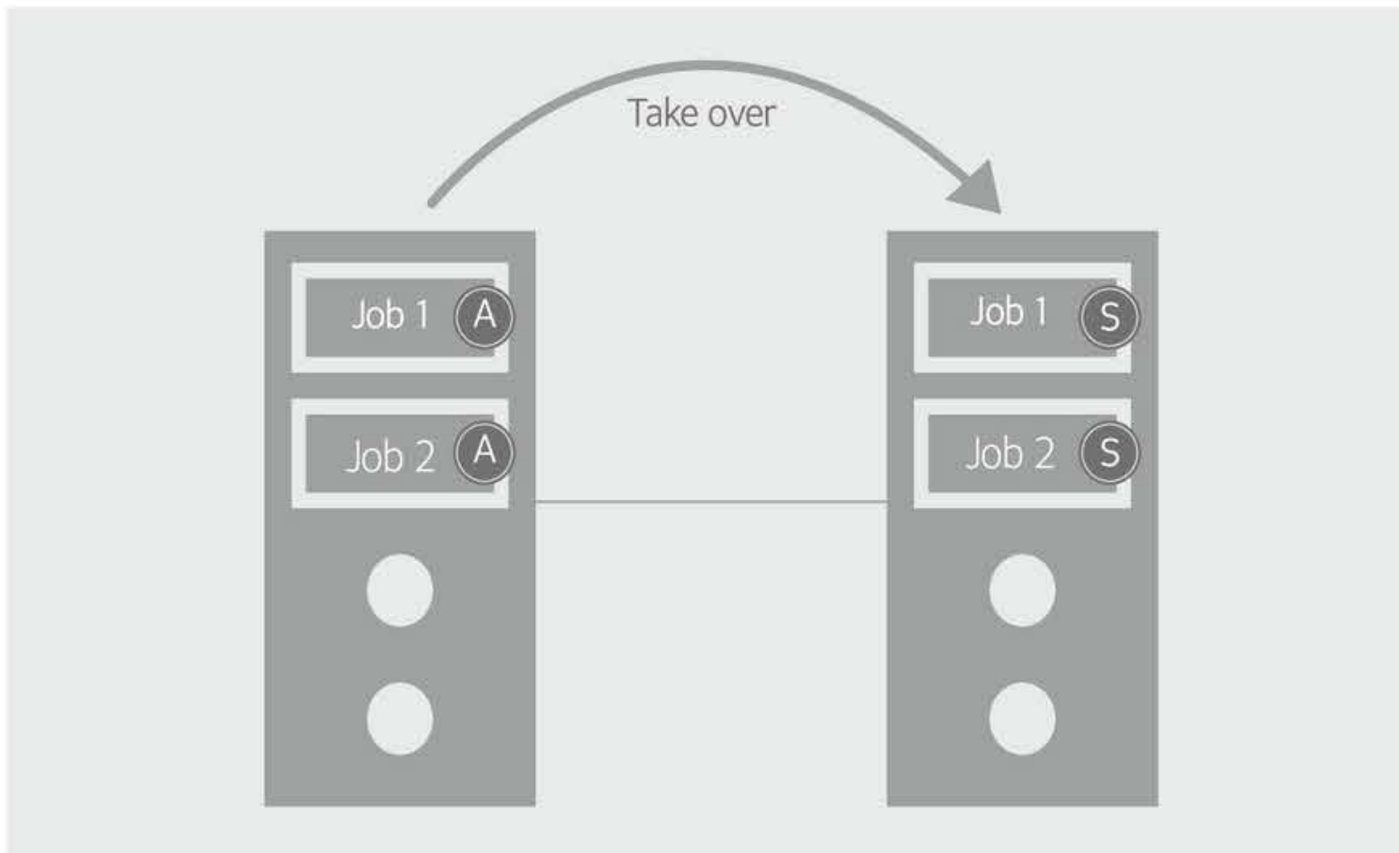


[Figure 90] Availability Indicators

B) HA configuration type

① Hot standby (Active-standby)

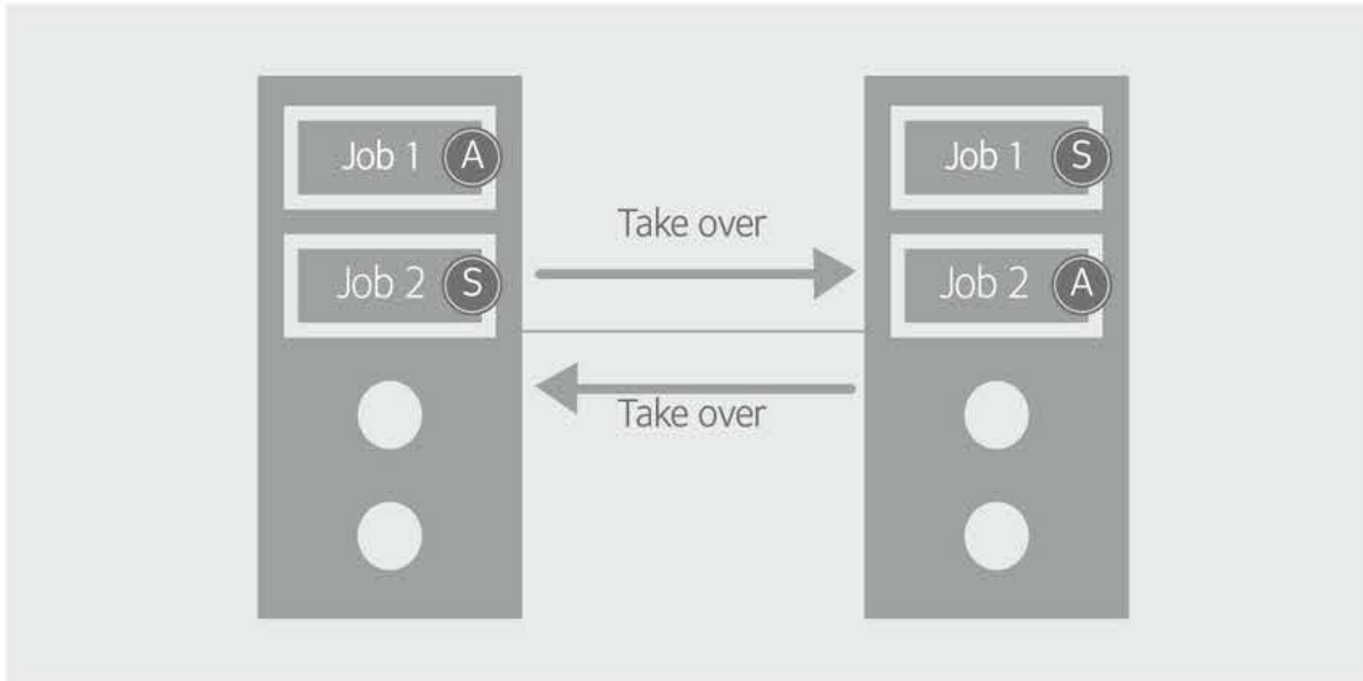
Hot standby is the simplest structure of HA clustering implementation. It consists of an active server and a standby server that are normally in a standby state. The standby server is powered on, and the operating system is running. In some cases, a standby server is used as a development system. If the active server fails, due to hardware, network, or process problem, the failure is detected by the heartbeat network, and all HA service operations are automatically switched to the standby server, through a fail-over operation.



[Figure 91] Hot Standby Configuration

② Mutual takeover (Active-active)

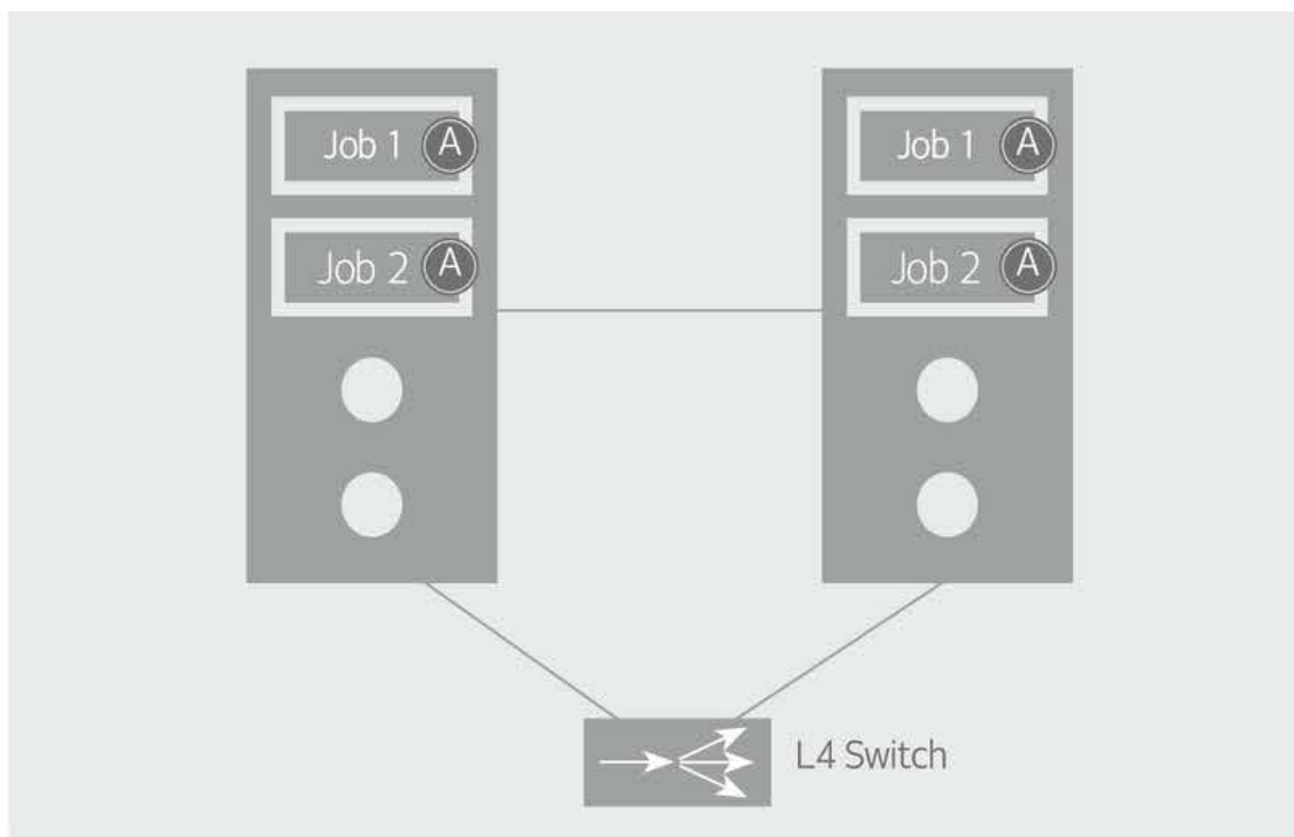
The mutual takeover is the case of two or more systems operating with separate servers. When a server fails, its services are switched to the designated server (takeover) to continue the services. Each server must have sufficient system capacity in order to provide services for the two servers, in preparation for a fail-over at the time of a failure.



[Figure 92] Mutual Takeover Configuration

③ Concurrent access

It is a structure in which multiple systems process tasks in parallel in order to provide availability. All servers operate in an active state. Even if a server fails, the service continuity can be guaranteed without a fail-over. In this structure, two or more servers perform load balancing, using an L4 switch to perform the same task.



[Figure 93] Concurrent Access Configuration

02 Fault-Tolerant System

A) Fault-tolerant system

A fault-tolerant system can continuously perform the functions specified in the design, even if a system part fails. A fault-tolerant system cannot use certain functions when a component fails. When components continue to fail, the unavailable functions gradually increase, and the system shuts down when a critical failure occurs.

[Table 23] Troubleshooting Steps

Step	Description
Fault detection	- Analyzes which module has caused fault, through comparative logic.
Fault diagnosis	- Determines whether the fault is temporary (transient) or permanent (hard) and excludes the hard module from an operation.
Fault isolation	- Blocks the spread of errors by fault.
Fault recovery	- Eliminates the module that caused the fault in order to recover and reconfigure the system.

[Table 24] Fault–Tolerant Technique

Technique	Type	Description
General faulttolerant technique	Checkpoint technique	Error detection for source code that may cause a fault
	Protocol monitoring	Application of fault-tolerant technique, through protocol monitoring and tracking
Hardware tolerant technique	Triple modular Redundancy	Configuration of modules in triple redundancy so that three or more processors perform the same operation for the same inputs
	RAID	Fault tolerance through disk mirroring and distributed storage of parity bits
	Duplication with Comparison	Hardware redundancy for fault detection
	Stand by Sparing	Use of spare hardware for fault detection
	Watchdog Timer	Initialization with periodic timer operation
Software tolerant technique	Checkpointer	Rerun from the point of the checkpoint, when there is a fault
	Recover block	Rollback and retry of a single processor
	Conversation	Processing between multiple processors
	Distributed rollback	Rollback technique in a distributed environment

03 Disaster Recovery System (DRS)

A) Definition of DRS

DRS is an overall plan and system to minimize the impact of a disaster on a business, by locating all or part of the information system infrastructure in a location different from the disaster point,then quickly recovering from a disaster

when it occurs.

[Table 25] DR Center Types

Type	Description	RTO
Mirrored site	- Construction of the same facility, IT devices, and network resources as the main center in a remote place - Real-time simultaneous services in the active-active state	Immediate
Hot site	- Construction of the same facility, IT devices, and network resources as the main center in a remote place - Standby state maintained (Active-standby) - Switching the remote system into the active state at the time of a disaster at the main center - Keeping data up to date, through synchronous or asynchronous real-time mirroring	Within 4 hours
Warm site	- Placing of only crucial IT resources at the DR center - Transferring of critical components in the main center to the DR center for operation at the time of the disaster	Days - weeks
Cold site	- Only keeping data in the remote location, but not IT resources, for services or only keeping minimal resources, such as power, communication, and network - Replacing the failed system by procuring computer components and establish a network at the time of disaster	Weeks - months

B) Disaster recovery goal

The most important factors to consider when implementing a DR system are RTO and RPO. RTO represents the time required to return to normal operation, when a disaster or failure occurs in the operating system. RPO represents an indicator of tolerable data loss, i.e., the amount of data that can be lost.

[Table 26] RTO and RPO

Type	RTO
RPO	Data loss converted into time at the time of disaster
RTO	Maximum time to allow downtime

IX. Cloud Computing Technology

Recent trends and issues

Conventional IT environments, based on high-performance servers represented by mainframes,as well as UNIX servers, are rapidly transforming into cloud computing environments characterized by virtualization, immediateness, and linear scalability (scale-out). The importance of cloud computing is increasing, particularly with the rapid growth of SNS and Internet services, and big data and IoT become keywords of IT. Understanding cloud computing is essential in order to respond quickly to rapidly changing business environments and to provide an IT environment with continuous scalability.

Learning objectives

1. To be able to distinguish and explain the characteristics, concepts, and types of cloud computing.
2. To be able to explain virtualization technologies and components that are the basis of cloud computing.
3. To be able to distinguish types of server virtualization technologies for cloud computing and explain the differences.

Keywords

Cloud computing, Cloud type, Virtualization, OpenStack, Docker, Linux container, Mesos, Kubernetes

✚ Preview for practical business

Gartner, which annually publishes strategic technologies for the year, began including virtualization in "Gartner's Top Ten Strategic Technologies" in 2006 and added cloud computing in 2009. AWS started its service in 2006, achieved annual sales of 5 trillion won in 2014, and generated tremendous sales growth every year. Now, cloud services are becoming a necessity, not an option. Most of the technologies, such as AI, big data, and blockchain, which have recently become the subject of conversation, operate based on cloud services that basically provide scalability and speed. It is why most cloud service providers provide platforms for new technologies. The server virtualization is changing. In the past, server virtualization was performed using a hypervisor. In recent years, container-type virtualization has emerged. It is quickly establishing a position in the cloud service market because it is lighter and faster than the existing methods. Typical container technologies include Linux Containers (LxC), Docker, Kubernetes, and Openshift.

01 Definition of Cloud Computing

Cloud computing was developed to efficiently utilize the surplus computer capacity of each server and to flexibly respond to the demand of uncertain services at enterprises. While users had to own and manage resources in order to use computer resources before the advent of cloud computing, cloud computing allows them to use computing resources, provided through the Internet, in a virtualized form when they need them.

The key to cloud computing is that computer resources are available through the network and are provided in an invisible state so that users can work anywhere using the computer resources they need, even without knowing the complex interior. Around 2008, various services that utilize computing resources on the Internet were launched, and "cloud computing" began to be used as a comprehensive term. Users could access desired services from the Internet without requiring expertise in the technology infrastructure they support.

The following benefits are expected from cloud services.

- Costs: CAPEX[6] reduction, OPEX[7] increase, and TCO[8] reduction
- Period: Reduced development time and product development cycle
- Operation: Reduced operating personnel and strengthened resource efficiency
- Product: Improved product integration

Although the terms "cloud service" and "cloud computing" are generally used interchangeably, the Korea

6 CAPEX (Capital expenditure): Expenditures spent to generate future profits

7 OPEX: Operating expenditure

8 TCO (Total Cost of Ownership): Total cost of owning a system, including the initial investment and maintenance costs

Telecommunication Association 9TTA defines them as follows:

〈Table 27〉 Definitions of "Cloud Computing" and "Cloud Service"

Terms	Definition
Cloud service	An on-demand outsourcing IT service that provides a user-centered cloud computing environment
Cloud computing	A computing environment that leases IT resources through the Internet, based on virtualization and distributed processing technology and pays a fee, based on usage

A) Comparison of cloud computing and other computing technologies

① Grid computing and cloud computing

Grid computing is defined as "a structure for virtualizing and resources, such as computers and data, on a network for integration and dynamically creating virtual computers if needed." Grid computing and cloud computing are similar in that they use a distributed computing structure and provide virtualized computing resources. The difference is that grid computing uses all the computer resources on the Internet, while cloud computing uses only the computing owned by the operator.

〈Table 28〉 Differentiation between Grid Computing and Cloud Computing

	Grid Computing	Cloud computing
Computer location	Geographically scattered and managed by different organizations	Geographically scattered but managed by a central organization
Computer configuration	Heterogeneous mixture	Most of the same model
Standardization organization	Existing	None
Technical standard	Standards for resource management, scheduling, and data management security	None
Interconnectivity	Important	Not considered
Usage	Highly parallel applications, such as scientific and technical calculations and large-scale operations	General-purpose applications, such as web applications

② Utility computing and cloud computing

Utility computing is a type that provides computing resources and charges for the resources used. Utility computing and cloud computing are similar, in that they charge for the resources used. Cloud computing has developed the abstraction of computing resources, based on utility computing.

02 Cloud Computing Types

A) Classification according to the service type

Cloud services can be classified into IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service), according to the service type.

① IaaS

IaaS provides infrastructure resources (server, storage, and network) as a service through the Internet. Although the

infrastructure resources provided by IaaS are generally in a virtualized environment, it is not a must for IaaS. The service that provides non-virtualized physical infrastructure resources is called a bare-metal cloud. IaaS users must directly manage areas beyond the OS. The service is similar to receiving a server that is hosting from an existing service, provided from the user's aspect.

② PaaS

PaaS is a service method in which the concept of SaaS is applied to the development environment. It uses the necessary development and operating environment as a service, without building a development and operation environment. The service scope of PaaS ranges, from the application development environment, to the service distribution and operation environment. PaaS provides the network infrastructure to the runtime for application execution, and the management scope of the user is the applications and data above the application runtime.

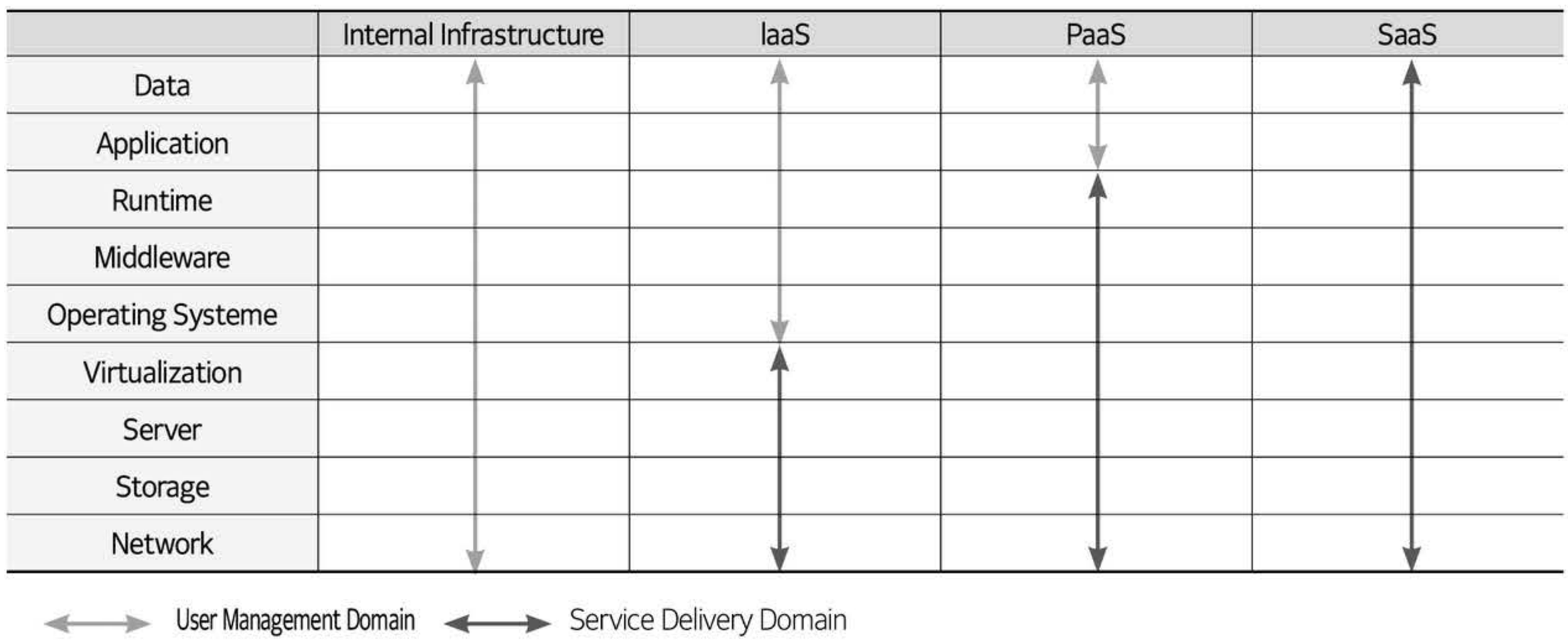
For example, a user developing a new Java web application that uses MySQL as a database, can quickly prepare a development/operation environment, and can quickly develop and service applications, by using PaaS service that provides MySQL and Apache Tomcat.

③ SaaS

SaaS is a service that provides software and is a model that delivers software serviced from the center to users, through a client, such as a web browser. SaaS users can use functions they need, without knowing where the software is located, which OS it uses, and in which language it is developed. Leading examples of SaaS include Google Docs and SalesForce.com. SaaS services generally have the following characteristics:

- Access with a web browser: Access with a web browser, instead of an installed software
- Usage-based cost: Paying fees, based on software use
- On-demand: Needed software available immediately
- IT requirement optimization: No management and scalability issues with the IT infrastructure

[Figure 94] shows the management scope of IaaS, PaaS, and SaaS users and service providers.



[Figure 94] Management Scope of in Each Cloud Service Type

B) Classification according to the cloud operation form

Cloud services can be classified into three levels, based on the service operation form. The following table shows the classification.

〈Table 29〉 Cloud Service Type Based on Disclosure Level

Cloud service	Description
Public cloud	• Cloud services disclosed to unspecified masses over the Internet and are accessible by the Internet
Private cloud	• Cloud services implemented over the closed network by enterprises and institutions to allow access only to authorized users
Hybrid cloud	• Services that use both cloud services open to the outside and closed cloud services, by combining the public cloud and private cloud • The compatibility by moving services deployed in the public cloud to the private cloud domain and sending services in the private cloud domain to the public is very important.

03 Server Virtualization Technology

The conventional computing environment generally installs an OS-dependent on single hardware and runs applications above the OS. Virtualization refers to the technology of abstracting the computer resources of multiple computers to make them look like servers, or by abstracting the computer resources of a computer to make them look like multiple computers. Since the server virtualization technology makes it possible to run one or more operating systems on a computer, it can maximize the utilization rate of computer resources and management efficiency. Cloud services generally use virtualization technology to provide abstract computer resources to users. The x86 virtualization technology, in particular, has been rapidly spread, based on the technology and market maturity, and has become the basis of the cloud computing environment for the logic sharing and allocation of computing resources beyond the physical limitations.

A) Hypervisor

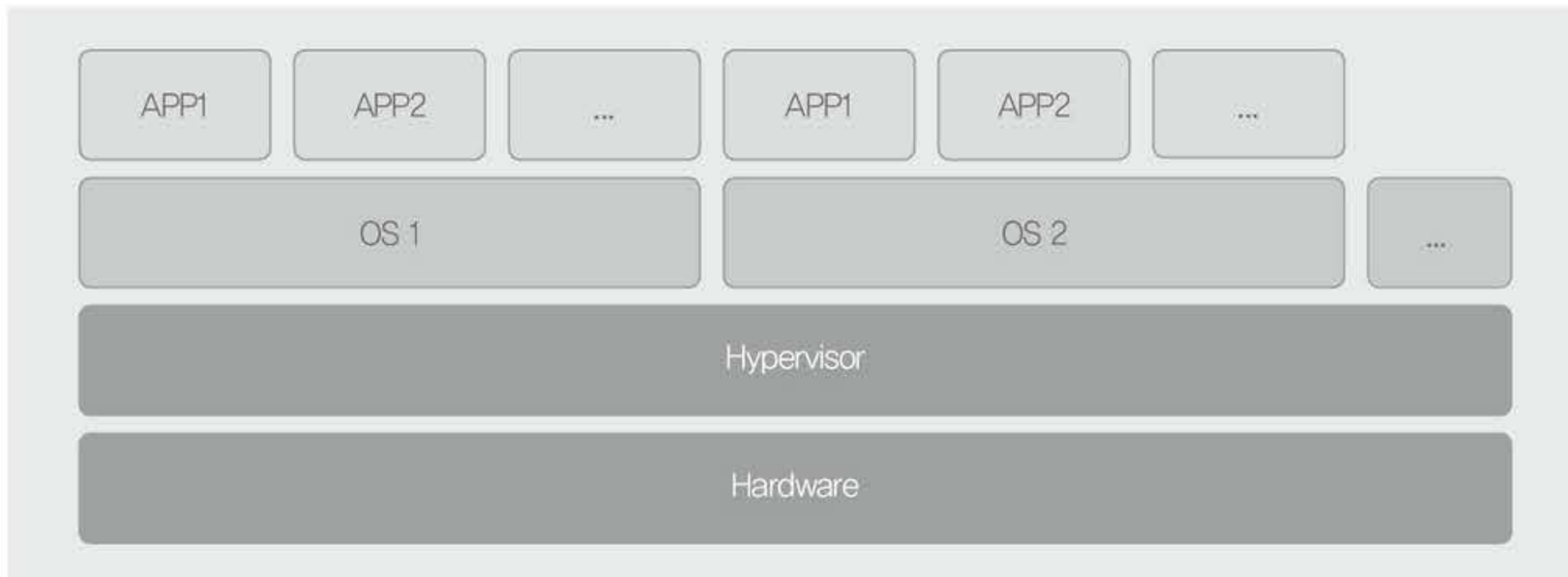
In server virtualization, a host OS is a physical server, and a guest OS is a virtually provided server. Server virtualization technology can be classified, according to how a virtual server allocates physical resources in order to make it look like a physical server. A hypervisor is a technology of virtualizing a physical server. It is a logical platform for server virtualization. It is also called a virtual machine monitor (VMM). The hypervisor can be classified by how it is installed and virtualized.

B) Hypervisor types

Hyperviors are classified into native types and hosted types, according to the installation method. The native type is installed directly on the hardware, while the hosted type (see [Figure 95]) is installed on an OS, like conventional programs.

① Native type

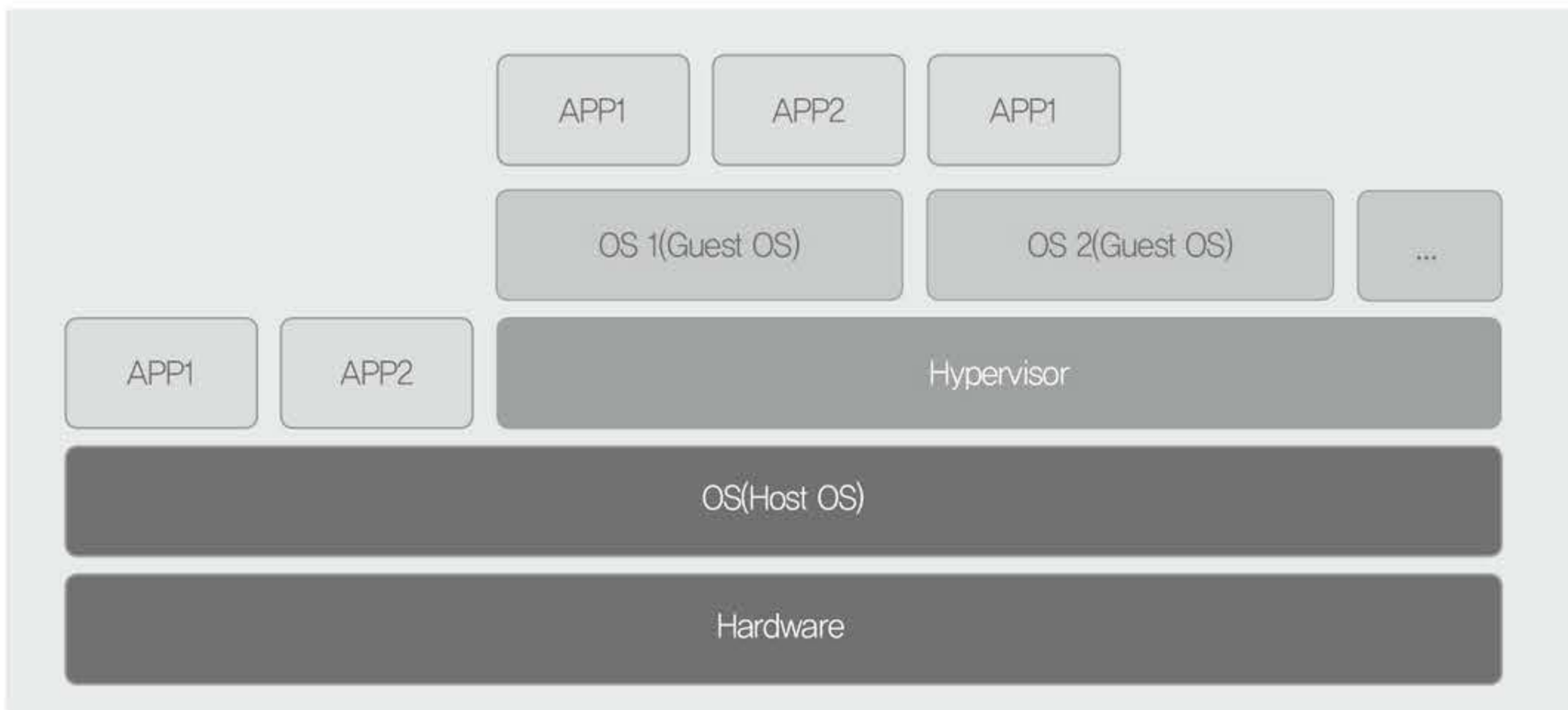
This type installs a VMM on physical hardware and requires no host OS. It saves resources, compared to the virtualization requiring a host OS, since there is no host OS. The native type is also called type1. The leading native virtualization technologies include Xen, XenServer from Citri, ESXServerBMWare, Power Hypervisor from IBM, and Hyper-V and KVM from Microsoft.



[Figure 95] Native Type Hypervisor

② Hosted type

The hosted hypervisor (see [Figure 96]) is software installed on an existing OS. The leading hosted type hypervisor includes VirtualPC from Microsoft, Workstation from VMWare, and VirtualBox from Oracle.



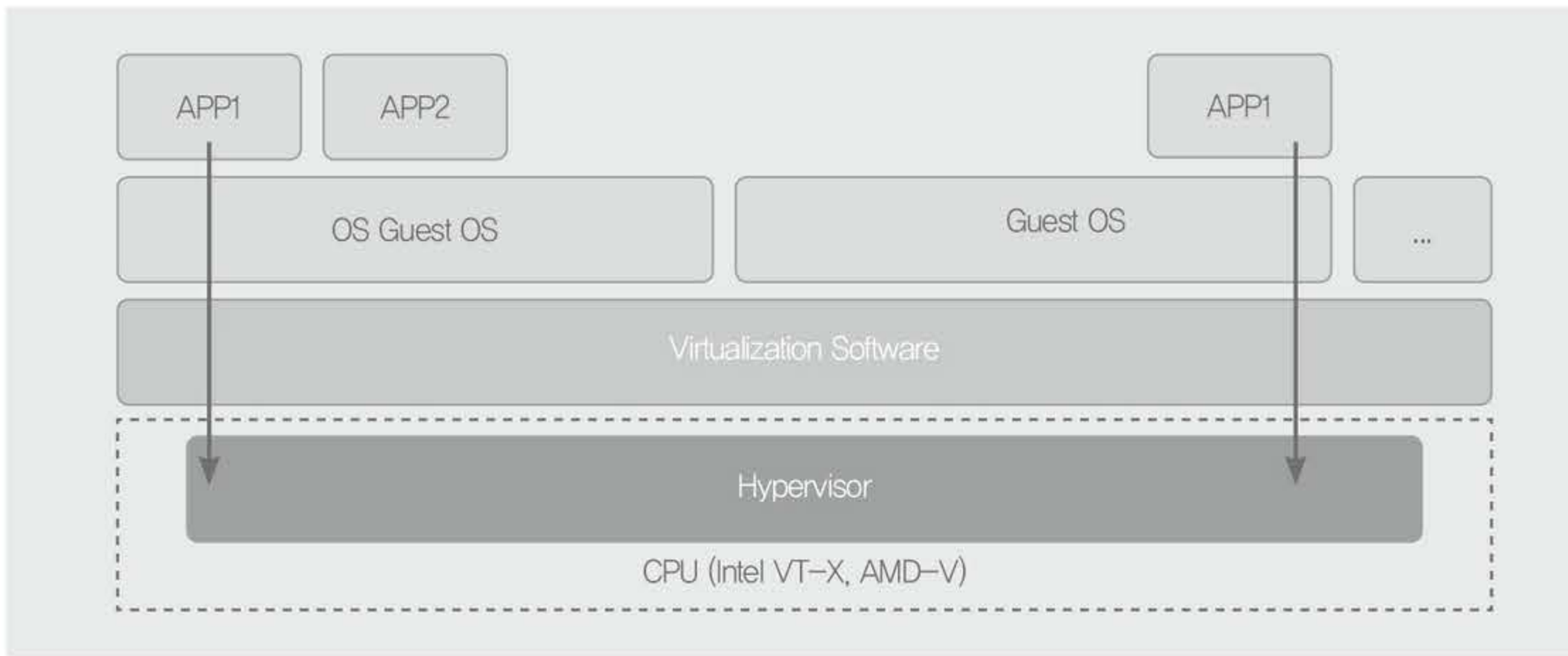
[Figure 96] Native Hypervisor

C) Server virtualization methods

According to the virtualization method, server virtualization technology can be classified into full virtualization, para-virtualization, and OS-level virtualization.

① Full virtualization

Full virtualization completely virtualizes the hardware. Although the guest OS perceives that it owns and accesses the hardware directly, the virtual server uses the resources that the hypervisor emulates to look like hardware. If the guest OS intends to monopolize the hardware, the hypervisor must handle these commands separately. A CPU (Intel's VT or AMD-V) that supports virtualization is necessary to emulate hardware resources, other than the CPU, fully. The performance has greatly increased since the hypervisor, included in the CPU, handles it at high speed. There is no need to modify the guest OS, since the CPU handles full virtualization. Therefore, there are no restrictions on the guest OS. Linux-based guest OS and Windows guest OS can operate simultaneously. The native hypervisor generally corresponds to full virtualization.

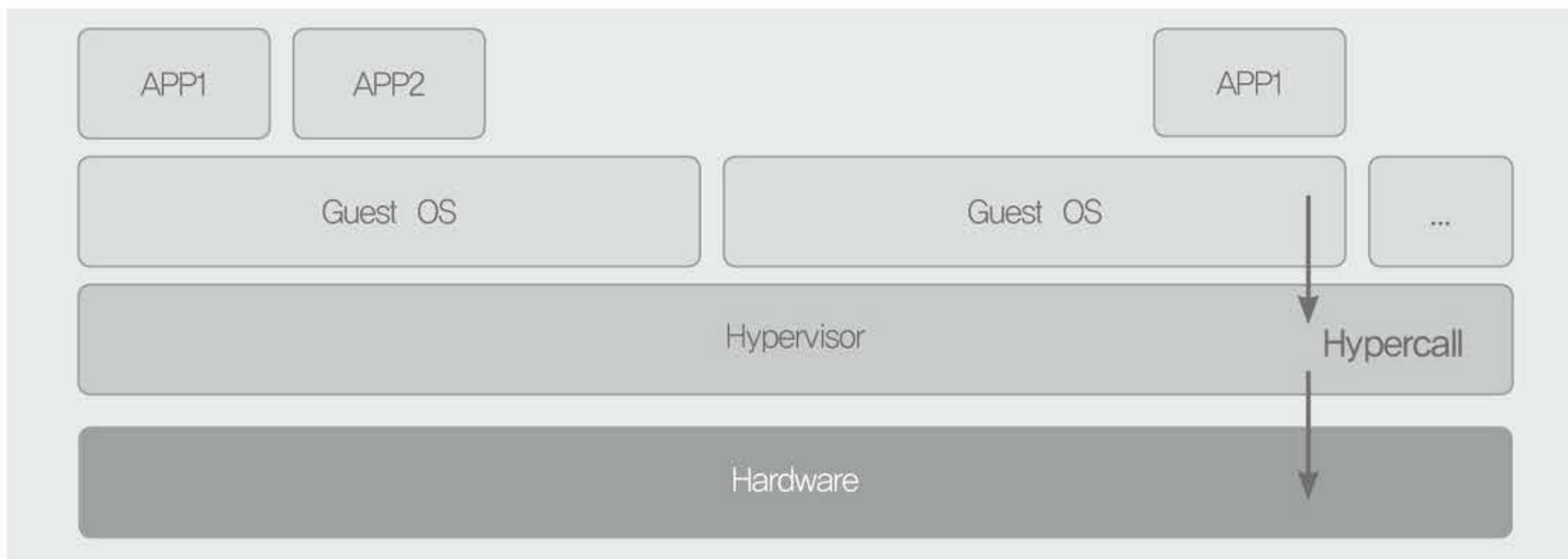


[Figure 97] Full Virtualization Conceptual Diagram

② Para-virtualization

Para-virtualization does not completely virtualize the hardware, and the guest OS must operate through the hypervisor in order to access the hardware resources. It is a method of using the hypervisor API, without hardware emulation. It is necessary to partially modify the guest OS's kernel to use the hypervisor API.

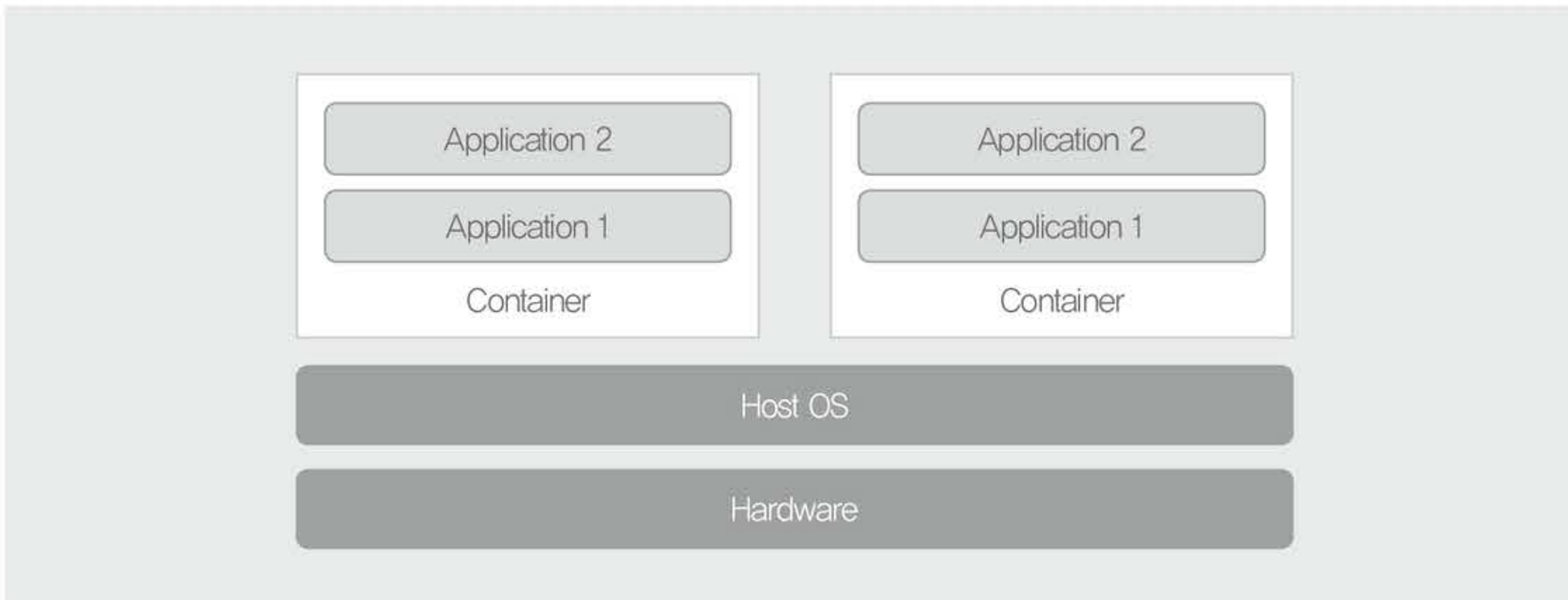
It has the advantage of providing a high performance, since it can control it through a hypervisor, without hardware emulation.It has the disadvantage that it can use only an open-source OS as a guest OS, since the kernel of the guest OS needs to be partially modified.



[Figure 98] Para-virtualization Conceptual Diagram

③ OS-level virtualization

It is a virtualization technology that makes the system look like it's using another OS on the OS. It uses the virtualization technology included in the OS, instead of a hypervisor. The container technology that allocates and isolates the resources included in the OS has gained interest. The leading OS-level virtualization technologies include Solaris Containers, FreeBSD Jails, and Linux Docker. The guest OS must be the same as the host OS, since it uses the OS's container function, instead of a hypervisor. Containers on the OS are allocated resources, such as memory, storage, and network, which are independently operated between containers.



[Figure 99] OS-Level Virtualization

Container virtualization technology can be characterized as being lighter and more portable than existing virtualization technologies. In Linux containers (LXC), there is kernel space for managing physical resources, and a user space for executing user processes inside the OS. The container virtualization technology partitions the user space into ten and limits the resources visible in each user process.

〈Table 30〉 Comparison of Hypervisor Virtualization and OS-Level Virtualization

Classification	Hypervisor Virtualization	OS-Level Virtualization
Hardware independence	Fully independent in VM	Use of host OS
OS independence	Fully independent from host OS	Host OS to be the same as the guest OS
Performance	High overhead Application of hardware virtualization to improve performance	No overhead
Management	Separately managed for each VM	Centralized management of common software
Application	Heterogeneous integration Linux and Windows mixed	Resource integration in a single OS environment
Virtualization technologies	Xen, MS Virtual Server, and KVM	Solaris, LXC, and Docker

OS-level virtualization (container virtualization) has the following advantages:

- Quick start and finish: Very fast compared to hypervisor-based virtualization
- High density: Minimized resource consumption, since an OS can operate multiple containers
- Low overhead: Userspace isolated without emulation

- Support for application container: Application-specific container configuration support

OS-level virtualization has the following disadvantages:
- Host OS-dependent
- Not possible to configure the kernel for each container

04 Storage and Network Virtualization Technologies

A) Storage virtualization

It enables service implementation by virtually allocating only the minimum space initially required, with thin-provisioning technology, instead of configuring the required storage space. Moreover, it provides an environment to use heterogeneous storage system integration.

B) Network virtualization

It implements L2, L3, and L7 switches, network firewalls, and security devices that existed in the form of hardware appliances as virtual machines. It enables networking resources to operate separately through internal virtualization, even in a shared physical environment.

05 Cloud Platform

A) Definition of cloud platform

A cloud platform is a cloud operation system to collect, control, and operate resources, such as servers, storage, networks, and virtualization technologies.

B) Openstack

Openstack is an open-source project that develops all software blocks needed to build public and private cloud computing platforms. It also refers to the community of open-source developers, enterprises, and users that want to implement and operate cloud computing with open source. Moreover, Openstack establishes industry standards for cloud computing platforms, independent of the equipment type or vendor's implementation technology.

① Main Openstack projects

- Nova: The core component for automatic control and the operation of large-scale virtual computer instances
- Swift: Implementation of large-scale, reliable cloud object storage services
- Glance: Storage, registration, management, and delivery of virtual disk image files
- Keystone: Integrated authentication and authority system
- Cinder: Volume replication, snapshot support, and block storage

- Ceilometer: Mirroring support
- HoriZone: Self-service portal
- Heat: Restoration and system recovery
- Neutron: Network service
- Trove: Delivery of database services
- Sahara: Support for Spark/data process and Hadoop

C) CloudStack

- CloudStack is the cloud computing open-source project developed by cloud.com and managed by the Apache Foundation by Citrix.

D) Kubernetes

Kubernetes is an orchestration platform for container management and operation. Google open-sourced the operation know-how that was developed for its services. Developers can manage and schedule multiple containers from a node with declarative programming. When a specific configuration is specified through the Kubernetes Control Plane, Kubernetes automatically maintains the state.

E) Mesos

Mesos is a resource management project that enables that integrated management of cloud infrastructure resources and computing engine resources. It provides a function to group distributed computing resources connected by a network into a resource pool and make it look like single computing resources. When a user requests the execution of an application program, it allocates an instance and allows the application program to run. Mesos consists of the Marathon and Chronos frameworks. Marathon allocates resources and creates jobs, while Chronos is responsible for scheduling.

X. Big Data System

Recent trends and issues

Modern society had passed the Big Data 1.0 era, when the quantitative explosion of big data began and entered the Big Data 2.0 era, which creates practical values from big data.
The introduction of Hadoop 2.2 has made big data technology more sophisticated and diversified the professional services of the big data vendors from the big data supply aspect, leading to the emergence of companies that specialize in big data. The demand for big data is rapidly increasing as large companies are quickly adopting the technology, mainly in the financial and services industries.
Companies that have adopted big data are practically using big data for business through pilot tests, beyond simply using big data for information collection. Governments are implementing policies to promote R&D and professional training, building industry-academia-R&D governance, and public data disclosure.
Of course, big data expansion has produced side effects, such as privacy infringement and personal profiling. Therefore, it is necessary to strengthen theflexibility of data use by establishing a new concept for personal data use. Comprehensive policy support that reflects the social and market environments is necessary for the big data ecosystem to evolve.

Learning objectives

1. To be able to explain the concept of big data systems.
2. To be able to explain big data system structure and characteristics.
3. To be able to discover the recent trends of big data systems.

Keywords

Data acquisition, storage, processing, analysis, presentation, Hadoop ecosystem, HDFS, MapReduce, Hadoop support program

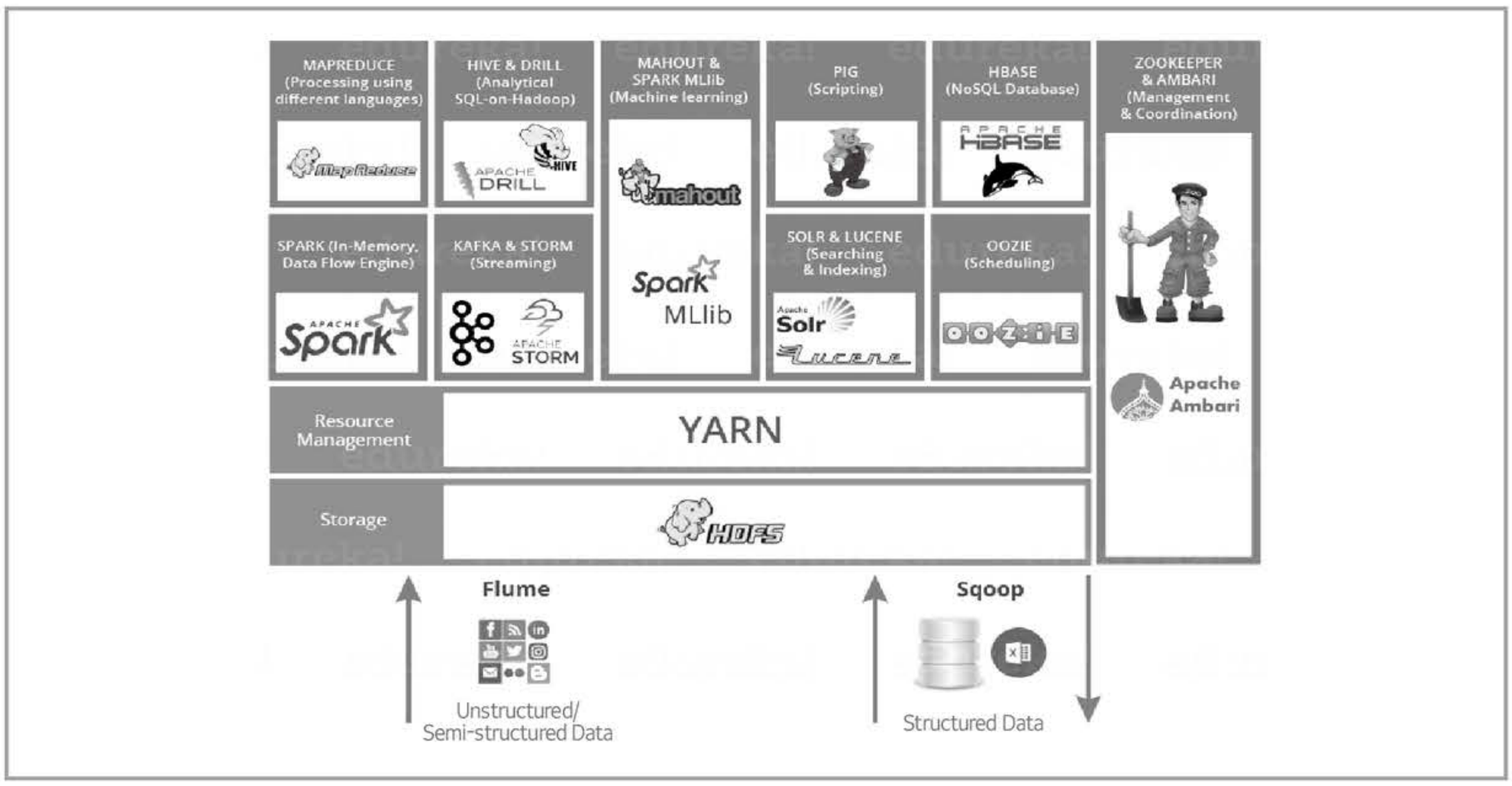
✚ Practical preview

Company A is an online retailer. Until recently, it competed for the No. 1 or 2 position in the domestic market, but has experienced declining sales, due to the influence of cheap imports from China entering the domestic online market. Therefore, the company needs a new competitive edge. Firstly, it wants to establish a new marketing strategy by collecting and analyzing customer data. It intends to expand the infrastructure in order to collect big data beyond the existing database and to hire analysts from outside to analyze the collected data. This case shows why enterprises seek to find ways to collect, store, analyze, and utilize big data. This chapter describes the big data system in more detail.

01 Bid Data System Structure

A) Hadoop ecosystem

Understanding the big data system structure begins with understanding the Hadoop ecosystem. Hadoop, the abbreviation of High-availability distributed object-oriented platform, refers to a Java-based framework for the distributed processing of a large volume of data in multiple distributed storage. It is an initial solution and consists of the Hadoop distributed file system (HDFS) and Map Reduce. However, these two modules are based on open source, and it was difficult for non-specialists to utilize big data. The modules are packaged with various peripheral modules, as shown in [Figure 100], to solve the problem. These peripheral modules support data integration and migration, application management, and system management, and they have been developed as part of the Hadoop project.



[Figure 100] Hadoop Ecosystem

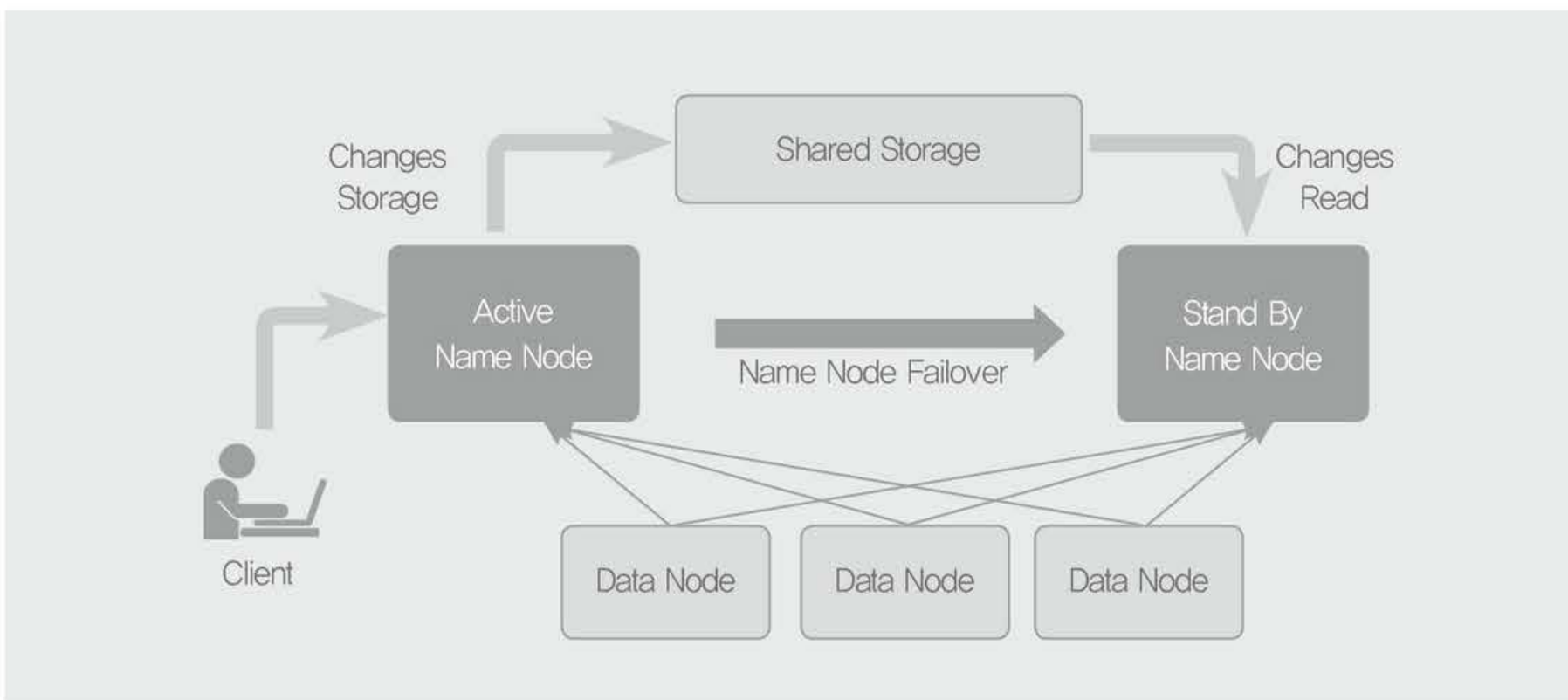
B) Hadoop's main technology elements

① HDFS

HDFS is a distributed file system to store data in the devices connected to the Hadoop network.

- HDFS prevents data loss by also storing replicated data in multiple nodes.
- It is necessary to access data through streaming in order to store files in HDFS or query the stored files.
- The data integrity is ensured as stored data cannot be modified and can be read-only. (Version 2.0 or higher allows appending to the saved file.)
- Although it does not allow data modification, it provides an interface to move, delete, and copy files.

As shown in [Figure 101], HDFS consists of one NameNode server,that serves as a master, and multiple DataNode servers, that serve as slaves. The NameNode server manages all metadata (the name of the directory where the blocks are stored, the file name, etc.) of HDFS, and the client can access the files stored in the HDFS by using it. Hadoop applications use HDFS clients to store files or to read the stored files in HDFS, and they provide the clients to users in API. The DataNode server periodically transfers a block report (information of the block stored in the node) from the NameNode, which helps the NameNode to confirm the DataNode's normal operation. Therefore, the client can log in to the NameNode to check the block's location that stores the file and directly query the data from the DataNode that stores the block.



[Figure 101] HDFS Architecture

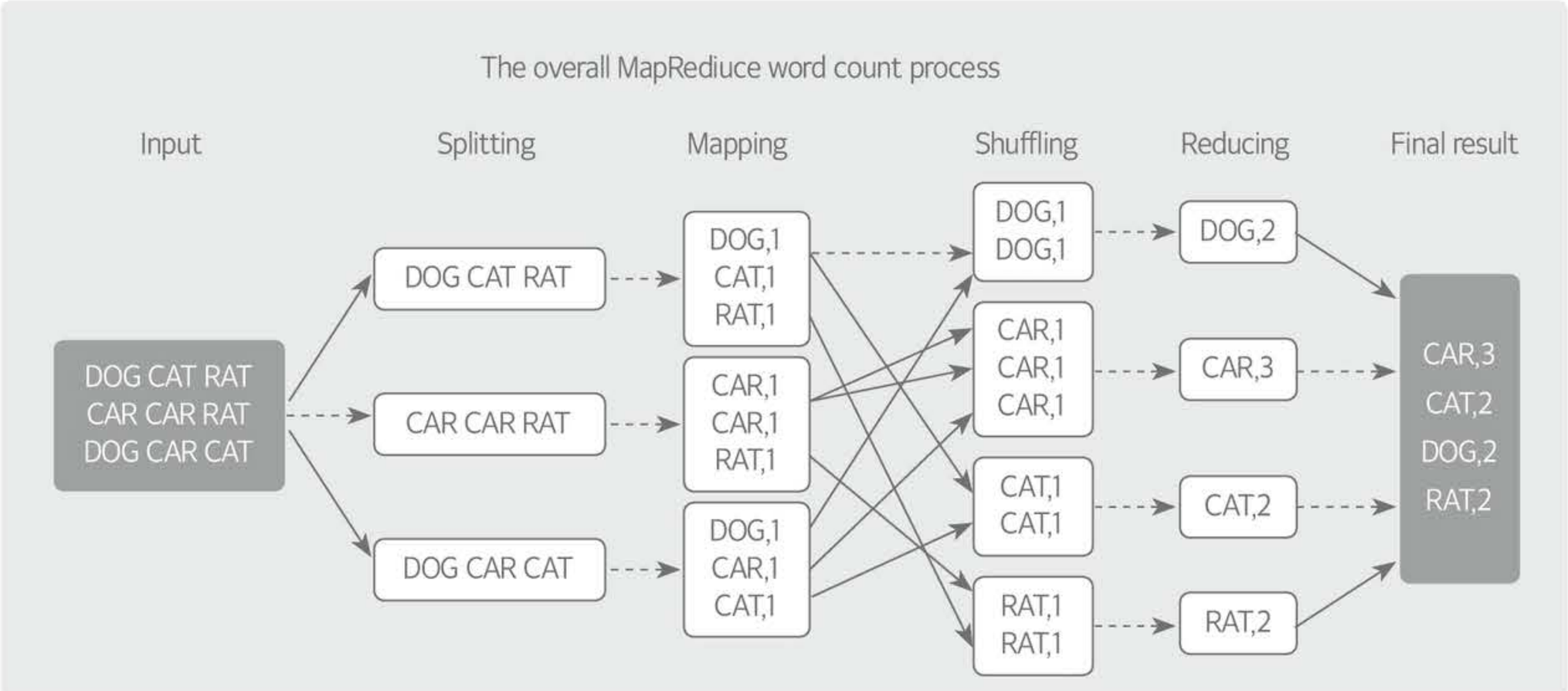
② MapReduce

MapReduce is a distributed programming model and software framework for processing a large volume of data. The MapReduce framework makes it possible to analyze data in parallel with a large-scale distributed computing environment. MapReduce consists of two methods, Map and Reduce, that the programmer writes.

- Map: Classifies scattered data into relevant data (key or value type).

- Reduce: Removes duplicate data from data output by the map and extracts the desired data.

[Figure 102] below shows an example of MapReduce outputting the frequency of words included in a character string. The process consists of dividing the string data by line (splitting) → Receiving the string by line and outputting it in the form of <Key, Value> (mapping) → Separating data by the key (shuffling) →Calulating the sum of the frequency of each key and outputting it (reducing) → Combines the output data and stores it in the Hadoop file system (final result).



[Figure 102] Process of Outputting the Frequency of Words Included in a Character String through MapReduce

C) Hadoop support program

As shown in <Table 31>, Hadoop support service programs to collect, store, utilize, process, and manage big data, are in development.

〈Table 31〉 Hadoop Support Service Programs

Big Data Type	Key Technology	Main Function
Streaming data collection	Flume	- Collection of unstructured data - Developed by Cloudera and currently included in Apache Incubation
	Scribe	- A platform to collect unstructured data - Developed by Facebook as a transfer to the centralized server
	Chuckwa	- Unstructured data collecting platform to store distributed data in HDFS
Structured data collection	Sqoop	- Structured data collection - Import of data from a relational database - Support for data transfer to various repositories, such as HDFS and NoSQL
	Hiho	- Large-capacity structured data collection and transfer solution
Distributed database	Hbase	- Distributed database - HDFS-based column-based NoSQL database developed, based on Google's paper on BigTable - Used by Yahoo and Twitter, and NHN's Line
	Cassandra	- Open-source distributed database management system - Hybrid of a Column-centered database and a row-centered database - One of the NoSQL databases

Big Data Type	Key Technology	Main Function
Real-time SQL query	Impala	- Hadoop-based real-time SQL query system - Developed by Cloudera - Use of internally developed engine, instead of MapReduce
Metadata management	HCatalog	- Big data metadata management
Data analysis	Hive	- Hadoop-based data warehousing solution - Similar SQL-based big data processing - Developed by Facebook and disclosed in opensource
	Pig	- Data analysis - To provide the proprietary language of Pig Latin, instead of MapReduce
In-memory processing	Spark	- Open-source cluster computing framework - First developed by UC Berkley's AMPLab
Data mining	Mahout	- Data mining - Hadoop-based open source
Workflow management	Oozie	- Big data processing management - Hadoop task management
Distributed coordinator	Zookeeper	- Big data server system management - Mutual coordination service between distributed environment servers
Serialization	Avro	- A framework that supports serialization of RPC (Remote Procedure Call) and data
Resource manager	YARN	- Resource management platform - Distributed computing environment - Management of computing resources inside clusters - User's application scheduling management

D) Commercial Hadoop solutions

Leading companies that develop and distribute open-source commercial Hadoop solutions include Cloudera, Hortonworks, and Map R. These companies create internal platforms, based on Hadoop, and distribute solutions through the strategic alliance with existing databases, data warehousing, and business intelligence solution companies.[9]

Commercial Hadoop version	Features
CDH, Cloudera Manager	- CDH: Includes Hadoop, Hive, OOzie, Pig, Zookeeper, and other open-source tools, but not internal products - Cloudera Manager: CDH environmental management tool (Support for CHD distribution and monitoring, free and enterprise versions). • Free edition: Includes CDH, supports up to 50-node cluster, and limited functions, other than the Hadoop infrastructure service and setting management. • Enterprise edition: Includes CDH, supports unlimited node cluster, active monitoring, and an additional data analysis tool.
Hortonworks Data Platform (HDP)	Includes Hadoop, Hive, OOzie, Pig, Zookeeper, and other open-source tools, but not internal products All software is free of charge and income is from education and support programs

9 Cloudera and HortonWorks were merged on October 3, 2018.

M3, M5, and M7	Includes Hadoop, Hive, OOzie, ZooKeeper, and other open-source tools - M3: Free version, NFS access, integrated management UI, and improved scalability - M5: Paid version (subscription), no single points of failure, mirroring, snapshots, NFS HA, and data placement control - M7: HBase improvement and improved speed, scalability, and stability

02 Big Data Dystem Trends and Forecast

A) Big data system trends

① Formation of system ecosystem, based on the strengths of each enterprise

Big data system service providers form a corporate ecosystem by promoting strategies to solve problems, such as real-time processing, memory-based processing, ease of query, and accessibility to various file systems that big data must overcome.

- Companies like Amazon, that have already secured cloud capabilities, are trying to build unique platforms by linking their cloud and open source-based big data solutions.
- IBM, Microsoft, and others, that were leaders in corporate information systems, have modified their strategies to focus on big data, utilizing their existing competency.
- Google, Apple, and other mobile-based emerging companies, are building big data analysis systems, based on existing servers and user data to provide their services.

② Differentiation of big data service delivery domain

Companies that are implementing big data systems are divided into those that provide a total solution and those that provide solutions in a specific area.

- Companies like Amazon, IBM, and Google create internal platforms that support the entire process of big data analysis and provide consulting to other enterprises, based on them generating revenue.
- Solution companies like Teradata and Good Data that implement functions, such as analysis and visualization for big data, also strive to implement unique performance.

B) Forecast of big data systems

① Still low versatility of big data systems

Many technologies related to Hadoop, the leading technology for big data, are currently being developed to the extent that it has the Hadoop ecosystem. However, it is still technically too difficult for non-engineers to use, without significant help from engineers.

② Growth of open-source big data systems

With the growth of open source-based technologies, the competitive landscape has been established to facilitate small venture companies that provide user-centered customized software, as well as traditional cloud companies, to grow in the market.

③ Strengthened linkage with AI

Starting with IBM Watson and Google AlphaGo, expectations for the industrial use of AI has risen significantly. Traditional cloud companies target consumers by linking their infrastructure and AI, and they are building newly expanded ecosystems by cooperating with competitors.

XI. Datalink Layer

▶▶▶ Recent trends and issues

Communication is established by transferring data through electrical or optical signals over physical media. In the OSI 7 hierarchy, the physical layer is at the bottom, and one can understand the mechanisms at the lower-level network by learning the media and standards for this physical layer. Moreover, it is necessary to understand the datalink layer that connects the physical media and standards with the upper layers, and to further learn the media access control method and the logical link control method.

▶▶▶ Learning objectives

1. To be able to explain the sub–layer (datalink control and media access control) of the datalink layer.
2. To be able to explain the error detection on the datalink layer and error correction techniques.

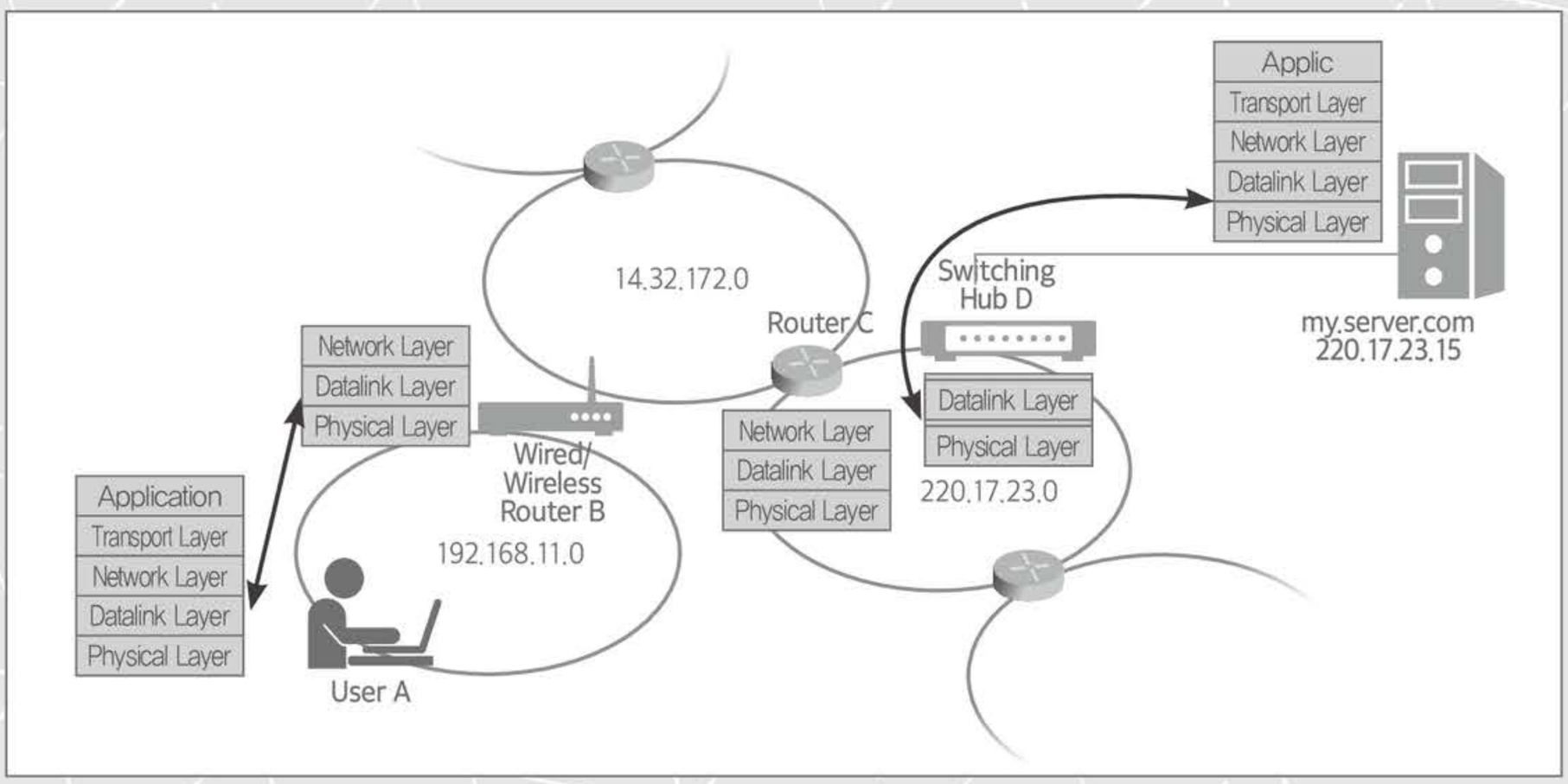
▶▶▶ Keywords

LLC, Topology, CSMA/CD, CSMA/CA, ARP, RARP, Forward error correction, Backward error correction, Turbo code, Convolution code, BCH, Hamming code, Reed-Solomon, Parity test, Checksum test, Cyclic redundancy check (CRC), Top-standby ARQ, Go-Back-N, Selective-repeat, Adaptive ARQ, H-ARQ, VRC, LRC, Optical cable, IEEE 802.11, DSSS, OFDM, MIMO, 256-QAM, FHSS, Beamforming, UHD, IEEE 802.15, WPAN, ZigBee, Bluetooth, UWB

✚ Preview for practical business Operation path of the datalink layer

[Step 3-3] An initial packet that contains the information for routing between networks is created. However, it must be further modified to be transferred between networks. The IPv4 address of user A's computer is 192.168.11.5 and cannot be transmitted to my.server.com directly. It must be transmitted to the computer's gateway wired/wireless router B (192.168.11.1). However, an IP address cannot be used for data transfer in a network, and the MAC address is needed for it. After finding the MAC address corresponding to 192.168.11.1, the gateway's IP address, the packet is transmitted to wired/wireless router B, through network broadcasting.

[Step 3-5] Upon receiving the packet, router C recognizes that the packet is sent to network 220.17.23.0, to which the router belongs, and intends to transmit it to the webserver's IP address 220.17.23.15. As is in [Step 3-3], since the packet movement in a network requires the physical address of my.server.com, the MAC address, it obtains the MAC address corresponding to my.server.com's IP address, 220.17.23.15, through the network broadcasting and transmits the packet.



[Figure 103] Operation Path of datalink Layer

01 Concept of Datalink Layer

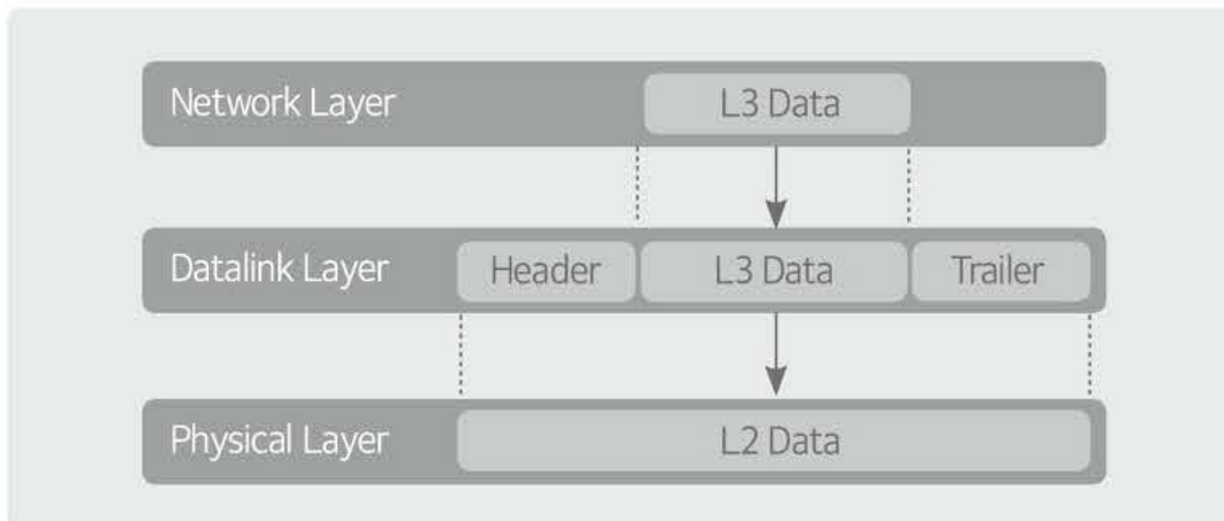
The datalink layer transmits data between peripheral devices on the network, using the physical layer that transmits signals between devices, and is responsible for address allocation and error detection. The address allocation function ensures that devices properly receive signals from the physical layer on the network, and the error detection function detects whether errors are included in the transmitted signals.

Application software developers have no problem in programming without understanding the datalink layer and the physical layer. However, a network product may need debugging for cases where there is no network connection or data error. If a product uses various medium access control (MAC) layers, it may be necessary to understand each standard protocol. Moreover, developing embedded software requires a basic understanding of drivers since it may be necessary to develop a driver. If the embedded software does not operate normally after the program is ported to the test environment prepared by the hardware developer, it is necessary to perform debugging to determine whether it is a hardware or a software problem. The developer needs to perform debugging, using JTAG, or in the case of a network environment, check whether an error occurs in the actual physically transmitted bit and datalink layer. As short-range wireless communication technologies, such as Bluetooth and ZigBee, have been recently widely applied, understanding the datalink layer has become more important.

02 Encapsulation of the Datalink Layer

A) Encapsulation of the datalink layer

As shown in the following figure, the datalink layer frame adds a header and a trailer to the network layer packet. This process is called encapsulation of the datalink layer.The process is reversed on the receiving end and is called decapsulation.



[Figure 104] Encapsulation of the Datalink Layer

B) Frame header and trailer

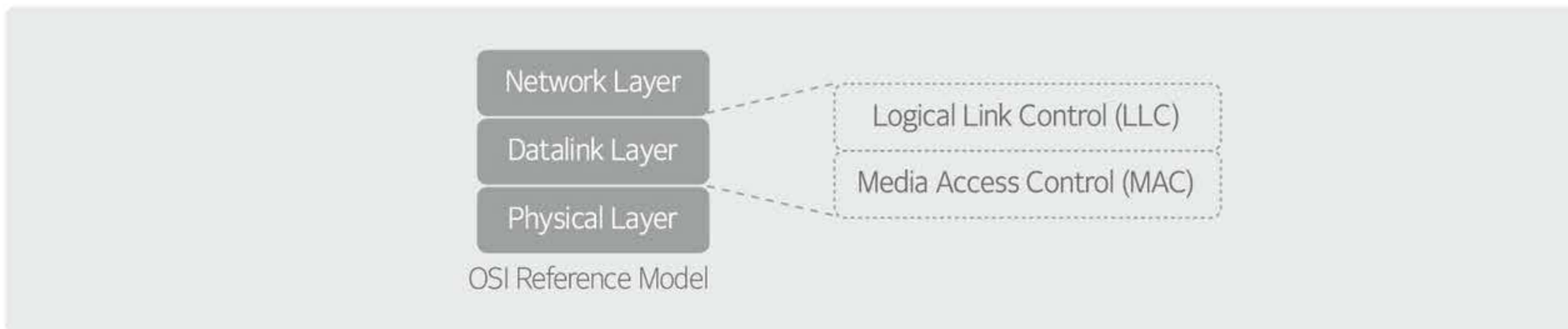
The frame header includes a preamble for bit synchronization between hosts, a start of frame delimiter (SFD) that

indicates the start of a frame, and destination and source addresses, which are physical addresses. Physical addresses have a 48-bit address system and have unique values for each device, unlike logical IP addresses. It is necessary to use the physical address to check a device location below layer 2, since IP addresses are used in layer 3. Trailers include a frame check sequence (FCS) to check for errors during a data transfer.

03 Datalink Layer Configuration

A) Sub-layers of datalink layer

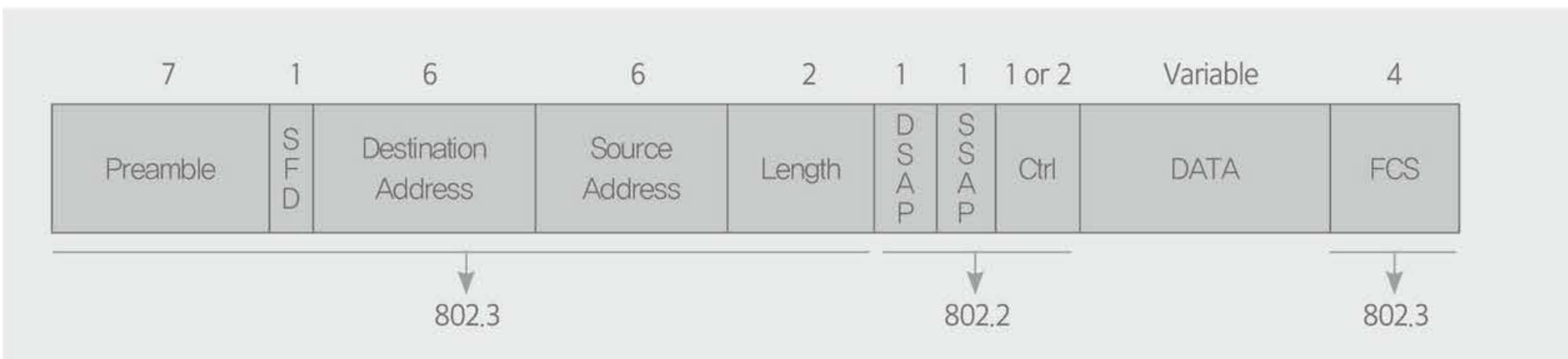
A datalink layer consists of two sub-layers: an LLC (Logical Link Control) sub-layer and a MAC (Media Access Control) sub-layer. The LLC sub-layer is responsible for the connection between the MAC sub-layer and the network layer (layer 3). The lower MAC sub-layers are responsible for control according to the topology on physical layers or other characteristics.



[Figure 105] Sub-layers of datalink layer

B) Logical link control (LLC)

① Concept of LLC



[Figure 106] IEEE 802.3 Frame Structure

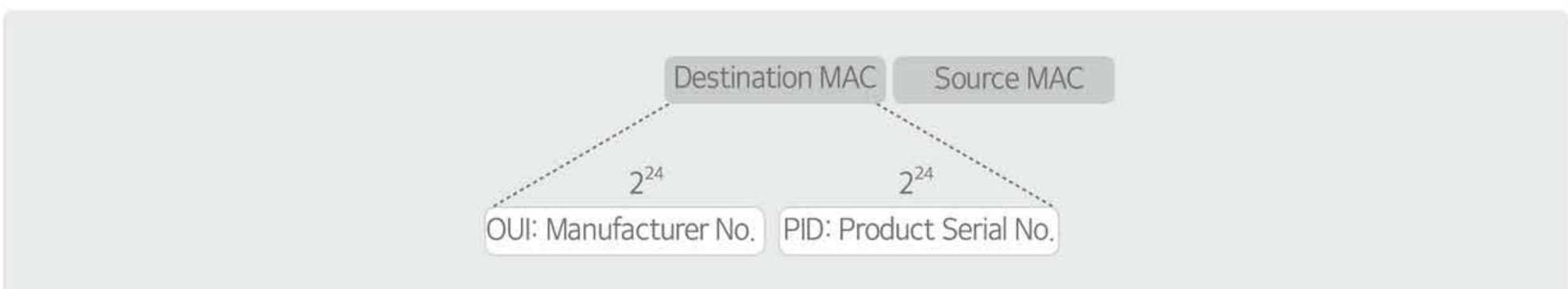
The sub-layer is located at a higher level among the sub-layers of the datalink layer. It is responsible for the data transfer between two adjacent nodes on the IEEE 802.2 or datalink layer network. The adjacent nodes have the destination service access point (DSAP) and the source service access point (SSAP) on the LLC layer. The LLC layer can use various MAC sub-layer protocols to facilitate the communication, regardless of the network topology.

② Options of the logical link layer service

The LLC options include Type 1, Type 2, and Type 3. Type 1 is an unconfirmed datagram service and called an unconnected service, as it does not require a receipt confirmation message. Type 2 is a virtual circuit access type and, as a TCP service, transfers data after establishing a virtual session. Type 3 is a confirmed datagram service and provides the receipt confirmed datagram in the point-to-point method.

C) MAC

The MAC sub-layer is responsible for determining how to transfer data through the physical media.
The MAC layer includes the MAC addresses of the transmitting and receiving systems, as shown in [Figure 107] below. The MAC address is mainly divided into two addresses. The first 224is OUI (Organizationally Unique Identifier), representing the manufacturer identification code, and the remaining 224 is the serial number of the NIC produced by the manufacturer.



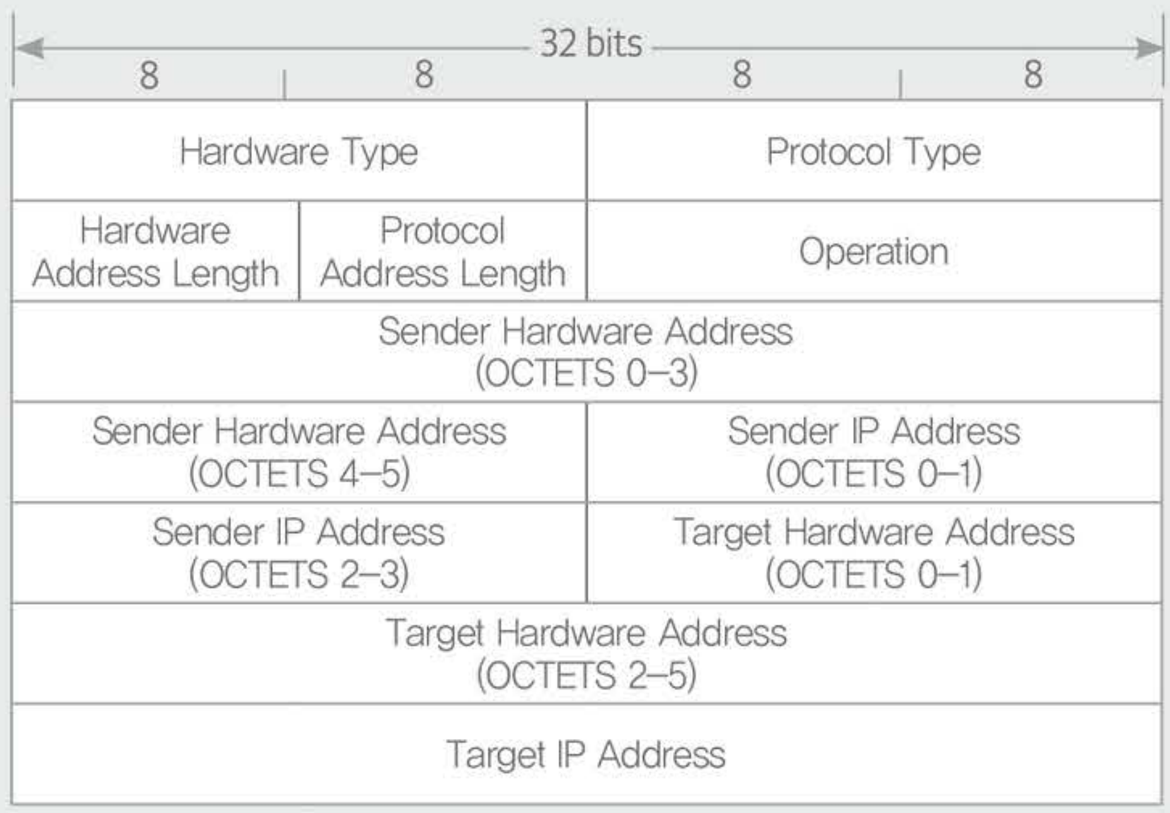
[Figure 107] MAC Address Configuration

The standardized MAC protocols include the CSMA/CD type IEEE 802.3, the token bus type IEEE 802.4, and the token ring type IEEE 802.5 for wired LAN. The CSMA/CA type IEEE 802.11 MAC sub-layer is used for wireless LAN.

04 MAC Address Search

A) IP address and MAC address conversion protocol

To transfer a packet to another host, requires the host's MAC address. ARP (Address Resolution Protocol) is responsible for it. The protocols to convert the IP and MAC addresses are the ARP and RARP (Reverse Address Resolution Protocol). The ARP that has the following address checks the MAC address through the IP address, and RARP is the inverse address conversion protocol that checks the IP address through the MAC address.



[Figure 108] ARP packet structure

The ARP packet is organized of the 6-byte destination MAC address, 6-byte source MAC address, 2-byte Ethernet protocol type, 2-byte hardware type, 2-byte protocol type, 1-byte hardware address length, 1-byte protocol address length, 2-byte operation code, 6-byte sender hardware address, 4-byte sender protocol address, 6-byte target hardware address, 4-byte target protocol address, and 18-byte padding.

B) MAC address search scenario

Since Router C and computer with user A are in the same network section, with user A, as shown in [Figure 103], the MAC address is required for communication. In that case, the computer with user A sends an ARP request through broadcasting to all systems in the network section. The data section of the frame, transferred through the ARP request, contains the ARP packet, as shown in [Figure 108]. On receiving the response, router C transfers its MAC address in the ARP packet to the computer with user A through unicast, to find each other's MAC address. The MAC address is stored in each system's cache memory. In that case, wired/wireless router B stores the port-MAC address table of the system connected to its cache memory ports.

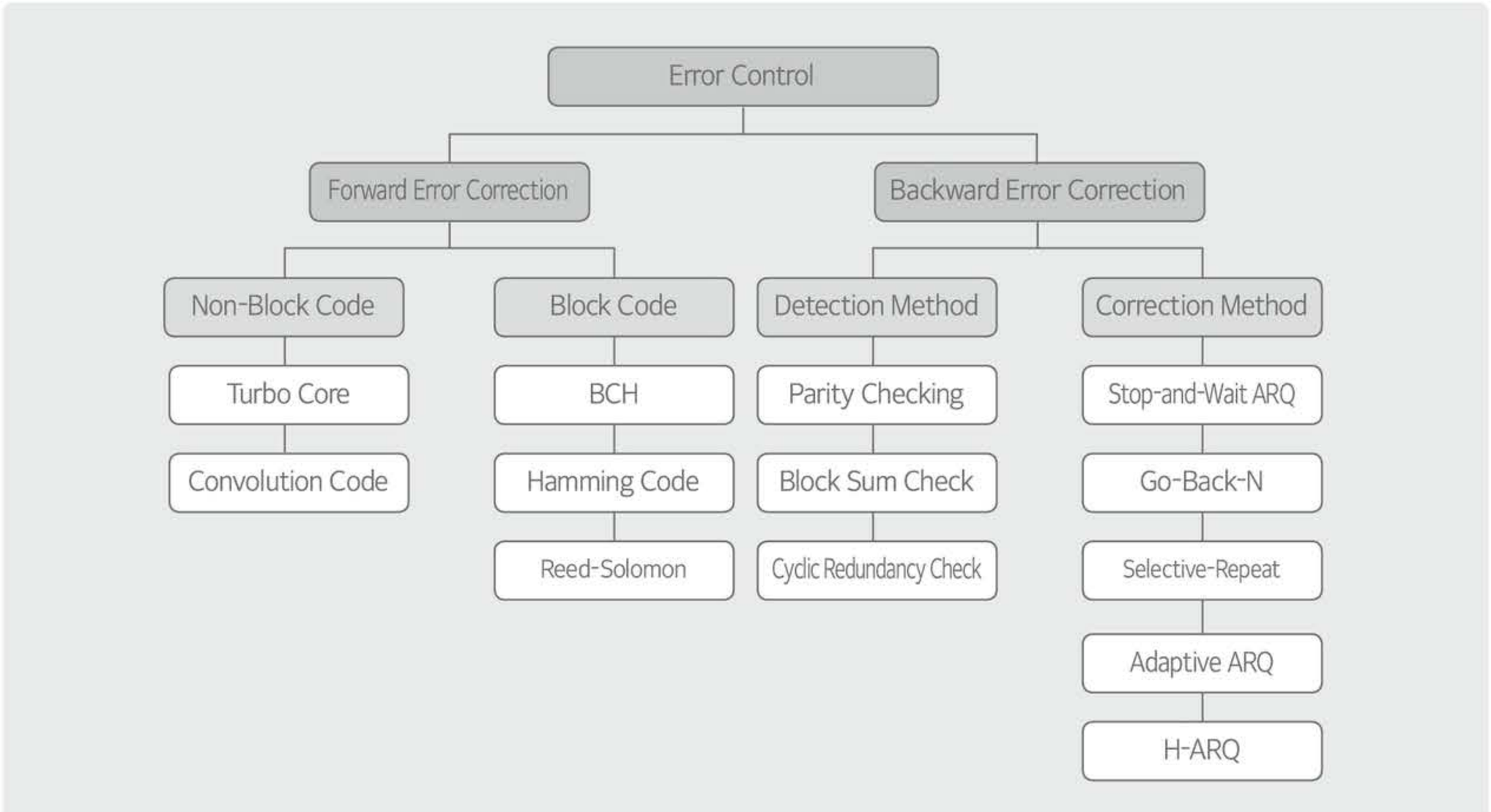
05 Techniques of Datalink Layer error Detection and Correction

A) Concept of error control

Transferring frames through the data link layer can cause various errors, depending on the network condition or the operation of the transmitting and receiving device. Types of errors include single-bit errors, in which only one bit of the data part is changed, multi-bit errors, in which two or more non-contiguous bits are changed in the data unit, and burst errors, in which two or more consecutive bits are changed.

Error control is the function of detecting and correcting errors when transmitted data does not arrive properly, or when an error occurs during transmission. Error control methods can be divided into two types, as shown in [Figure

109]. Forward error correction (FEC) allows the receiving device to detect errors and recover from them by itself, by transmitting a spare bit for error recovery in the packet. Backward error correction (BEC) notifies an error to the transmitting device to recover from the error, when an error occurs during the data transmission.



[Figure 109] Error Control Types

B) Error detection and correction

① Error detection

Error detection uses the redundancy (spare) concept by adding a spare bit to detect the destination's error. Error detection methods include VRC (Vertical Redundancy Check), LRC (Longitudinal Redundancy Check), CRC (Cyclic Redundancy Check), and checksum. <Table 32> shows details of each detection method.

〈Table 32〉 Error Detection Methods

Error detection method	Description
VRC	• Vertical Redundancy Check • Most widely used for error detection and includes the parity check
LRC	• Longitudinal Redundancy Check • A method of collecting the even parity of all bytes to create a data unit and adding it to the end of the data block
CRC	• Cyclic redundancy check • A detection method using binary division
Checksum	• A method used by higher-level protocols and based on the concept of redundancy (VRC, LRC, CRC, etc.)

② Error correction

Error correction methods include a method in which the receiving device requests the sending device to retransmit the entire data, or the receiving device automatically performs a correction using an error correction code.

〈Table 33〉 Error Correction Methods

Correction	Description
Single bit error correction	• The principle of error correction is to locate the wrong bit, which is called a parity bit. • ASCII code requires a 3-bit redundancy code.
Redundant bit error correction	• A method of checking the relationship between the number of data bits and the number of redundant bits to calculate the number of redundant bits required to correct the number of data bits given
Hamming code	• A method to find the redundant bit position in Hamming code.
Multi-bit error correction	• A method of using the redundant bit, which is calculated by overlapping a set of data bits, in order to correct the multi-bit error

③ Automatic repeat request (ARQ)

ARQ is an algorithm in which the receiving side informs the sending side of the error, and the sending side retransmits the error frame to correct the error. Typical ARQ methods include stop-and-wait ARQ, Go-back-N ARQ, and adaptive ARQ. Details of each retransmission algorithm type are shown in the <Table 34> below.

〈Table 34〉 ARQ Types

ARQ type	Description
Stop-and-wait ARQ	• The transmitting side transmits one frame to the receiving side, and the receiving side transmits an ACK or NAC signal to the transmitting side, depending on the presence of errors in the received frame.
Go-back-N ARQ	• The method continuously transmits frames to overcome the disadvantage of the transmission efficiency of the stop-and-wait ARQ. The transmitting side assigns a frame sequence number as much as the window size, and continuously transmits it to a certain unit.
Selective-repeat ARQ	• The method is similar to Go-back-N ARQ but retransmits only the frame with an error. • The transmission and receipt buffers must be large enough
Adaptive ARQ	• The method detects the error rate of the communication line and dynamically changes and transmits the most appropriate frame length
Hybrid-ARQ	• The method combines the FEC and REC. • It maintains the network efficiency, by performing the FEC normally, and improves the reliability, by performing the BEC when an error occurs.

06 IEEE 802 Standard

A) Concept of IEEE 802

[Figure 110] shows the services for each layer ad defined by the IEE Committee. The LLC layer includes, Type 3 unconfirmed datagram service, Type2 virtual circuit service, and Type 3 confirmed datagram service.

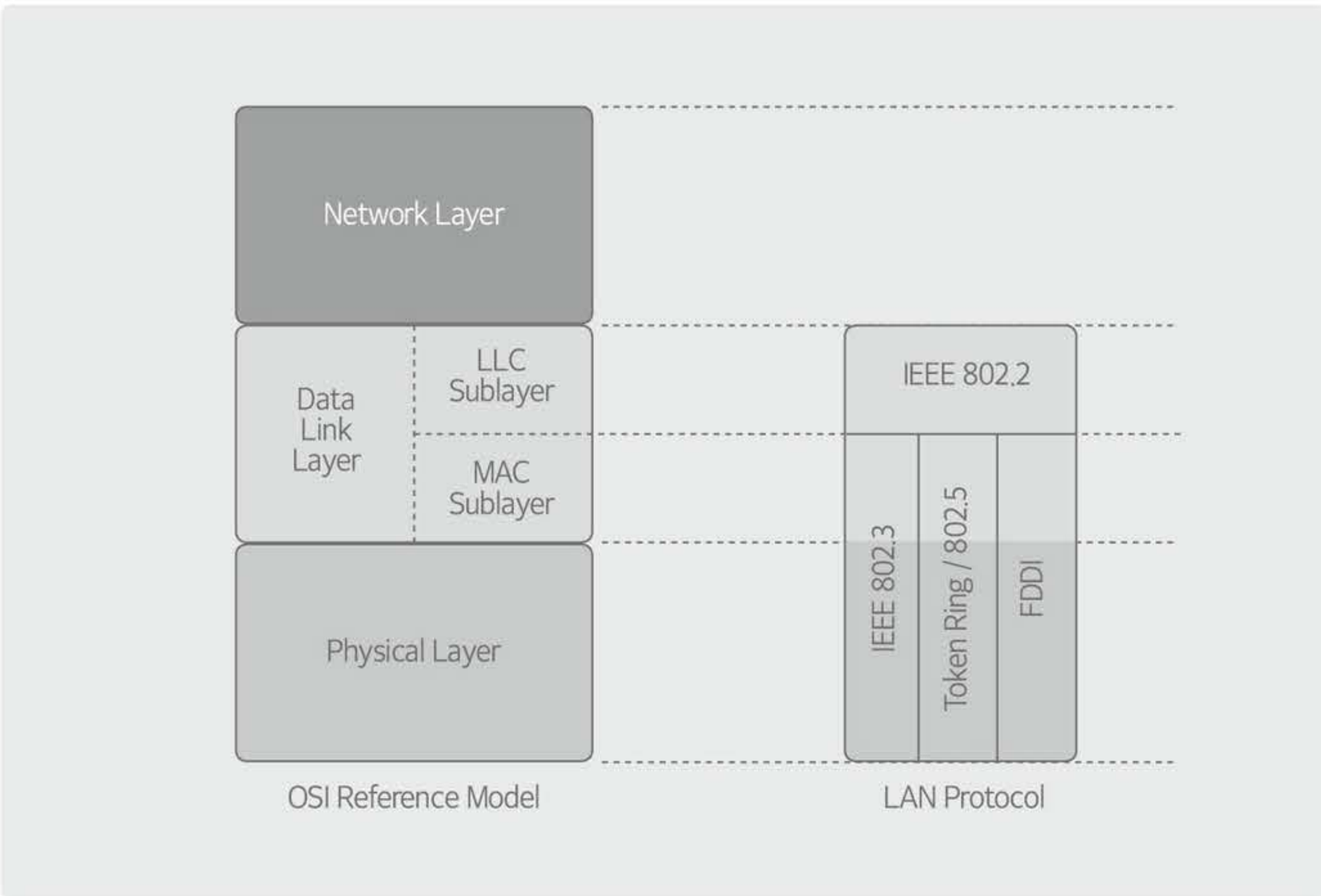
The physical medium access control (MAC) layer includes the IEEE 802.3 CDMA/CD MAC, IEEE 802.4 token bus MAC, and IEEE 802.5 token ring MAC.

Logical Link Control (LLC)	Type 1 (Unconfirmed diagram service) Type 2 (Virtual circuit service) Type 3 (Confirmed datagram service)					
Physical media access control (MAC)	802.3	CSMA/CD media access control	802.4	Token bus media access control	802.5	Token ring media access control
Physical media		• Reference band coaxial: 1 and 10Mbps • Reference band twisted pair: 10Mbps • Broadband coaxial: 100Mbps or higher		• Broadband coaxial: 1, 5, and 10Mbps Optical fiber: 5, 10, and 20Mbps		Shielded twisted pair: 1 and 4Mbps

[Figure 110] Services for Each Layer

B) IEEE 802.3 standard

The IEEE 802.3 protocol stack, the most widely used IEE 802 standard, includes IEEE 802.2 that corresponds to the LLC layer of the datalink layer and IEEE 802.3 that include the MAC layer and the physical layer, IEEE 802.5 token ring, and FDDI.



[Figure 111] IEEE 802.3 Protocol Stack

C) IEEE 802.11 standard

IEEE 802.11 is the name of the wireless communication standardization committee for wireless LAN, called Wi-Fi.

It seeks to minimize wiring work and maintenance costs to compensate for the shortcomings of wired LAN-type Ethernet. It includes IEEE 802.11b using the DSSS modulation method at the maximum transmission speed of 11 Mbps, IEEE 802.11a supporting a speed of up to 54 Mbps by using the frequency of the 5 GHz band and the OFDM modulation method, and IEEE 802.11g using the frequency of the 2.4 GHz band to support the speed of 54 Mbps. IEEE 802.11n uses the 2.4GHz and the 5GHz bands and supports MIMO (Multiple-Input, Multiple-Output) to increase the transmission speed up to 600Mbps. Wireless AP products supporting the speed of Gpbs have recently been introduced. IEEE 802.11ac supports a theoretical maximum speed of 6.93Gbps by applying the 80/160MHz bandwidth, multi-user MIMO/multi-spatial stream MIMO, 256-QAM modulation, and beamforming. IEEE 802.11ad is a standard that supports the speed of Gbps by applying various technologies in the 60GHz bandwidth. The standard for gigabit wireless LAN technology to transmitting UHD-level video and explosively increasing wireless data at high speed is in development.

〈Table 35〉 IEEE 802.11 Protocol Standards

802.11 Protocol	Frequency (GHz)	Bandwidth (MHz)	Speed per stream (Mbps)	MIMO Stream	Modulation method
-	2.4	20	1,2	1	DSSS, FHSS
a	5	20	6,9,12,18,24,36,48,54	1	OFDM
b	2.4	20	1,2,5,5,11	1	DSSS
g	2.4	20	6,9,12,18,24,36,48,54	1	OFDM, DSSS
n	2.4	20	7.2,14.4,21.7,28.9,43.3,57.8,65,72.2	4	OFDM
	5	40	15,30,45,60,90,120,135,150		
ac	5	20	87.6	8	OFDM
		40	200		
		80	433.3		
		160	866.7		

D) IEEE 802.15 standard

IEEE 802.15 is the name of the short-range wireless communication standardization committee separated from IEEE 802.11, which standardizes wireless LAN. It seeks to establish wireless networks for mobile communication devices, PCs, and other peripheral devices in the home. Sub-organizations include, the WPAN (Wireless Personal Area Network) research group, for a maximum transmission rate of 1Mbps, and WPAN HSRC (WPAN High Rate Study Group), with a maximum transmission rate of 20Mbps.

The leading WPAN technologies, corresponding to the IEEE 802.15 standard, are IEEE 802.15.1 Bluetooth, IEEE 802.15.3 UWB, and IEEE 802.15.4 ZigBee.

IEEE 802.15.1 Bluetooth is a short-range wireless communication technology standard for exchanging information by connecting mobile devices, such as mobile phones and laptops. IEEE 802.15.3 UAB is a wireless technology that quickly transmits a large volume of digital data through a large spectrum frequency at low power in an ultra-broadband and short distance. IEEE 802.15.4 ZigBee is a standard technology for home automation and data networks with low data transmission rates.

〈Table 36〉 IEEE 802.15 Protocol Standards

WPAN	Bluetooth	UWB	ZigBee
Standard	IEEE 802.15.1	IEEE 802.15.3	IEEE 802.15.4
Frequency	2.4GHz	3.1~10.6GHz	868MHz
Transmission rate	Different according to the version	480Mbps	20/40/250Kbps
Transmission distance	10~100m	10m	10~75m
Utilization	Voice and file transfer	Multimedia	Sensor communication

Embedded software developers should be able to select the chipset and the communication method to control the chipset, and to check which development tools are available, and have a broad understanding of the chipset by reviewing the specification of ZigBee and Bluetooth.
In general, the software developer should check the following in the chipset specification:

- Block diagram: Understanding of the overall configuration and flow of the chipset
- Communication method: Chipset control methods, such as SPI, I2C, etc.
- Memory: RAM, ROM, and memory size available to developers, address methods, etc.
- GPIO: Port type and setting method for external signal input and output control
- Development tool: Tools necessary to develop with the chipset

According to the application field, the hardware developer reviews the initial chipset to develop an embedded product and reviews the entire circuit configuration through the chipset. The hardware developer reviews the chipset in terms of electrical characteristics, reliability, complexity in circuit configuration, and chipset package according to device design, and cost. The software developer reviews if the chipset meets the standard, whether the control method is convenient, whether the existing control method can be reused, whether the memory capacity is adequate, whether the number of ports for I/O control required by the target product is sufficient, whether the chipset should be separated, and whether it is preferable to have multiple functions in a chipset. The software developer should select the optimal chipset after sufficient consultation with the hardware developer.

XII. Network Layer

Recent trends and issues

A network layer is the most closely related to data transmission, among the network standard layers and plays an important role in efficient data transmission. Through this chapter, you will understand how IP-based packets are generated, how they are efficiently transmitted in the Internet network, and how to move and process data over the network. This chapter describes the router's basic structure, which is a device of the network layer, the principle of packet routing, the routing algorithm, and the actual implemented routing protocol.

Learning objectives

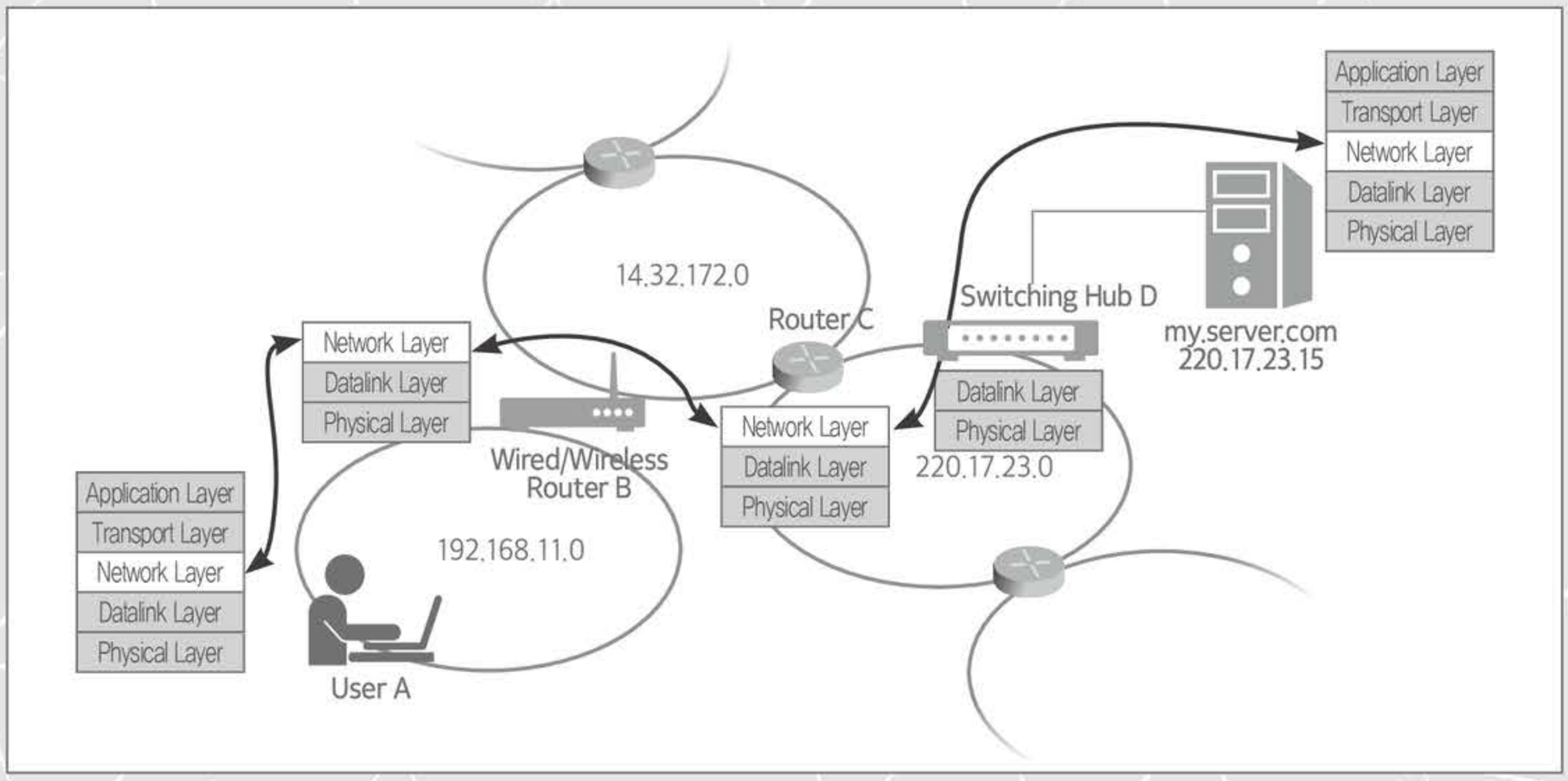
1. To be able to explain the protocol concept and equipment of the network layer.
2. To be able to explain the routing protocol concept, types, and algorithms.
3. To be able to understand the IPv4 address specification system and subnetting.

Keywords

Router, Routing table, Packet, PDU, Datagram type, Virtual line type, APIPA, Segment, Metrics, MTU, STP, ARP, RARP, ICMP, IGMP, QoS, Bandwidth, IntServ, RSVP, DiffServ, Routing algorithm, Routing protocol, EGP, IGP, Distance vector, Link state, BGP, RIP, IGRP, OSPF, IPv4, Subnetting, Supernetting, APIPA, CIDR, DHCP, NAT

✚ Preview for practical business Network layer and routing protocol

Operating path of the network layer



[Figure 112] Router Conceptual Diagram

The above sections described the network layer operation and related scenarios, as shown in [Figure 112].

[Step 3-2] Information created by the TCP protocol is capsulized in order to be routed to the 220.17.23.0 network, in which my.server.com is present, through several hops (IP).

[Step 3-3] An initial packet that contains the information for routing between networks is created. However, it must be further modified to be transferred between networks. The IPv4 address of user A's computer is 192.168.11.5 and cannot be transmitted to my.server.com directly. It must be transmitted to the computer's gateway wired/wireless router B (192.168.11.1). However, an IP address cannot be used for data transfer in a network, so the MAC address is needed. After finding the MAC address that corresponds to 192.168.11.1, the gateway's IP address, the packet is transmitted to wired/wireless router B, through network broadcasting.

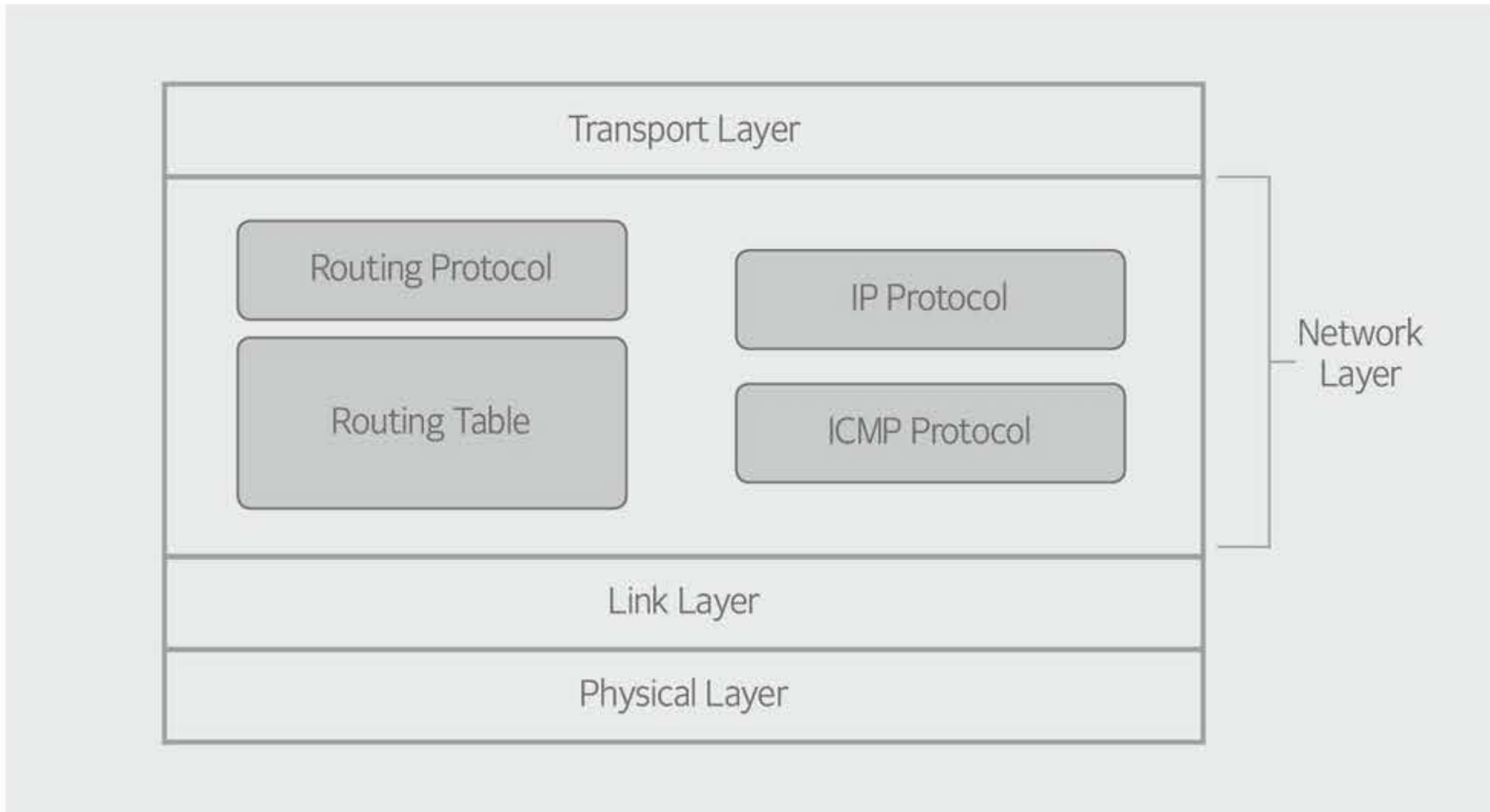
Step 3-4] Upon receiving the packet from user A's computer, wired/wireless router B, aware that the packet is not sent to the router as the final destination, examines to which address the packet must be transmitted. The IP header indicates that the packet's final destination is my.server.com, and since it does not belong to network 192.168.11.0 or 14.32.172.0, to which wired/wireless router B belongs, it transmits the packet to router C, to which it belongs.

[Step 3-6] Upon receiving the packet, my.server.com examines the IP address and recognizes that the packet was sent to it. It removes the header data, such as the IP address, and sends the remaining data up to a higher layer.

01 Network Layer Overview and Equipment

A) What is a network layer?

The network layer is the third layer of the OSI 7 model and TCP/IP and sends packets, from a transmitting side, to a receiving side. It receives a segment from the transmitting layer, encapsulates it, and sends it to the datalink layer.



[Figure 113] Network Layer in TCP/IP

B) Network layer function

The transmitting side of the network layer performs payload encapsulation, and the receiving side performs payload decapsulation. The packetizing function is responsible for ensuring that there is no change to the payload during data transmission. The routing function finds a path through which packets can be delivered. The forwarding function is performed by a router when a packet arrives through one interface on the router. When performing forwarding, the router creates a decision table by applying the routing rules. The decision table is also called the forwarding or routing table.

〈Table 37〉 Network Layer Functions

Function	Description
Packetizing	- The transmitting side encapsulates the payload, and the receiving side decapsulates the payload. - The function is responsible for transmitting a payload, from the transmitting side, to the receiving side, without a change.
Routing	- The function finds the path to transfer the payload, from the transmitting side, to the receiving side.
Forwarding	- The router performs the function when a packet has arrived through an interface. - The router applies the routing rules to create a decision table. ※ The decision table is also called the routing table.

C) Internetworking equipment

Internetworking refers to a connection between networks. The leading internetworking equipment includes repeaters, bridges, switches, and routers. .

〈Table 38〉 Internetworking Equipment

Equipment	Description
Repeater	A device that enhances (amplifies or reproduces) the signals between AN connection points
Bridge	A device that connects two LANs and makes them look like one, through interpretation and format conversion
Switch	A device that has the concept of a multi-port bridge to separate the network, based on the MAC address
Router	A device that finds the optimal communication path when connecting to a heterogeneous network

This section describes for which of the OSI 7 layers and TCP/IP 4 layers these internetworking devices are responsible. The router performs packet routing through the IP address on the network layer. The bridge and switch are responsible for frame transfer on the datalink layer, while the hub and repeater transfer physical signals on the physical layer. <Table 39> compares the function of each device with the protocol layer.

〈Table 39〉 Protocol Layers and Device for Each Layer

OSI 7 Layers	TCP/IP 4 Layers	Equipment
Application	Application	Gateway
Presentation		
Session		
Transport	Transport	
Network	Internet	Router
Data Link	Network Access	Bridge, Switch
Physical		Hub, Repeater

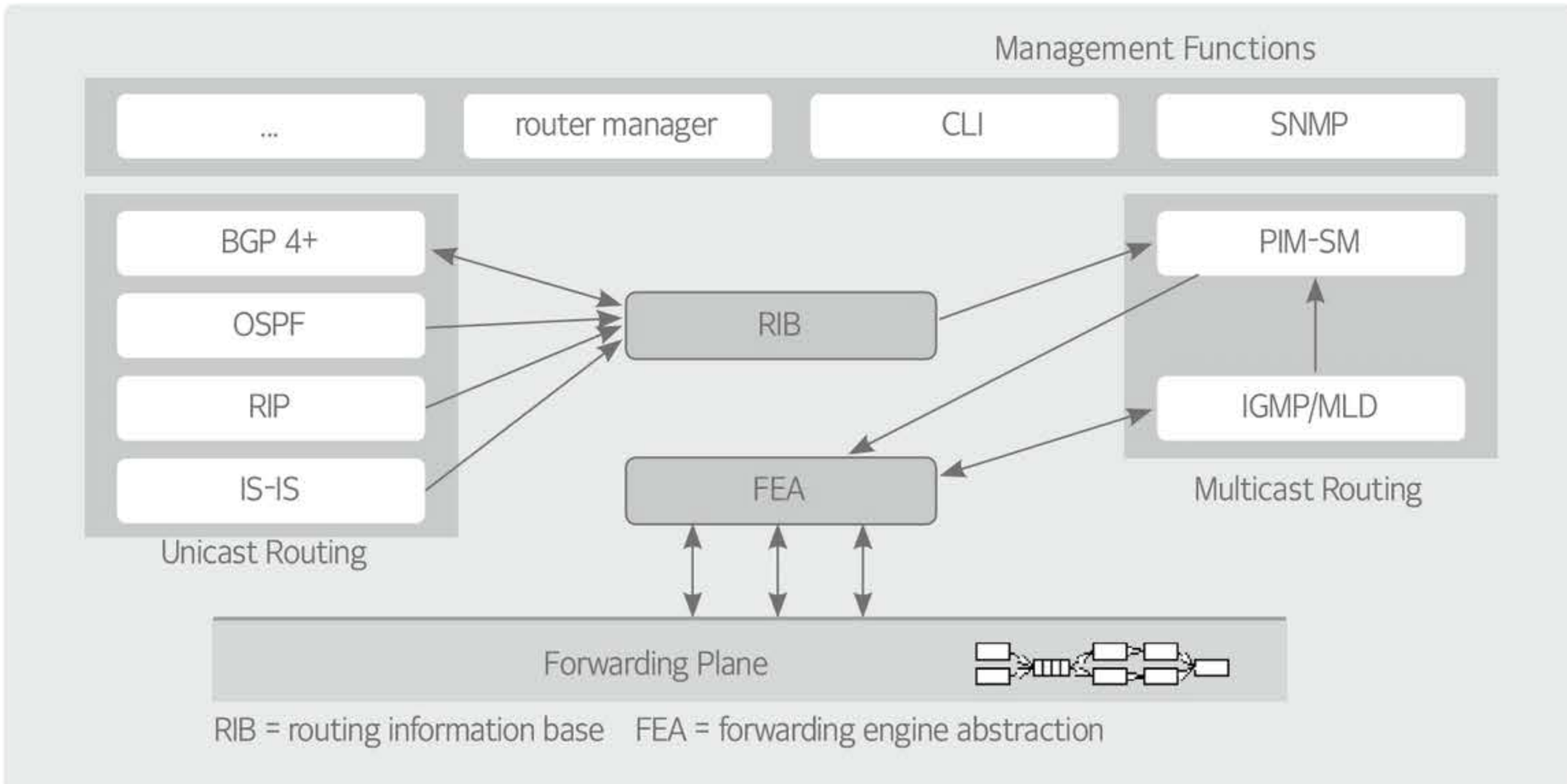
D) What is a router?

A router forwards traffic and determines the optimal path, using one or more metrics. In other words, it forwards packets from one network, to another, based on the network layer information.

.

① Router structure

As shown in [Figure 114], a router mostly consists of a router control plane and a forwarding plane. The control plane, implemented in the software in a router, determines where to send the packets it receives and uses tables to decide. The forwarding plane actually sends the packets according to the requirements created by the control plane.



[Figure 114] Router Conceptual Diagram

<Table 39> shows the shapes of a router. Small networks mostly use cisco 2500, while larger networks use cisco 7300.

〈Table 39〉 Router Shape [2]

Cisco 7300	Cisco 2500

② Router metrics

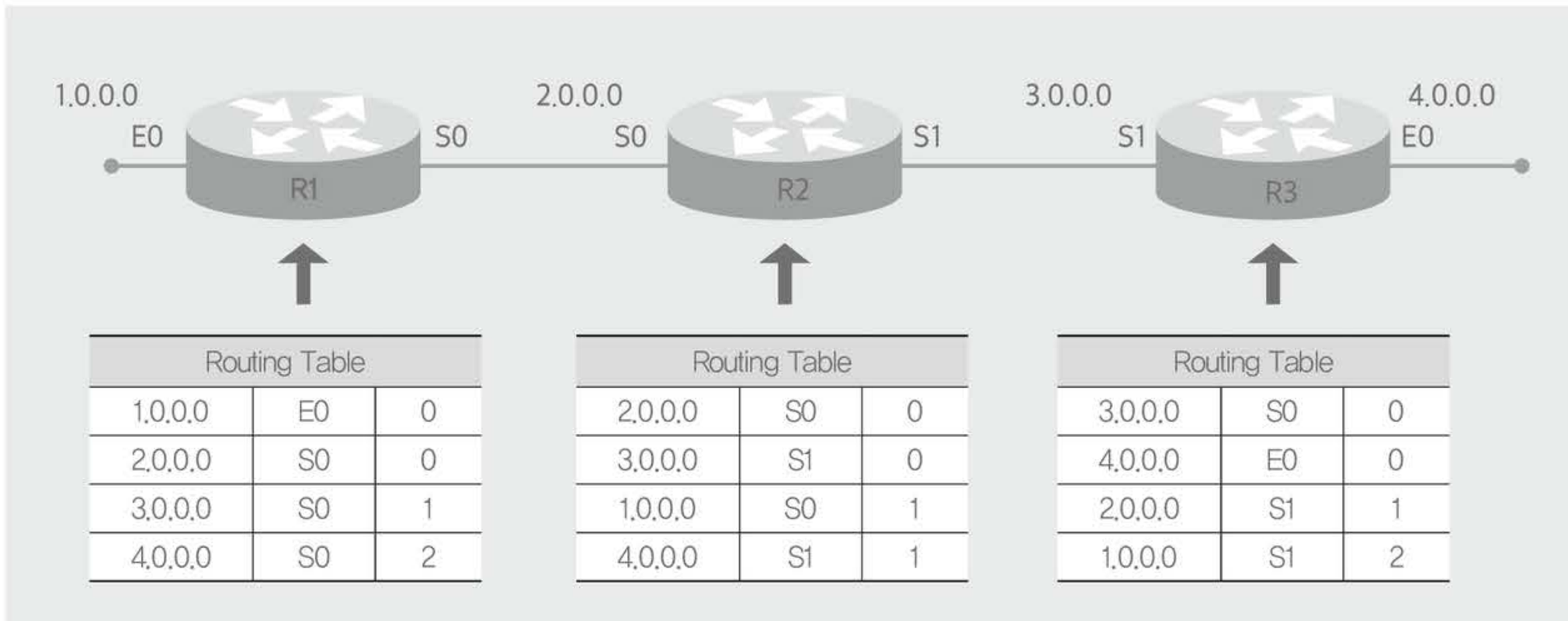
It refers to the data collected by a router at a given time, when the traffic passes through the path. <Table 40> shows the metric types.

〈Table 40〉 Routing Metrics [3]

Metric	Description
Number of hops	It refers to the number of routers to pass through in order to reach the destination. The processing speed is higher when there are fewer routers.
MTU	Maximum Transmission Unit. It refers to the maximum data that the protocol can transfer.
Cost	The cost according to the transfer time, link reliability, and band characteristics is determined. A higher cost means lower efficiency.
Latency	The packet delay records and bottlenecks during the transfer between routers are determined.

③ Routing table

A routing table is a database that stores the output interface of each destination network and the next hop IP address.



[Figure 115] Routing Table

E) What is a switch?

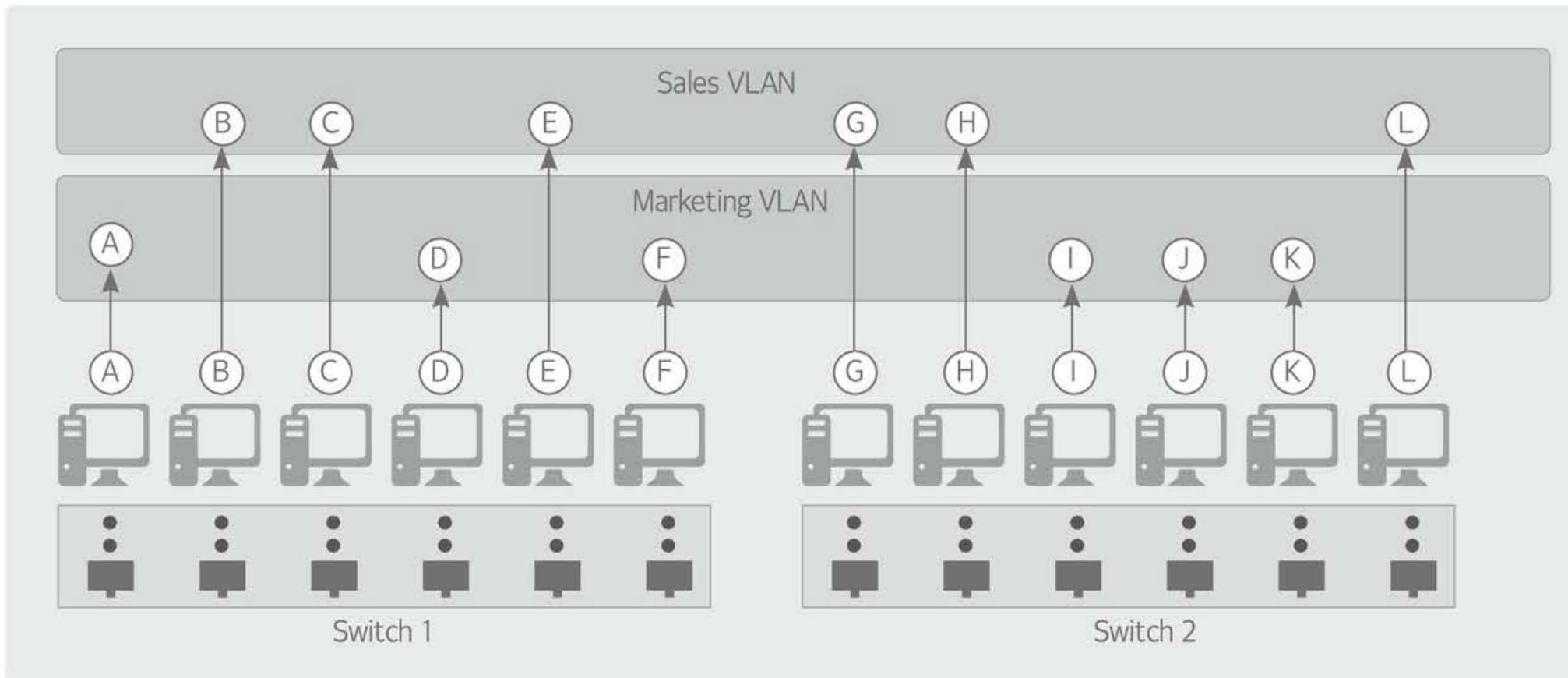
A switch is a device that connects network units. A switch device has an important role in MAC address learning, filtering, and loop prevention. Other switch functions include port mirroring and VLAN. Port mirroring, also called port monitoring, is the function of sending a replicate of the data sent from a specific port to a mirrored port. It is an important function used by administrators to monitor network traffic through the mirrored port.

〈Table 41〉 Main Switch Functions

Function	Description
Address learning	- Learning the MAC address of all systems that are connected to a port
Filtering	- Filtering data traffic to a specific port using the MAC address
Loop prevention	- Preventing a network loop when multiple paths to a destination are generated - The system usesa spanning tree protocol (STP) to prevent the network loop.

F) Virtual local area network (VLAN)

VLAN is a method of configuring a logical network, desired by the user, to overcome the physical and regional constraints of the existing networks. In other words, it uses the IP unit, protocol unit, MAC address, and connecting port to configure a logical network, instead of separating networks by spatial and geographical locations. The VLAN protocols of a switch include ISL, 802.1Q, and VTP. ISL and 802.1Q are the VLAN tagging protocols, and VTP is a VLAN management protocol.



[Figure 116] VLAN Configuration [4]

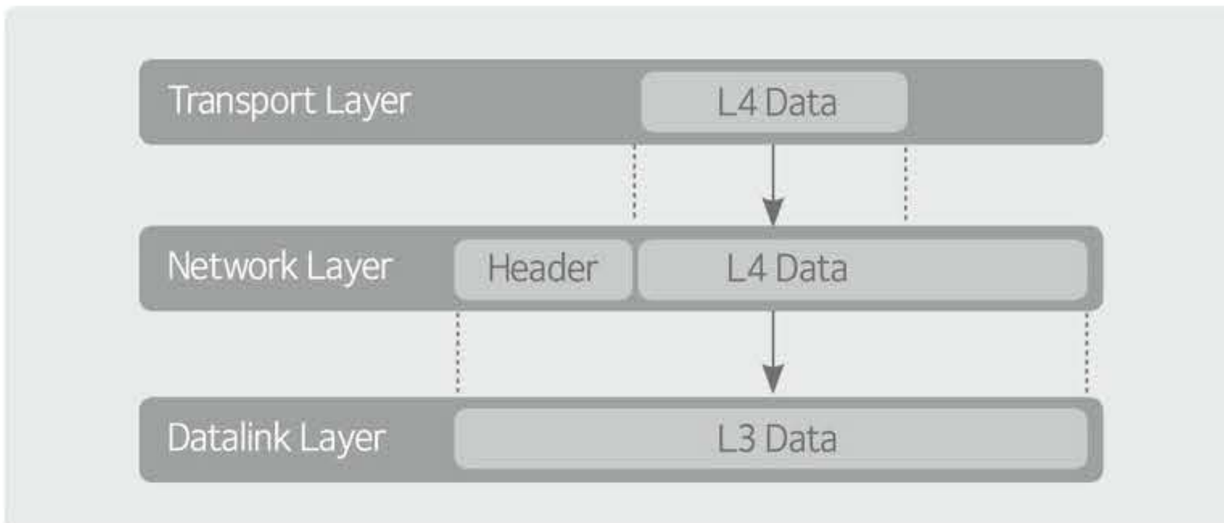
〈Table 42〉 Comparison of Switch and Router

Compared Items	Switch	Router
Reference table	MAC address table	Routing table
Reference PDU	Ethernet frame	IP packet
Reference field	Destination MAC address	Destination IP address
Frame used	Ethernet	Ethernet, frame relay, PPP, etc.
2-Layer header	No change	Replaced by a new header

02 Network Layer Encapsulation

A) Network layer encapsulation

As shown in [Figure 117], the packets on the network layer are configured by adding a header to the transmitting layer segment. This process is called encapsulation of the datalink layer, and the process is reversed on the receiving end and is called decapsulation.



[Figure 117] Network Layer Encapsulation

B) IPv4 header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

[Figure 118] Network Packet Structure [5]

In [Figure 118] above, "Version" refers to the protocol version of the datagram, "IHL" refers to the header length, and "Type of Service" refers to which type of service the host wants in the subnet. "Total length" refers to the sum of the header and data lengths in the datagram, "Time to Live" refers to the packet life limit, and "Protocol" refers to the notification of TCP or UDP for the recombination of the datagram on the network layer. "Source Address" and "Destination Address" refer to the source and destination addresses of the 32-bit datagram.

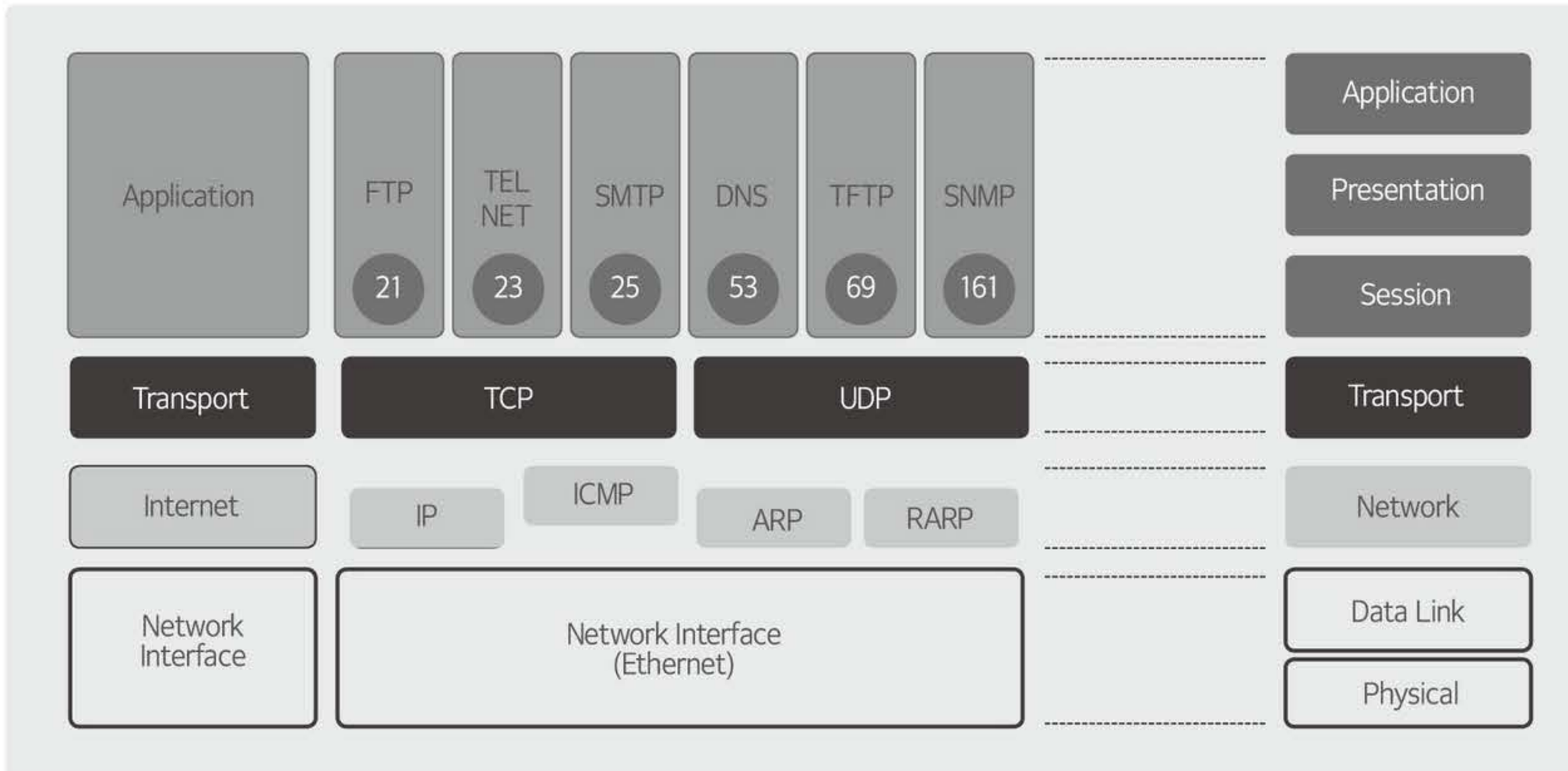
03 Packet Switching and Network Layer Protocol/Instruction

A) Packet switching

The methods of finding the packet path in a packet-exchanging network, include the datagram approach and the virtual circuit approach. The datagram approach is a non-connected service that independently processes all packets. Packets can be transferred through different paths, and the packet's destination address determines the forwarding path. The virtual circuit approach is a connection-oriented service. It establishes a virtual path for the datagram before the transfer, and all datagrams are transmitted to the same path after the connection. The forwarding path is determined by the packet label (virtual circuit identifier).

B) Network layer protocol/command

The network layer protocols that convert the IP and MAC addressesinto an IP address, are the ARP (Address Resolution Protocol) and the RARP (Reverse Address Resolution Protocol). ICMP (Internet Control Message Protocol) is a protocol used to transfer the network error information, and IGMP (Internet Group Management Protocol) is the protocol for IP multicast transfer.



[Figure 119] Network Layer Protocols

〈Table 43〉 Network Layer Protocols

Protocol	Description
ARP	- Address Resolution Protocol(RFC 826) - Conversion of an IP address into a MAC address
RARP	- Reverse Address Resolution Protocol(RFC 903) - Conversion of a MAC address into an IP address
ICMP	- Internet Control Message Protocol(RFC 792) - Transfer of network error information
IGMP	- Internet Group Management Protocol(RFC 1112) - IP multicasting

C) Network layer command

<Table 44> shows commands to check the network layer status or perform a specific action.

〈Table 44〉 Network Layer Commands

Number of Commands	Description
Ping	• A command that uses an ICMP message to test the connectivity to the network layer. • It uses the echo request and replies to messages from the ICMP message.
Tracert/ Traceroute	• A command to find the path to the desired destination. • It checks which network section caused the bottleneck, if there is a problem in connecting a specific site.
Route	• A command to allow manual modification of the routing table.
Ipconfig/ Ifconfig	• A command to check the computer's TCP/IP network settings. • It is used to check and update the DHCP and DNS settings.
Netstat	• A command to check the network connection, routing table, and network interface status.
Arp	• A command that shows or changes the local ARP cache.

04 Network Service Quality

A) Quality of Service (QoS) characteristics

QoS is the technology of guaranteeing a service quality and performance of a communication service to meet user requirements. The attributes to determine the QoS include confidence, delay, jitter, and bandwidth, as shown in <Table 45>.

〈Table 45〉 QoS Measurements

Attribute	Description
Confidence	- An attribute needed by a flow in order to deliver a packet safely to the destination
Delay	- A delay when transmitting a packet from the transmitting side to the receiving side
Jitter	- Variation of delay for packets belonging to the same flow
Bandwidth	- The maximum rate available in communication, i.e., the ability to transmit information

B) Quality implementation technique

① Scheduling

Routers are responsible for handling packets over the Internet. Therefore, the packet flow can differ, according to the schedule.

〈Table 46〉 Packet Scheduling Methods

Scheduling Method	Description
FIFO queuing	- The basic Internet scheduling method delivers packets in the order of their arrival.
Priority queuing	- The method assigns a priority to each transferred packet and handles the packet with higher priority first.
Weighted fair queuing	- The method assigns a priority class to packets, and packets are assigned to each priority queue. - It selects a queue in the form of a round-robin, and the packets in the selected queue are delivered. The number of packets delivered varies, according to the weight of the queue.

② Traffic shaping and traffic policing

As a way to control the traffic speed and volume, traffic shaping is used when the traffic leaves the network, and traffic policing is used when the traffic enters the network.

〈Table 47〉 Traffic Shaping and Traffic Policing

Traffic Shaping/Policing	Description
Leaky bucket	- The burst input packets are stored in a bucket and output at an average speed. - The packets can be discarded if the input packets exceed the bucket size.
Token bucket	- The basic algorithm used by traffic shaping and policing

③ Resource reservation

It is the method of reserving resources, such as buffer, bandwidth, CPU, and time needed for data flow,for a specific service to ensure the service quality.

④ Admission control

It is a procedure used by the control unit of a communication network to determine whether to accept a connection request.

⑤ Service quality model and protocol

Service quality models include the integrated service (IntServ) model and the differentiated service (DiffServ) model. The service quality protocol is the resource reservation protocol (RSVP).

〈Table 48〉 Service Quality Model and Protocol [6]

Service quality model	Description
Integrated service (IntServ)	- Flow-based quality model that explicitly reserves resources, such as bandwidth for the given data flow
Resource reservation Protocol (RSVP)	- A protocol standardized to reserve and secure the bandwidth needed by the application programs between terminals (RFC 2110)
Differentiated service (DiffServ)	- Class-based service quality model that specifies the required service type each time a packet is transmitted, and operates according to packet priority

05 Routing Protocol Algorithm and Type

A) What is a routing protocol?

Routing protocol defines the types of messages exchanged between routers, exchange procedure, and actions when receiving a message, for efficient setting and updating of the routing tables.

B) What is a routing algorithm?

The routing algorithm finds the low-cost path between a source router and a destination router in a graphical network, connected by links with a cost between routers. The low-cost path refers to the path of minimum value among the sum of link costs out of all possible paths from the source to the destination. <Table 49> shows the types of routing algorithms.

〈Table 49〉 Routing Algorithm Types

Algorithm	Description
Link state Routing	• Link state routing algorithm • Each router calculates the low-cost path to all destinations with the configuration and link-state information of the entire network.
Distance vector Routing	• Distance vector routing algorithm • Each router maintains the low-cost path table (vector) to all destination routers. • The initial router path cost table only maintains the cost data of the directly connected routers, and the router notifies the path cost data to the neighboring routers.

① Dijkstra algorithm

[Figure 120] shows the pseudo-code of the Dijkstra Algorithm, the leading link-state routing algorithm.

```
Dijkstra's Algorithm()
{
  // Initialization
  Tree = {root}              //Tree is made only of the root
  for (y=1 to N)             // N is the number of nodes
{
    if (y is the root)
     D[y] = 0                // D[y] is shortest distance from root to node y
    else if (y is a neighbor)
     D[y] = c[root][y]       // c[x][y] is cost between nodes x and y in LSDB
    else
     D[y] = ∞
  }
  // Calculation
  repeat
{
    find a node w, with D[w]
    Tree = Tree U {w}        // Add w to tree
   // Update distances for all neighbors of w
    for (every node x, which is a neighbor of w and not in the Tree)
  {
    D[x] = min{D[x], (D[w] + c[w][x]}
  }
 } until (all nodes included in the Tree)
} //End of Dijkstra
```

[Figure 120] Pseudo Code of Dijkstra's Algorithm

② Bellman-Ford Algorithm

The Bellman-Ford Algorithm is the leading distance vector routing algorithm.

The shortest path can be decomposed as follows, by expanding the optimal substructure of the shortest path problem. The shortest path from the starting node s to v, is the sum of the shortest path from s to u, and the weight factor (distance) between u and v.

$$D(s, v) = D(s, u) + w(u, v)$$

The Bellman-Ford Algorithm performs edge relaxation of all edges in the graph, when calculating the shortest path between s and u. Then, how many times should it be performed? The shortest path between s and u may be only s

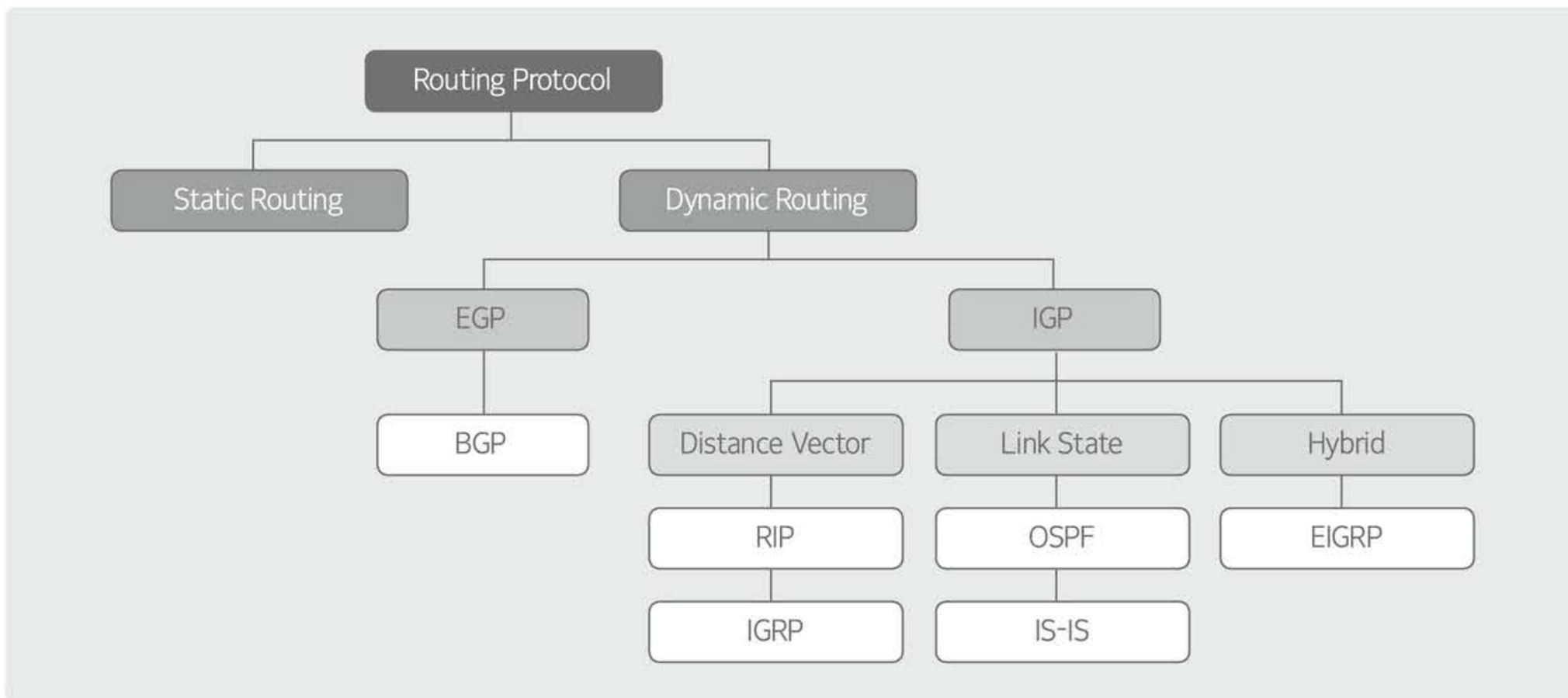
and u, and all (|V|—1) nodes of the graph, excluding u, can be the shortest distance between s and u. Therefore, the Bellman-Ford Algorithm performs the edge-relaxation |V|—1 times for all edges.
[Figure 121] shows Bellman-Ford Algorithm's pseudo-code.

```
BELLMAN-FORD (G, w, s)
INIT-SINGLE-SOURCE (G, s)
for i = 1 to |G.V| -1
    for each edge (u,v) ∈ G.E
        Relax (u, v, w)
for each edge (u, v)∈ G.E
    if v. d > u.d + w(u, v)
        return FALSE
return TRUE
```

[Figure 121] Bellman-Ford Algorithm's Pseudo-code

C) Routing protocol types

The routing protocol is classified into static routing and dynamic routing, depending on the routing type, as shown in [Figure 122]. The dynamic routing method is divided into the EGP (Exterior Gateway Protocol) method and the IGP (Interior Gateway Protocol) method, according to the classification of AS (Autonomous System). The IGP (Interior Gateway Protocol) method is further divided into the distance vector method, the link-state method, and the hybrid method, according to the routing setting method.



[Figure 122] Routing Protocol Classification

The type of routing protocol can be divided into an internal router protocol and an external router protocol, based on the autonomous system (AS). Here, AS refers to networks with an independent management system and the same operating policy.

When classifying routing protocols, based on AS, the routing protocol within the AS is called the inter gateway protocol (IGP). The leading IGP includes the routing information protocol (RIP), using a distance vector routing algorithm, and the open shortest path first (OSPF), using a link-state routing algorithm. The routing protocol between AS is called the exterior gateway protocol (EGP). The leading EGP is the border gateway protocol (BGP) that uses a path-vector routing algorithm.

D) Routing protocol type

The routing protocol includes RIP, which uses the distance-vector routing algorithm and delivers the routing table information every 30 seconds through broadcasting, IGRP, which improves some of the RIP problems by reflecting the network state (bandwidth, delay time, load, etc.), OSPF, which uses the link-state routing algorithm, and BGP, which connects between AS. <Table 50> shows the details of each protocol.

〈Table 50〉 Routing Protocol Types

Protocol	Description
RIP (RFC 1058)	• RIP uses the distance-vector routing algorithm and delivers the routing table information every 30 seconds through broadcasting. • The number of hops is limited (up to 16hop), VLSM is not supported, and load balance is not available. • It does not reflect the network state (bandwidth, delay time, load, etc.).
IGRP	• IGRP enhances some of the problems of RIP. • It reflects the network state (bandwidth, delay time, load, etc.). • The number of hops is increased (up to 255 hops), VLSM is not supported, and load balance is available.
OSPF (RFC 2328, RFC 1247)	• OSPF uses the link-state routing algorithm. • Information on the occurrence of changes, such as user-specified routes, the most economical routes, and multiple routes, spreads faster than RIP • Topology information remains the same across all routers, VLSM, and load balancing.
BGP (RFC 4271)	• EGP method that connects AS and the interconnection of large networks • Uses the TCP technique, based on the path-vector routing

06 IPv4 Overview

A) What is an IPv4 (Internet Protocol Version 4) address?

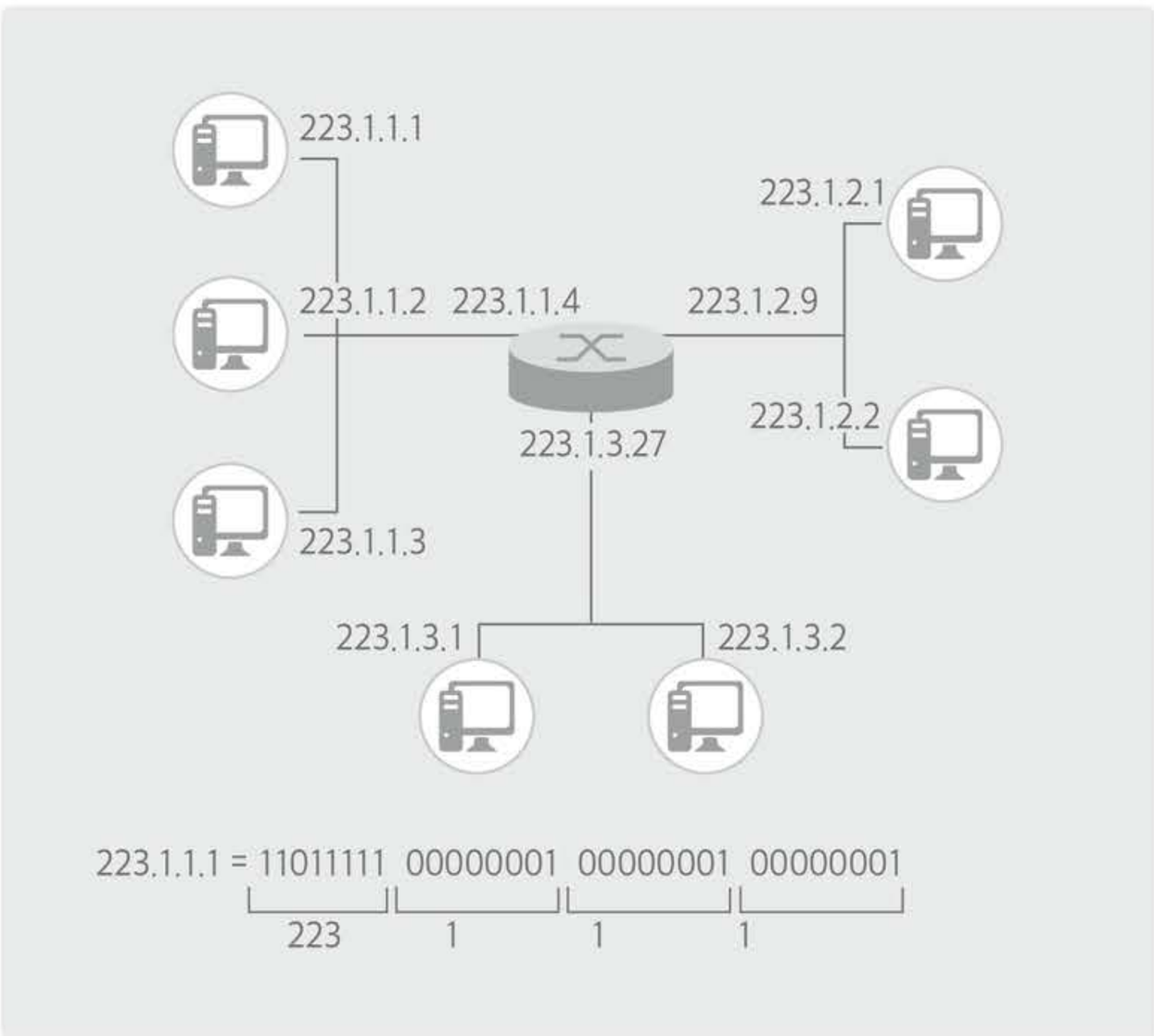
The IPv4 address is the 32-bit address used on the IP layer, as shown in [Figure 123]. It is an identifier that makes the router or host connection to the Internet unique and universal.

[Figure 123] IPv4 Address Structure

The elements of the IPv4 address include an IP address, which is a unique identification address given to all communication networks that are connected to the Internet and computers connected to the network, a network ID, which identifies TCP/IP hosts in the same physical network, a host ID, which identifies the TCP/IP host on the network, and the 32-bit subnet mask, which specifies the network name for the iP address from the hostname. CIDR (Classless Inter-Domain Routing, RFC 1519) is a method of expressing the IPv4 address and one of IP allocating methods that does not divide the network part from the host part in 8-bit units. Moreover, subnetting and supernetting can flexibly express the network and host Ips, according to the number of IPs required by the network, without dividing the IPv4 address and the network ID into fixed 8-bit units. Here, subnetting refers to dividing a given IP address into small subnets, according to the network environment, and supernetting refers to combining multiple network IDs into one network ID.

Here, subnetting refers to dividing a given IP address into small subnets to suit the network environment, and supernetting refers to combining multiple network IDs into one network ID.

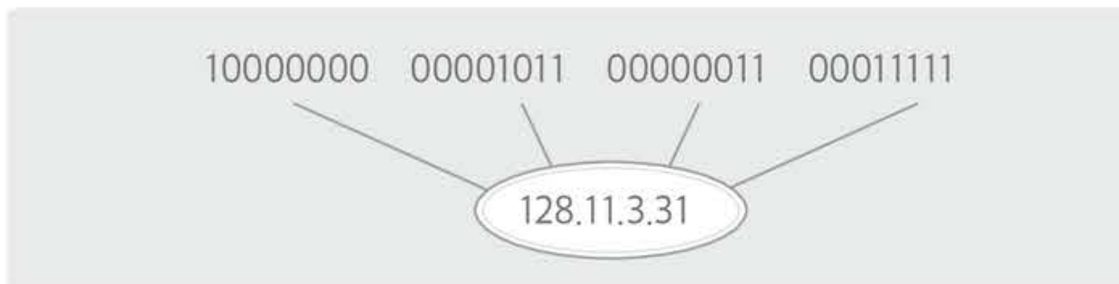
B) Network using IPv4 addresses (Example)



[Figure 124] Network Using IPv4

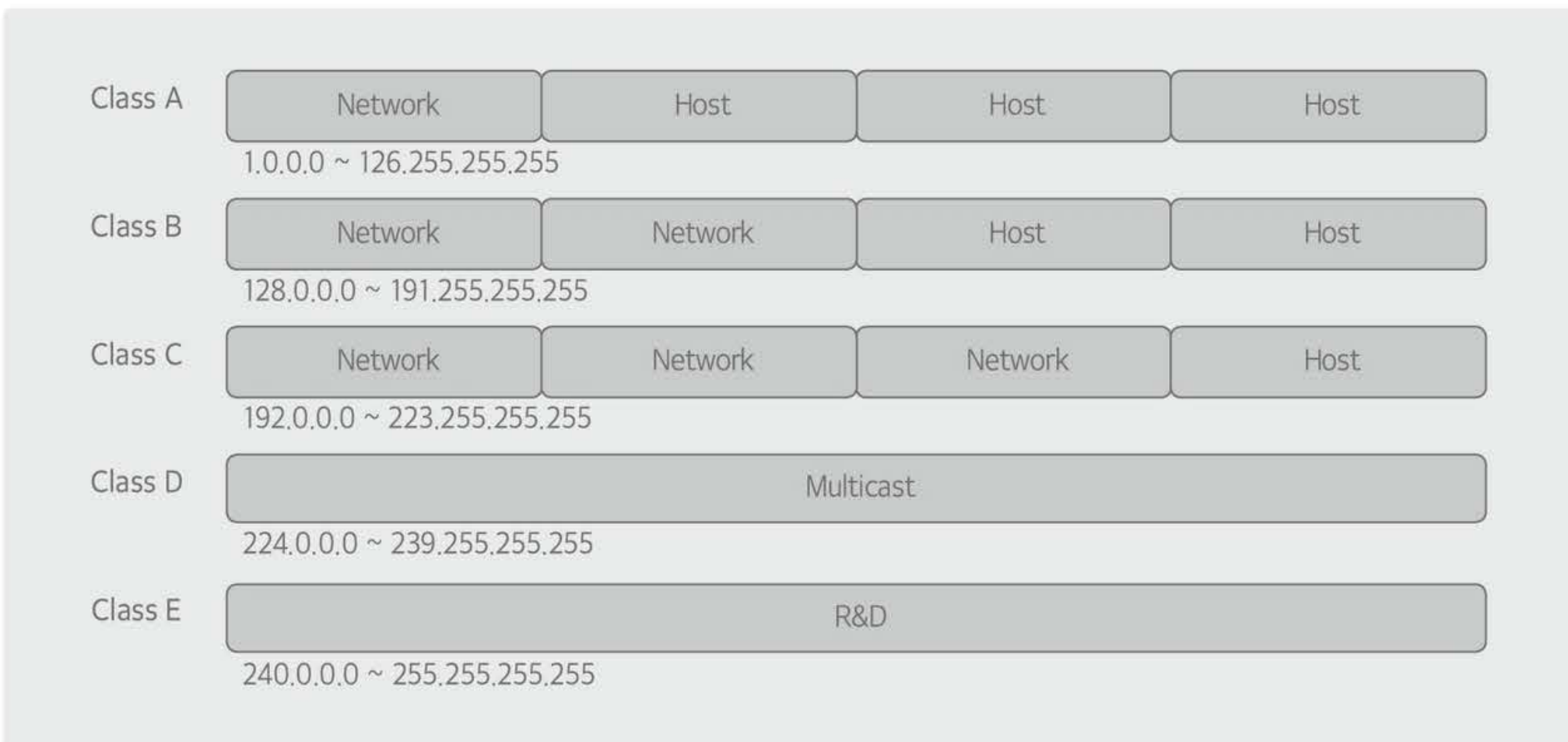
07 IPv4 Address Designation System and Subnetting

A) IPv4 expression



- Binary expression: 01110101 10010101 00011101 00000010
- Dotted decimal notation

B) IPv4 address designation system



[Figure 125] IPv4 Address Designation System

The IPv4 addressing system consists of a network address, assigned by an Internet address resource management organization, and a host address, assigned by a network administrator to identify individual hosts on the network. As shown in <Table 51>, it is divided into Classes A, B, C, D, and E, depending on the network size or the number of hosts. Classes A, B, and C are for network configuration assigned to general users, Class D is for multicast, and Class E is an address reserved for future use.

〈Table 51〉 IPv4 Address Designation System [7]

Class	Address	Description
A	1.0.0.0 ~ 126.0.0.0	Network ID: First octet Host ID: Last three octets Default subnet mask: 255.0.0.0
B	128.0.0.0 ~ 191.255.0.0	Network ID: First two octets Host ID: Last two octets Default subnet mask: 255.255.0.0
C	192.0.0.0 ~ 223.255.255.0	Network ID: First three octets Host ID: Last one octet Default subnet mask: 255.255.255.0
D	224.0.0.0 ~ 239.0.0.0	Multicast address
E	240.0.0.0 ~ 255.0.0.0	Experimentally used

In the IPv4 address designation method, subnetting is the address partition method used to separate a network ID into multiple network Ids, and supernetting is a method of integrating multiple networks into a large network.
The 169.254.0.0 network band is reserved for automatic private IP addressing. For versions after Windows 2000, DHCP automatically generates it to set the IPv4 address and cannot be routed on the Internet. The addressing system is called the automatic private IP addressing (APIPA).

C) Special IPv4 address

In IPv4, there is an addressing system in a different form from the conventional A, B, and C class addressing system. These addresses are called the special IPv4 addresses. <Table 52> below shows the special IPv4 addresses.

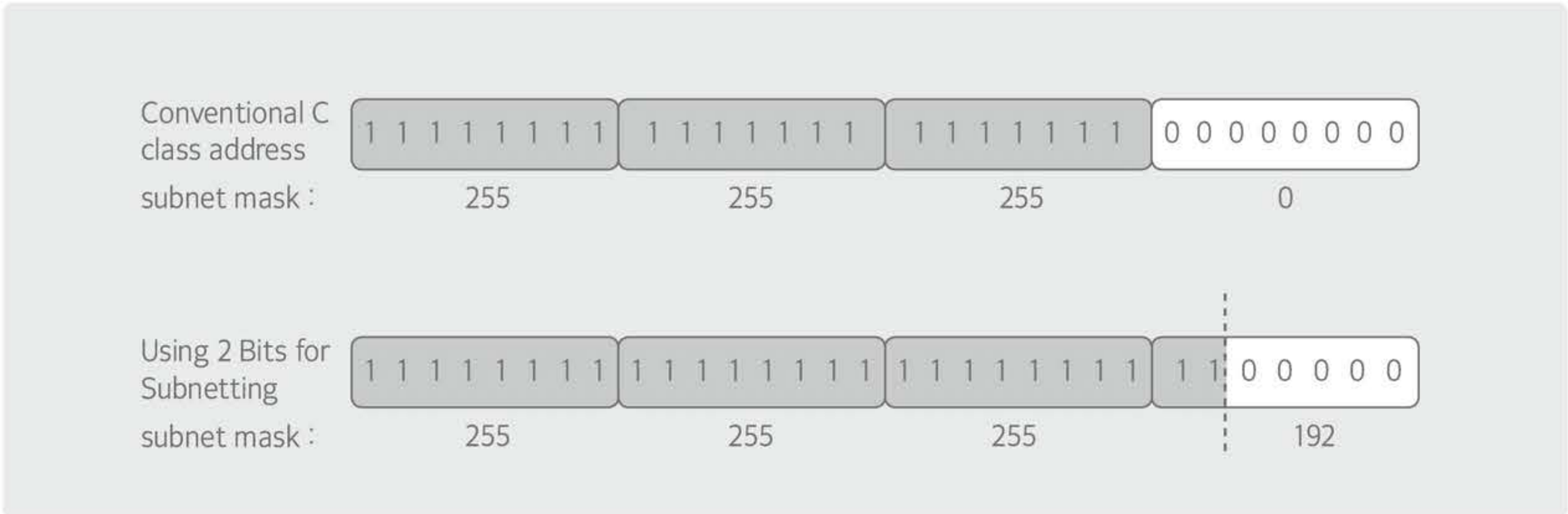
〈Table 52〉 Special IPv4 Address [7]

Network ID	Host ID	Address name	Usage
specific	All 0	Network address	• Network address
specific	All 1	Net-directed broadcast to netid	• Used to broadcast all terminals of the specific network
All 0	specific	Specific host on this network	• Indicates a specific terminal inside the connected network • Not able to pass through routers
127.X.X.X		Local loopback address	• Used as the destination address for loopback inside the system
255.255.255.255		Limited broadcast	• Used for broadcasting to all terminals inside the router network • Not able to pass through routers
0.0.0.0		This host on this network	• Used to specify itself when the terminal does not know its IP address • Not able to pass through routers
10. ~	Any	Private address for Class A (10.0.0.0 ~ 10.255.255.255)	• Private network address usable without an official approval
172.16~172.31	Any	Private address for Class B (172.16.0.0 ~ 172.31.255.255)	
192.168.0 ~ 192.168.255	Any	Private address for Class C (192.168.0.0 ~ 192.168.255.255)	

D) Subnetting

① Concept of subnetting

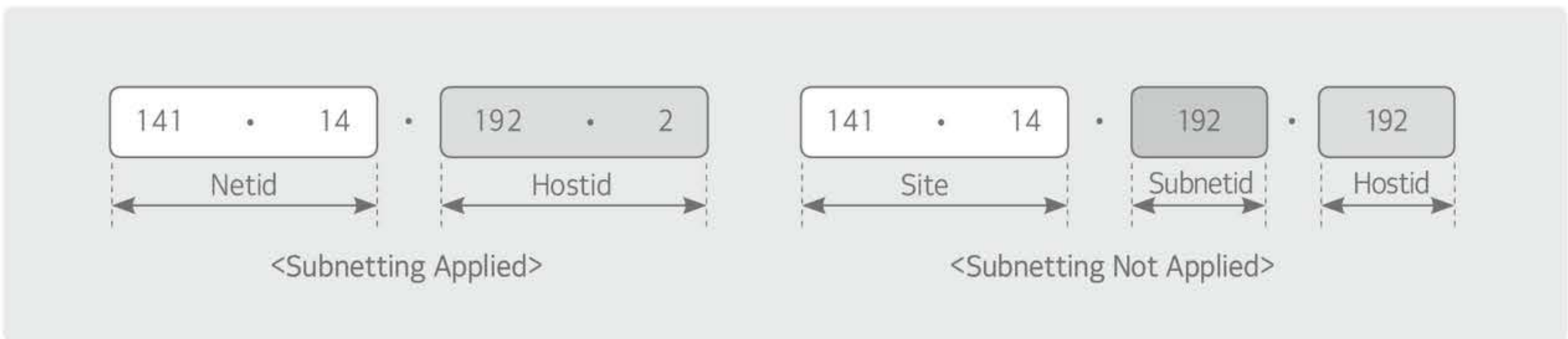
Subnetting is a method of dividing a network address into multiple smaller networks. A subnet mask refers to a mask to separate the network address of the IP address [8].



[Figure 126] IP Address Applying Subnetting

② Use of subnet

Adding a sub-network generates an intermediate layer in the IP address in order to separate the address into the site, subnet, and host.



[Figure 127] Addressing with and without a Subnetting Application

③ Use of subnet (Results of subnetting using 2 or 3 bits in a class C address)

〈Table 53〉 Subnetting Results

Type	Number of bits used in the subnetting address		
	(Not divided)	Using 2 bits	Using 3 bits
Subnet mask	255.255.255.0	255.255.255.192	255.255.255.254
Number of subnets	1	4	8
Number of hosts in each subnet	256-2 = 254	64-2 = 62	32-2 = 30
Number of IP addresses that can be assigned	254	4x62 = 248	8x30=240

E) CIDR (Classless Inter-Domain Routing)

① Concept of CIDR

CIDR is a method of combining multiple class C addresses into a network. It can reduce the size of the routing table and use IP addresses in various subnet formats.

② CIDR's role

- Provides flexibility in IP address operation, by allowing users to freely specify the range of network identifiers without classifying the IP addresses for each A, B, and C class.
- The network address can be set freely, preventing wasted IP addresses and enabling effective network configuration.
- It is an address allocation technique that enables the effective management of IP addresses used for routing between domains.

③ CIDR expression

The subnet part of an IP address has an arbitrary length with a format of a.b.c.d/x. In the address format, x refers to the subnet of the address.

[Figure 128] CIDR Notation

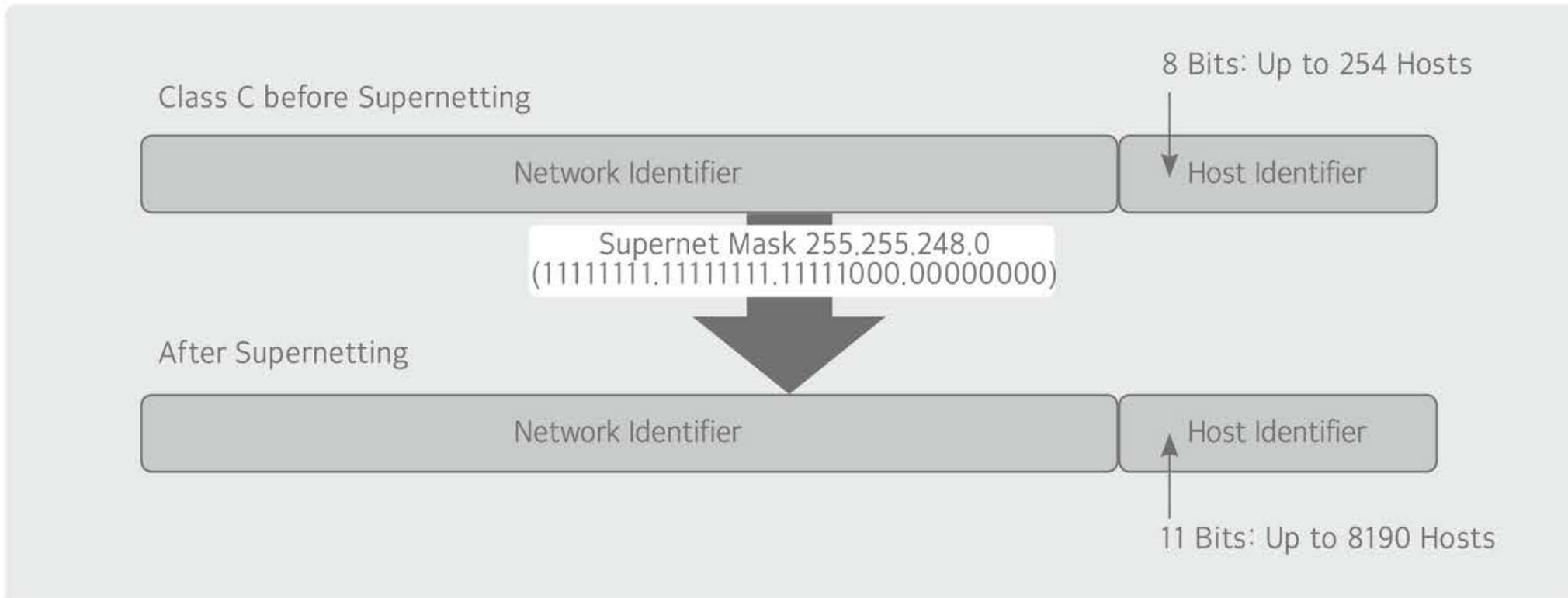
F) Supernetting

① Concept of supernetting

Supernetting is the opposite concept of subnetting, referring to the consolidation of several small networks into one large network. It is an address allocation technique that allocates consecutive class address blocks to an organization and enables routing from the outside, as if it were one address block [9].

② Supernetting method

- Grouping multiple class C addresses into a network
- Use of part of the network identifier as the host identifier
- The number of consolidated class C addresses must be a power of 2 and must consist of sequential addresses.



[Figure 129] Supernetting Method

G) IPv4 address allocation

① DHCP (Dynamic Host Configuration Protocol)

DHCP is a protocol that provides a method of dynamically allocating IP addresses, and other related configuration details, to DHCP-enabled clients on a network, using a DHCP server.

The DHCP server can assign a default gateway, a DNS server address, a local router, a lease period, and a domain name as DHCP options, in addition to the IP address and subnet mask.

The IPv4 DHCP lease process is established through the exchange of four UDP messages. The four UDP messages are DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, and DHCPACK. <Table 54> lists the details.

〈Table 54〉 DHCP Message [10]

Message	Description
DHCPDISCOVER	The IP address, sent by a client, has the address value of 255.255.255.255.
DHCPOFFER	The DHCP server delivers information, such as the IP address, subnet mask, default gateway, and lease period.
DHCPREQUEST	The client sends DHCPREQUEST to the server, and the message includes that the client has selected the lease information.
DHCPACK	The DHCP server sends the DHCPACK message to the client to allow the use of the IP address.

② NAT (Network Address Translation)

NAT is a function that interconnects terminals on a private network to communicate with public networks, such as the Internet. It is defined in RFC 3022 and RFC 2663.

The purpose of using NAT is to hide internal IPs to defend them from outside attacks while solving insufficient IP addresses. NAPT (Network Address Port Translation) is an extension of the basic NAT and converts the packet's source port number. The reason for using NAPT is to connect to multiple internal hosts with an IP address. NAPT can theoretically process communication with up to 65,535 internal hosts with an official IP address.

〈Table 55〉 NAT Type [11]

Type	Description
Basic NAT	- NAT that connects to the Internet with public IP addresses, as many as the number of terminals with private IP addresses
NAPT	- NAT in which multiple terminals with private IP addresses connect to the Internet, with a public IP address to save public IP addresses

08 IPv6 Overview

A) What is an IPv6 (Internet Protocol Version 6) address?

IPv6 is the next-generation Internet protocol with a 128-bit addressing system developed to solve the current IPv4 problems, such as address depletion, security, and mobility support.

B) Background of IPv6

The rapidly increasing hosts and terminals connected to networks have led to problems with the existing IPv4 addressing system, such as a shortage of IP address space and poor Internet security. The IPv6 protocol [16] was developed in the mid-90s to solve the problems. Compared to the existing IPv4, the most notable feature of IPv6 is the expansion of the address space, from 32 bits, to 128 bits, to enable the use of approximately 3.4 x 1032 addresses, thereby fundamentally solving the problem of address shortage. Moreover, it has simplified the head format, by removing header fields that IPv4 does not frequently use and supports effective QoS services. While IPv4 does not include security functions, it is necessary to install a security protocol called IPsec (Internet Protocol Security) [17][18] separately. IPv6 has added security and personal protection functions, by providing the end-to-end encryption function in order to solve these problems.

09 IPv6 Addressing System and Header Structure

A) IPv6 addressing system

IPv6 addresses are 128 bit long, four times that of IPv4, and expressed in hexadecimal. It consists of the format fix (PF)that describes the address format and the address field. A colon separates each 16-bit field, and each field is expressed in hexadecimal to make it easier to read the long addresses. The notation of the IPv6 address network prefix is similar to the CIDR notation of IPv4. The IPv6 address is followed by "/", and the length of the network prefix is written in decimal. <Table 56> below shows an example of the IPv6 address format prefix.

〈Table 56〉 Prefix of IPv6 Address Format

Format prefix	Description
0000 001	Reserved for NSAP (Network Service Access Point)
0000 010	Reserved for IPX (Internetwork Packet Exchange)
001	Integrated global unicast address
1111 1110 10	Link-local address
1111 1110 11	Site local address
1111 1111	Multicast address

The IPv6 addresses are classified into link-local, unicast, multicast, and anycast addresses, according to use. <Table 57> summarizes the IPv6 address notation.

〈Table 57〉 IPv6 Address Notation

Address format	Description
Link Local addres	• An address that can only be used within the internal network • The address can only be used within a network and is used to identify within that network only. FE80:0000:0000:0000:2CDA:D834:CBA2 Fill with 0s up to 64bit 10bit Prefic EUI-64 ID
Global unicast address	• While the link-local addresses can only be used internally, the global unicast addresses can be used externally. 2001:DBB:131F:0000:0000:070D:126A:140B / 64 ② Network Address ③ Interface ID ① Prefix
Multicast address	• Delivers packets to all interfaces that are registered in the multicast group. • FP is FF with 8 bits, followed by a 4-bit flag, a 4-bit scope, and a 112-bit group ID. • The upper 3 bits of the 4-bit flag are not used. If the last bit is 0, it means it is a well-known multicast address. If it is 1, it means it is a temporarily used multicast address.
Anycast address	• It is similar to a multicast address, in that it designates multiple interfaces, but packets are only delivered to the nearest interfaces, rather than to all the hosts in the group. • It is expressed as n-bit FP and 0 of (128-n) bits..

B) IPv6 header structure

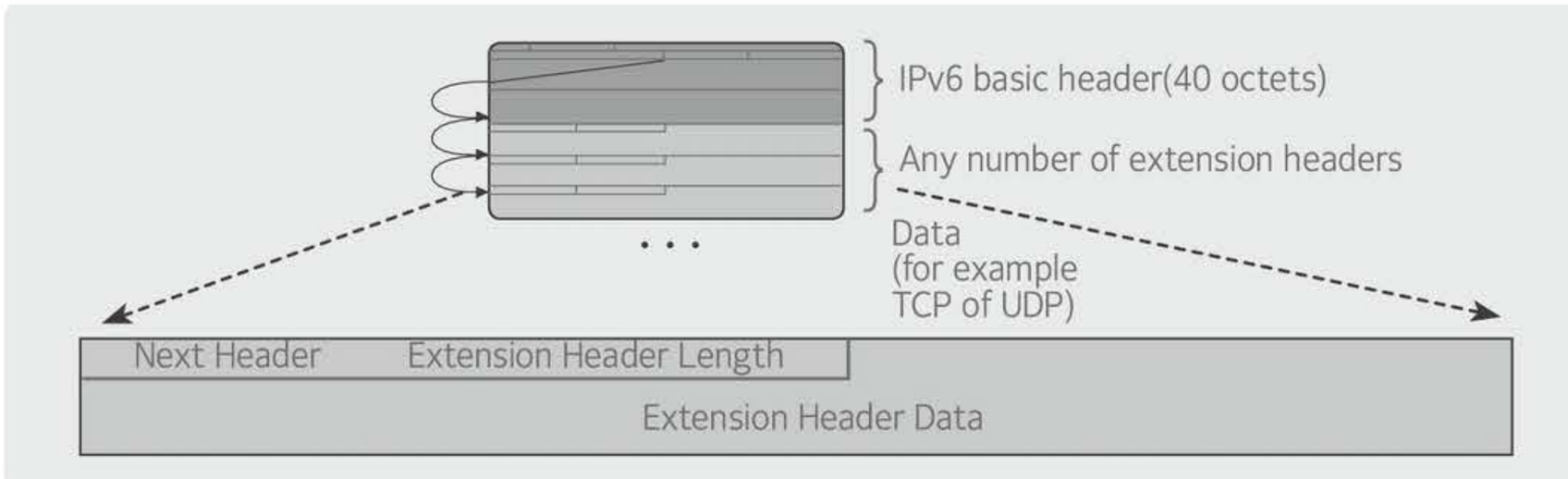
<Table 58> describes the IPv6's basic header structure and component. IPv6 can have extended headers, in addition to the basic header.

Version(4)	Traffic class(8)	Flow label(20)	
Payload length(16)		Next header(8)	Hop limit(8)
Source address(128 bits)			
Destination address(128 bits)			

〈Table 58〉 IPv6 Basic Header Structure and Components

Component	Size	Description
Version	4 bit	IP Version is marked. (version 4 or 6)
Traffic Class	8 bit	A function to request a transmission priority to a transmitting device
Flow Label	20 bit	Service classification for QoS is marked.
Payload Length	16 bit	Data length is marked.
Next Header	8 bit	Definition of the header type following the IP header
Hop Limit	8 bit	Forwarding limit for packet transfer is marked.
Source Address	128 bit	Address of the source device is marked.
Destination Address	128 bit	Address of the destination device is marked.

IPv6's extension header is set to 64 bits and configured in a daisy chain type. <Table 59> shows the structure and type of extension headers.



〈Table 59〉 IPv6 Extension Header Structure and Components

Header Type	Description
Hop-by-Hop	Information needed by all communication devices in the path to process packets
Destination	Information needed by the communication devices in the final destination to process packets
Routing	Routing path list by the sender
Fragmentation	Packet partition and recombination information after the increase of the transmission length
Authentication	Data integrity and sender authentication information
Encapsulation Security	Encryption of the packet's payload section

XIV. Transport Layer Protocol

▶▶▶ Recent trends and issues

As 4G wireless communication services have become popular, voice communication is also implemented through data communication, such as VoIP. The protocol layer directly related to these applications is the transport layer, and the network can be used effectively, only by selecting and using an appropriate transport layer protocol, according to the application characteristics.

▶▶▶ Learning objectives

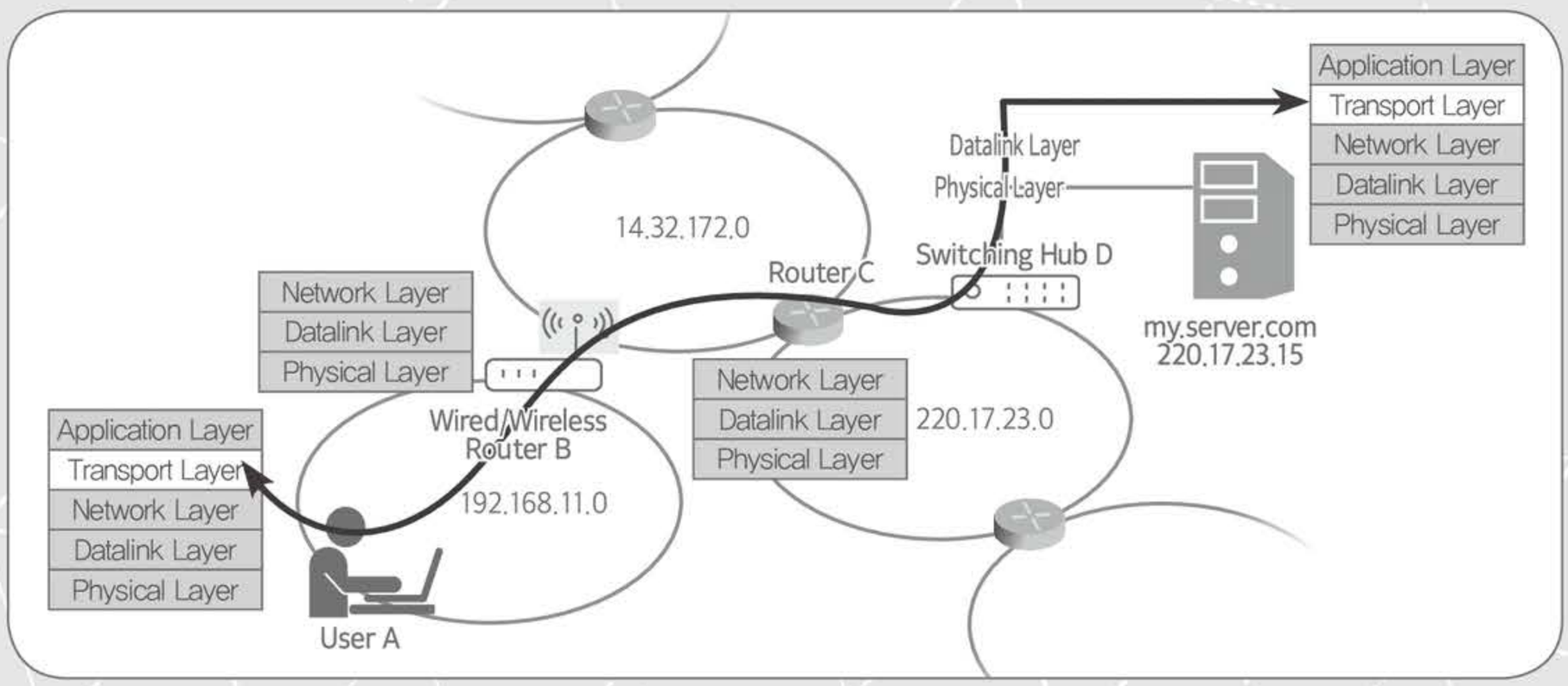
1. To be able to understand the purpose and functions of transport layer protocols and use them.
2. To be able to understand the concepts, services, and control techniques of TCP and use them.
3 To be able to understand the concepts, services, and control techniques of UDP and use them.
4. To be able to explain the concepts and basic operations of SCTP.

▶▶▶ Keywords

TCP, UDP, SCTP, Flow control, Error control, Congestion control, Slow start, SNMP, Multicasting, NTP, DHCP, Kerberos, Multi-homing.

✚ Practical preview **Operating path of transport layer protocols**

The transport layer protocols operate between the computer of user A, and the webserver my.server.com that requests data, i.e., end-to-end service. Data packets are delivered through many hops between a client and a server. The nodes, positioned between the terminals, such as the wired/wireless routers, are not involved in the transport layer protocol operation. TCP [1], UDP [2], and SCTP [3] are the leading transport layer protocols.



[Figure 130] Protocol Operation Path of Transport Layer

01 Concept of Transport Layer Protocol

The transport layer protocol transmits application data between end-point hosts. The server application running on the server side receives a request from the client side, processes it, and sends a response to the client. On the client side, a user selects an application to receive the service the user wants, makes a request to the server, and displays the response to the user. Applications that traditionally use networks, operate in a client-server structure, and the lowest protocol, connecting the client application and the server application, is the transport layer protocol.

Transport layer protocols that have been widely used are TCP (Transmission Control Protocol) [1] and UDP (User Datagram Protocol) [2]. SCTP (Stream Control Transmission Protocol) [3], which combines the advantages of TCP and UDP, is a new protocol introduced after 2000. This chapter focuses on TCP and UDP, which are currently widely used.

UDP is usually referred to as a connectionless service. However, it does not mean that the service is performed without a connection, but that the pre-configuration process for connection is omitted, and that data can be sent and received immediately. On the other hand, TCP is a connection-oriented service and requires a pre-configuration process to connect before sending and receiving data. The connection is terminated in order to end the service between the client and server after completing the data transmission and reception.

Reliability is a factor mentioned, along with the concept of connection-orientation, when referring to TCP and UDP. UDP is generally considered an unreliable protocol, and TCP is considered to provide reliable interprocess communication. TCP uses an acknowledgment technique that uses a sequence number and an acknowledgment number to check whether the receiving side has properly received the transmitted data.

02 TCP

A) Characteristics of TCP

TCP is positioned between the application layer and the network layer in a TCP/IP model and provides communication between the application layer processes. TCP is a stream-based protocol and has the transmission buffer and the reception buffer to transfer data. IP at the lower layer of the TCP protocol, transmits data in a packet format, rather than a stream format, by partitioning the data into several segments. Flow control, error control, and congestion control techniques are defined to reliably deliver data. TCP is numbered in units of bytes. When a byte is numbered, a sequence number is assigned to the segment. The maximum value must be set for a header to include the sequence number of each packet. If m bits are set for the sequence number in the packet header, the sequence number ranges from 0 to 2m-1. The sequence number is modular 2m.

① Flow control

Flow control controls the packet flow volume to prevent packet loss, due to packet inputs exceeding the reception limit when transmitting data packets. The flow control uses transmission and reception buffers that can store packets at the sending/receiving sides, and the sizes of the sending/receiving buffers are specified in the files /proc/sys/net/core/wmem_max and /proc/sys/net/core/rmem_max, respectively for Linux systems.

② Error control

Error control detects code errors, that occur during transmission, and reproduces accurate information. It detects and discards damaged packets, tracks and resends lost or removed packets, and checks and discards duplicate received packets.

③ Congestion control

Congestion occurs when the network load, representing the number of packets transmitted to the network per unit time, exceeds the network capacity, representing the number of packets that can be processed by the network per unit time. In TCP, the congestion control controls issues, such as end-to-end congestion, and ensures that the load is smaller than the network bandwidth.

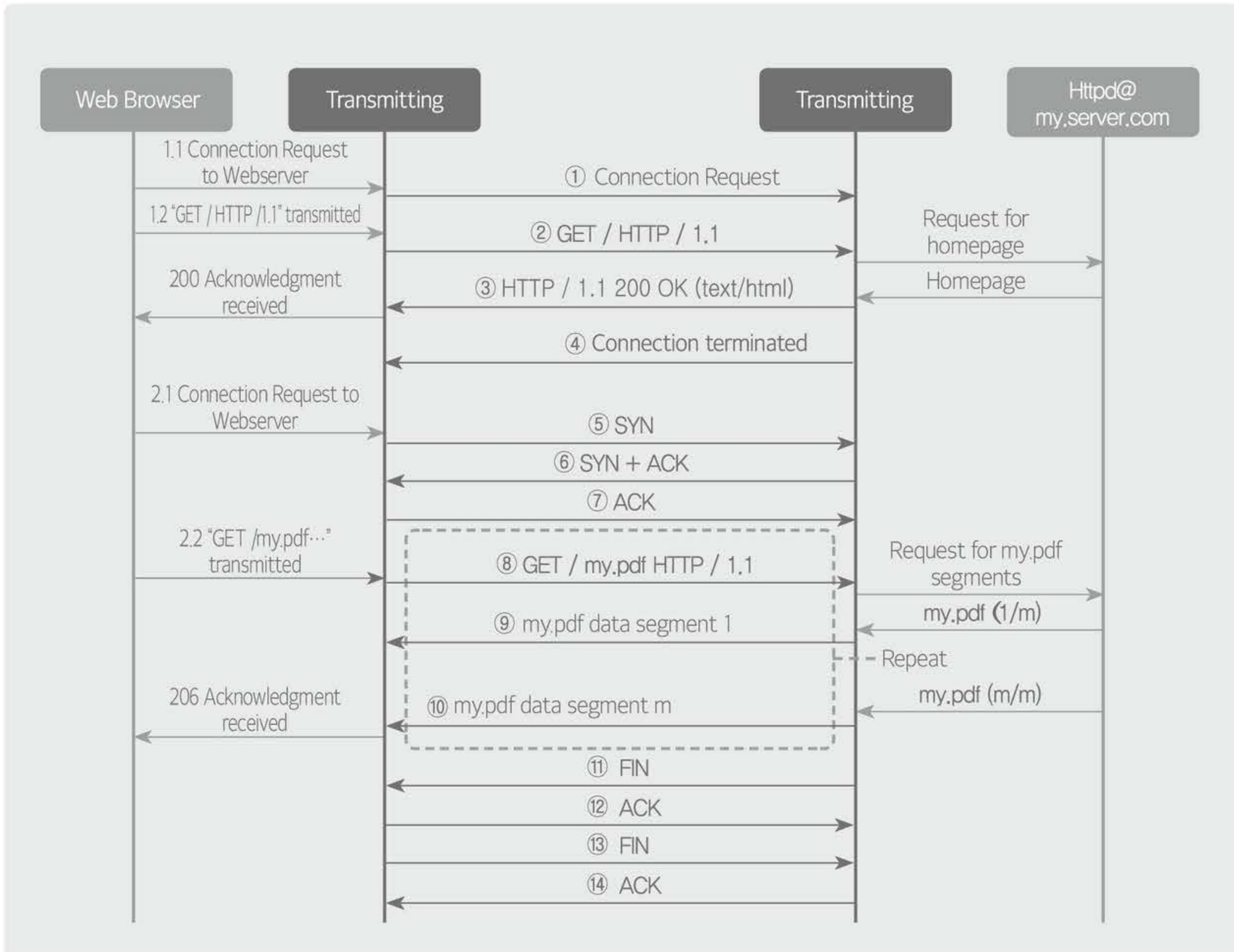
The following table shows well-known ports for widely used TCP services. These services include FTP, for file transfer, SSH, for secure remote login, SMTP, POP3, and IMAP4, to send and receive e-mail, and HTTP for web services.

〈Table 60〉 Well-known TCP Ports

Service	TCP Port	Service	TCP Port
FTP [4]	21 (for control) and 20 (for data)	FTP [8]	53 (also using UDP port 53)
SSH [5]	22	HTTP [9]	TCP 80
Telnet [6]	23	POP3 [10]	110
SMTP [7]	25	IMAP4 [11]	143

B) Virtual TCP operation scenario

The figure below shows a case of data transfer on the TCP protocol layer when user A enters the my.server.com homepage and downloads the my.pdf file.The web browser exchanges requests and responses: the transport layer protocol, TCP deliver an application layer program, and the webserver Httpd. Here, the discussion focuses on the operation of the TCP protocol.



[Figure 131] Virtual TCP Operation Scenario

The connection request in step ① is actually a simple drawing of steps ⑤ through ⑦, and the connection termination in step ④ is also a simple drawing of steps ⑪ through ⑭. Steps ② through ③ and ⑧ through ⑩ are application layer data that is immediately appended after the TCP header. According to the HTTP protocol, this data is generated by the web browser on user A's computer and the my.server.com web server program. The following section describes the TCP protocol and corresponding steps in detail.

TCP flow control is important when a large volume of data, such as PDF files, is transmitted simultaneously, as shown in steps ⑧ through ⑩, because IP packets can be transmitted through multiple network paths. Error control is crucial to cope with transmission errors. Moreover, it is also necessary to understand congestion control, since the transmission of more packets that can be processed per unit time in the network, greatly affects TCP performance.

C) TCP protocol

The header format used in the TCP protocol is shown in the following figure [1].
When adding an option, it is necessary to add a padding byte to fit the 4-byte unit. When there is no option, the TCP header is 20 bytes, which is the same size as the IP header. Therefore, the TCP/IP header is 40 bytes and must be sent, even when sending one byte of user data. For effective network transmission in an application layer program, data larger than an appropriate size must be sent to the transport layer.

〈Table 61〉 TCP Protocol Header Components

Type	Description
Source port address	• Defines the port number of the host application sending the segment (16 bits) • Performs the same purpose as the source port number in the UDP header
Destination port address	• Defines the port number of the host application receiving the segment (16 bits) • Performs the same purpose as the destination port number in the UDP header
Sequence no./ Serial no.	• Defines the number assigned to the first byte of data contained in the segment (32 bits)
Acknowledge number (ACK no.)	• The byte number that the segment sender expects to receive (32 bits)
Header length	• TCP header expressed as the number of 4 bytes (4 bits) • (Header length: 20 - 60 bytes)
Control	• 6 different control fields or flag bit definition
Window size	• The window size in bytes that the other party must maintain • 16-bit long. The maximum window size is 65,536 bytes. • The data size that can be sent to the segment, used by the windowing mechanism • Buffer size
Checksum	• Error detection for all checksums and segments (16 bits)
Urgent pointer	• Valid only when the urgent flag value is set (16 bits) • Used when the segment contains urgent data

The sequence number SEQ (sequence number) increases by increments of 1 for each 8-bit octet and performs the unsigned module 2^{32} calculation. Therefore, the sequence value changes to make $0 \rightarrow 2^{32} - 1 \rightarrow 0 \rightarrow \cdots$

The six flags for control have the following meanings. The SYN, ACK, and FIN control flags are widely used in the TCP protocol.
• URG: The urgent pointer field is meaningful.

- ACK: The acknowledged response field is meaningful.
- PSH: Push function
- RST: Connection reset
- SYN: Sequence number synchronization
- FIN: No more data from the sender

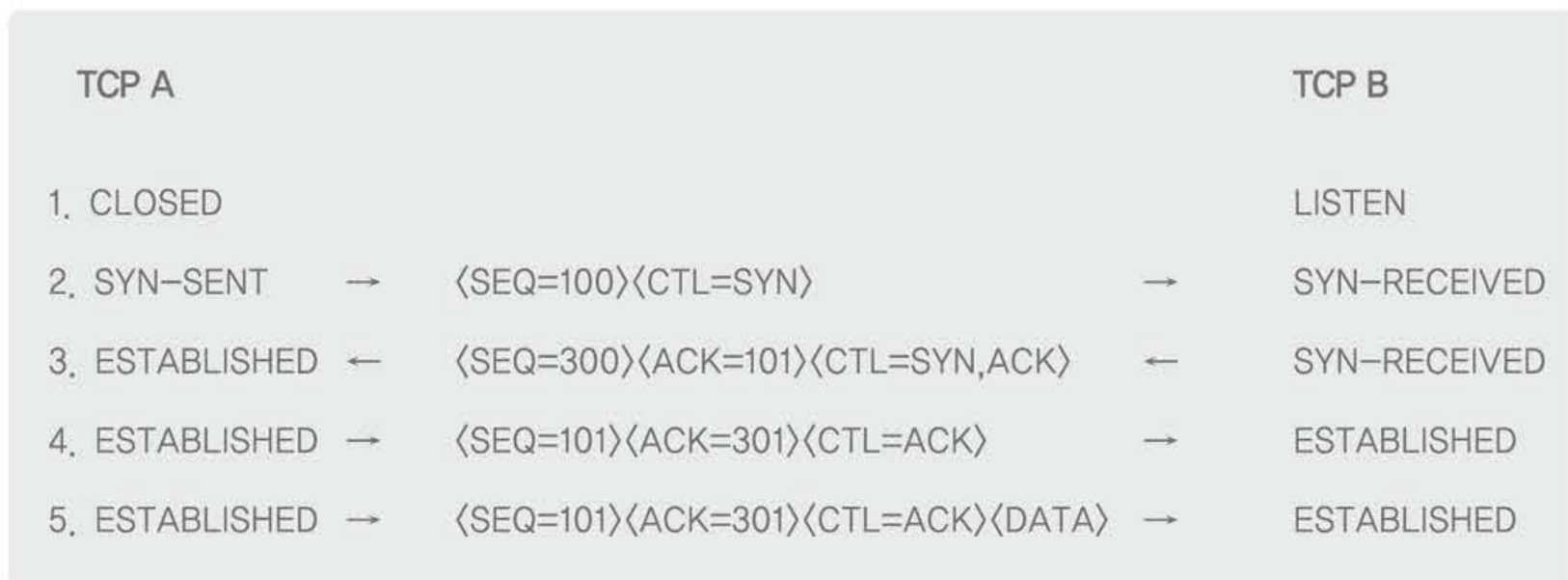
① TCP connection setting

The basic way to establish a TCP connection is with a three-way handshake procedure. This procedure begins by sending an SYN segment with a sequence number (SEQ) 100, from one TCP endpoint, to another TCP end, as shown in the second step of [Figure 132]. The SYN segment means that the SYN control flag of the TCP header is set to 1. The ACK segment means that the ACK control flag is set to 1. The SYN and ACK segments mean that both the SYN control flag and the ACK control flag are set to 1.

Upon receiving the SYN segment, TCP B sets the SYN control flag (SEQ=300), requesting connection establishment with TCP A from TCP B, along with acknowledgment number (ACK) 101[10], and sends the SYN and ACK segments, as shown in step 3 in [Figure 132]. Upon receiving it, TCP A confirms the ACK control flag and ACK=101 to recognize that the connection request has been properly delivered.

The SYN control flag set together indicates that TCP B sent the connection request. The three-way handshake is complete when the ACK segment (ACK=301) is sent to TCP B, and the bidirectional connection, between TCP A and TCP B, is established.

As described above, the three-way handshake for TCP connection confirms that the SYN segment, the SYN and ACK segment, and the ACK segment are transferred between TCP terminals. At the same time, the sequence number (SEQ) that is transferred between TCP terminals plays an important role in flow and error controls to check whether the data streams are transferred properly in byte units.



[Figure 132] Three-way Handshake for Connection Establishment [1]

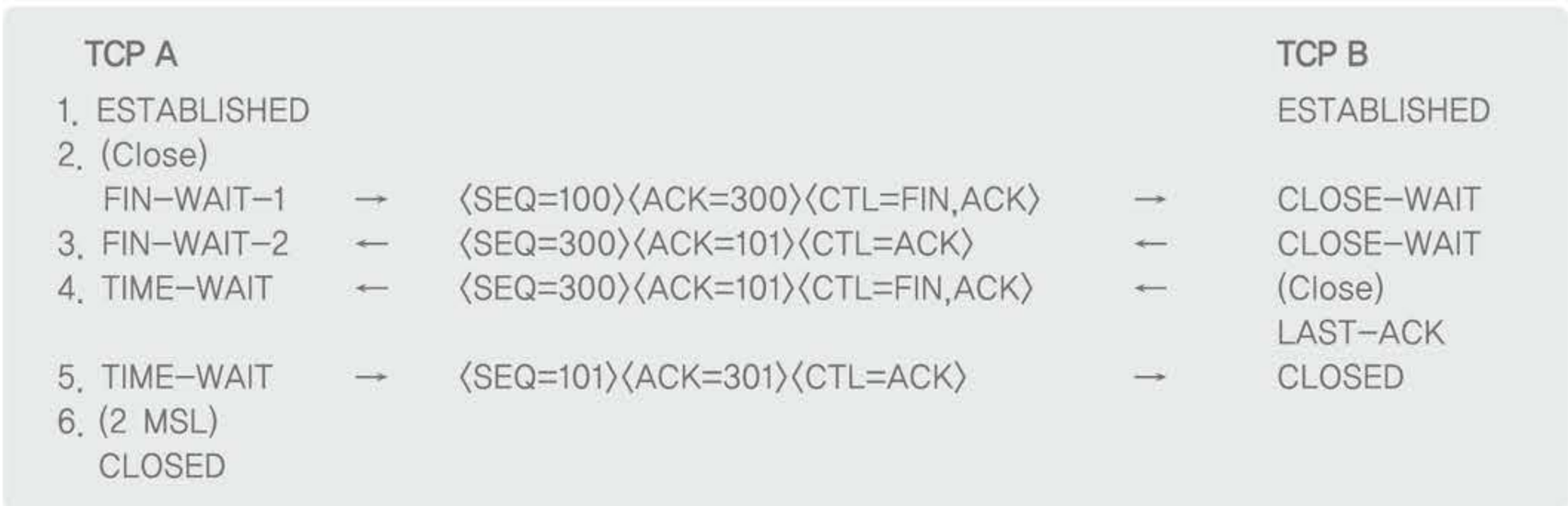
TCP manages connection by allocating a system resource, TCB (Transmission Control Block), upon receiving the SYN segment for connection. If a large number of SYN segments are sent in a short time for malicious purposes, the TCP end-host that has received it may not be able to allocate any more TCBs, resulting in the service being unavailable. This attack is called SYN flooding, and it can be referred to as a type of denial-of-service attack.

10 According to the TCP flow control method, the next desired byte sequence number 101 is sent as the acknowledgment number.

② TCP connection termination

The TCP connection is terminated in the following sequence. The terminal TCP A that intends to terminate the connection, sends the FIN segment (SEQ=100, ACK=300) to TCP B, as in step 2. Here, the ACK control flag is set and sent as an acknowledgment response to the previous transmission. Upon receiving it, TCP B sends an ACK segment (SEQ=300, ACK=101) in response, as in step 3. In response to SEQ=100 of the FIN segment, the ACK segment increased by increments of 1 to 101 is transmitted.

TCP B then sends the FIN segment (SEQ=300, ACK=101) to TCP A to terminate the connection, as in step 4. As in step 2, ACK is sent together for acknowledgment of previous data transmission. Then, TCP A completely terminates the TCP connection by sending an ACK segment (SEQ=101, ACK=301), as in step 5.

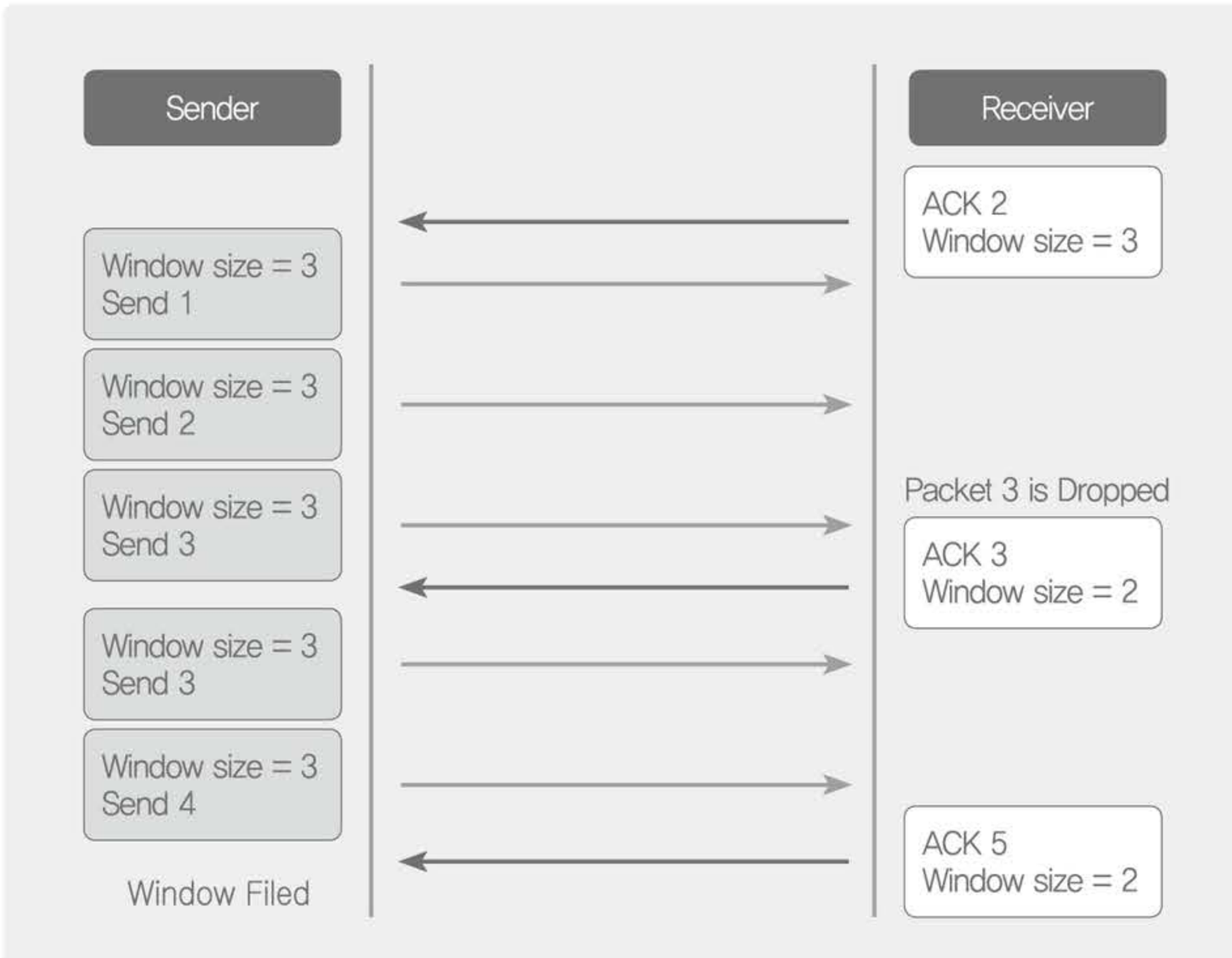


[Figure 133] Typical Termination Sequence [1]

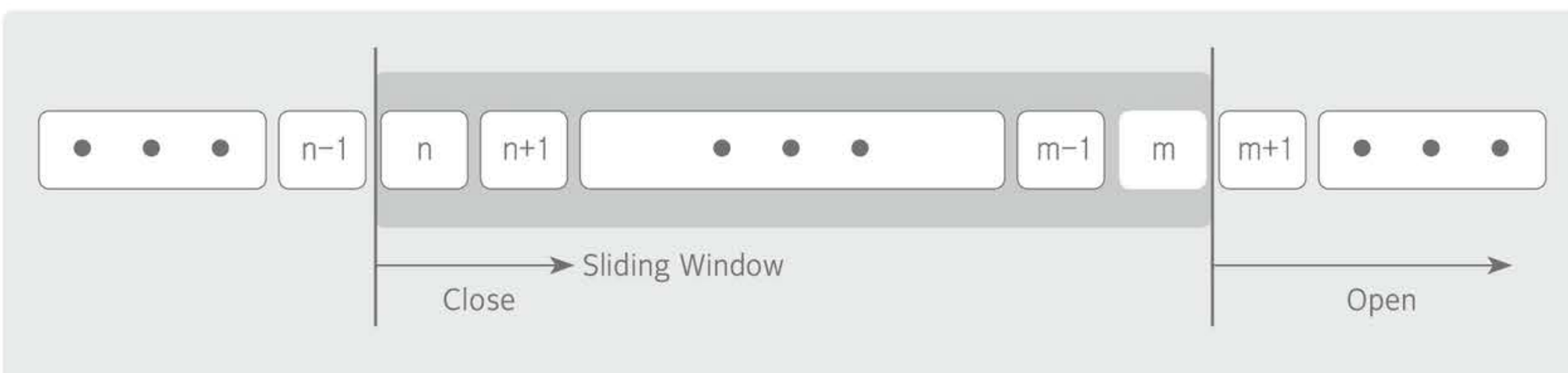
In the virtual TCP operation scenario shown in [Figure 133], steps ④ and ⑪ through ⑭ correspond to TCP connection termination.

③ Flow control

The flow control balances the rate of data creation and use. TCP uses the sliding window protocol. The receiving TCP notifies it by designating the number of octets (1 byte) prepared in advance. The sending TCP may refer to it and adjust the number of sliding windows. Since this field is 16 bits long, the maximum size of the window is 65,535 bytes.



[Figure 134] Example of Packet Sending and Receiving by Sliding Window Protocol



[Figure 135] Concept of Sliding Window Protocol

〈Table 62〉 Sliding Window Protocol Open/Close Operation

Operation	Description
Open operation	• Means that an ACK has arrived from a sender, and the window's right boundary moves to the right. • More data can be transmitted as it increases.
Close operation	• Means that the data (byte) is transmitted, and the window's left boundary moves to the right. • The sender side does not have to intervene with this data any longer.

The sliding window protocol uses the receiver window (RWND) and the congestion window (CWND). The window size is the lesser of the RWND and CWND.

〈Table 63〉 RWND and CWND

Type	Description
Receiver window (RWND)	• The size of the packet that the receiver can receive without data loss. • It notifies the status to the other end of the segment containing an ACK.
Congestion window (CWND)	• The size of the packet that can be sent at once, according to the network congestion • The CWND gets smaller, since data loss during the transmission increases if the network congestion is high. It gets larger if the congestion is low.

④ Error control

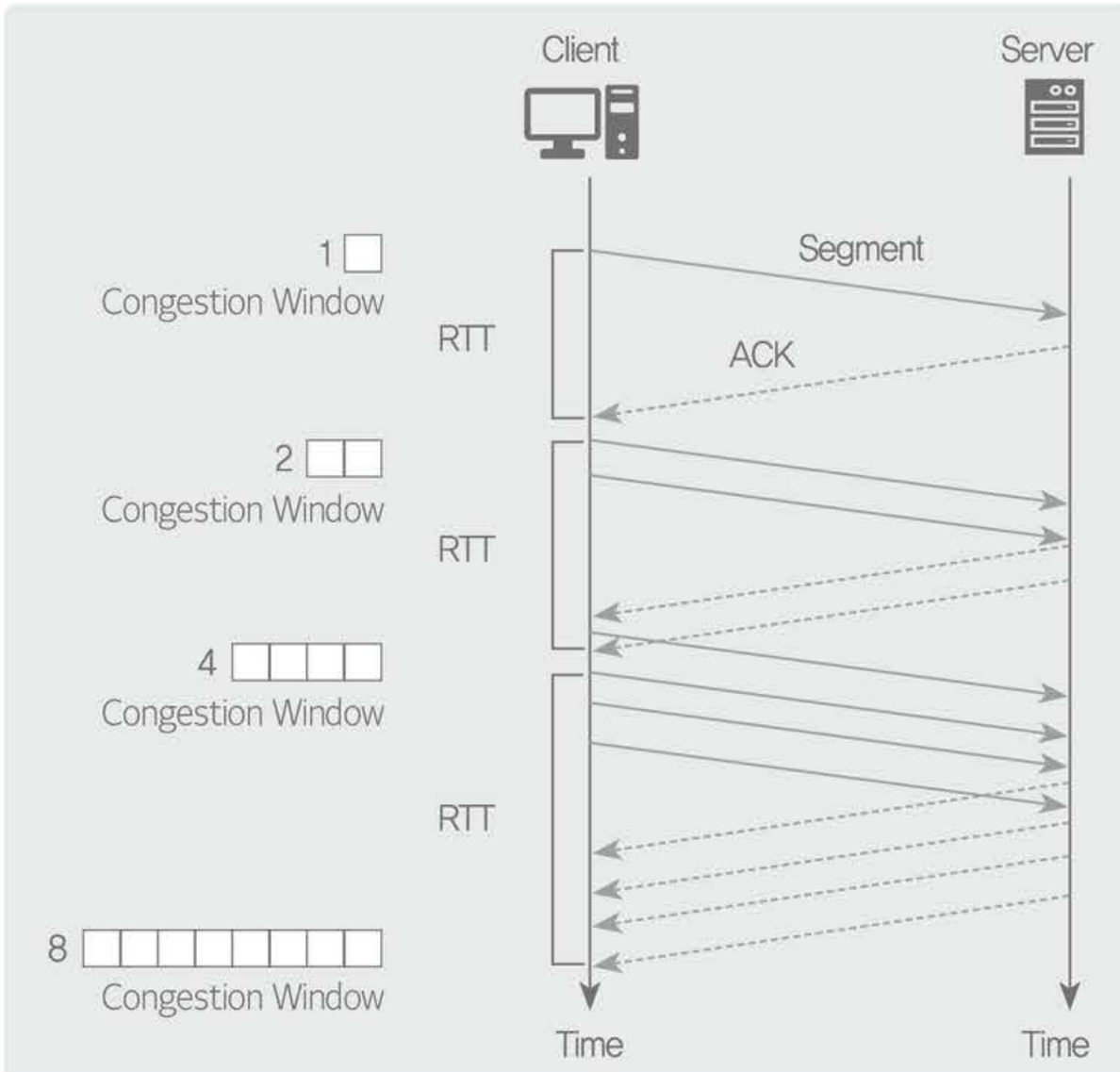
Error control is activated when the transmitted segment is lost, damaged, out of order, or duplicated.

〈Table 64〉 TCP Error Control Tools

Tool	Description
Checksum	• Each segment contains a checksum field to indicate if the segment is damaged.
Acknowledgment	• The acknowledgment is used to confirm receiving the data segment.
Resend	• The key to the error control mechanism. The segment is stored in a buffer until the acknowledgment is received.

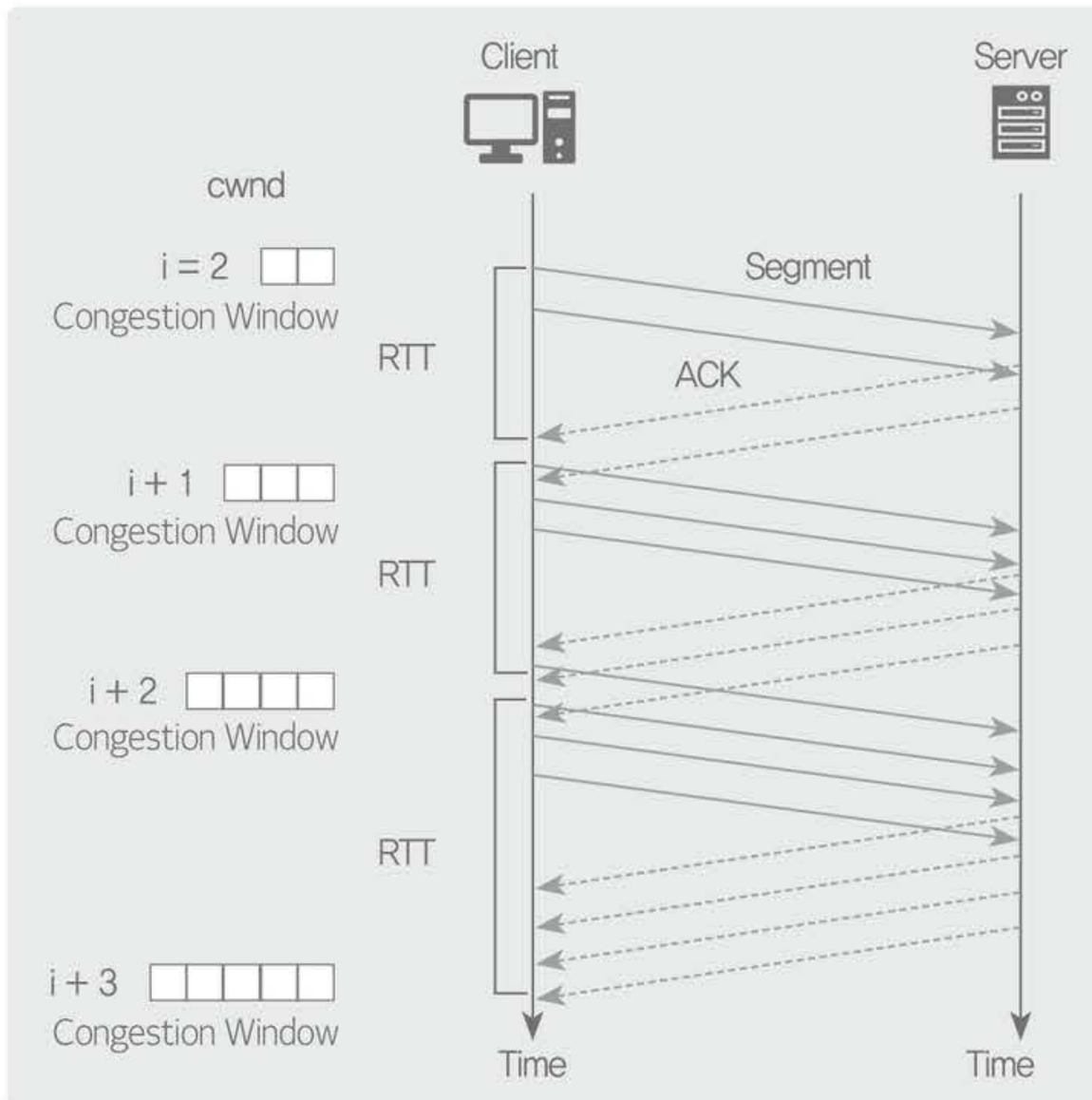
⑤ Congestion control

Congestion control is a mechanism and technique of maintaining the user traffic volume input to the network to ensure that it does not exceed the network capacity. TCP performs the congestion control, based on the slow start algorithm and the congestion avoidance algorithm.
The slow start algorithm exponentially increases the congestion window size for congestion control, until it reaches the limit.



[Figure 1362] Concept of Slow Start

The congestion avoidance algorithm additively increases the congestion window size, until the congestion stat is detected.



[Figure 137] Concept of Congestion Avoidance

⑥ TCP timer

TCP mostly uses four timers for its operation.

〈Table 65〉 Types of TCP Timers

Timer Type	Description
Retransmission	• A retransmission timer is set to a retransmission time-out (RTO) value, which is the time to wait for acknowledgement by a segment for retransmission of a lost segment.
Persistence	• A persistence timer is used for each TCP connection to resolve the deadlock between the receiving and sending TCPs.
Keepalive	• A keepalive timer is used to prevent TCP established connections from being idle for long periods of time.
Time-wait	• Itime-wait timer is used during connection termination. TCP performs an active close, and the time-wait timer is activated when sending the last ACK.

03 UDP (User Datagram Protocol)

A) Characteristics of UDP

User datagram protocol (UDP) provides a connectionless service for sending and receiving data without a pre-configuration process for connection. Unlike TCP, it is a protocol that does not guarantee the reliability of the transport layer level.

Each user datagram of UDP is an independent datagram and is not assigned a number, unlike TCP. UDP is a very simple protocol, and there is no flow control, so there is no window mechanism, and packet overflow may occur. Error Control also has no other error control mechanism, except for checksum. When the receiver detects an error through a checksum, the user datagram is removed, and congestion control is not provided. Due to this nature of the UDP transport protocol, the application layer may be necessary in order to define and implement the flow control and error control. As described above, since the UDP has a continuous minimum transmission rate, despite some packet loss, due to a lack of reception confirmation and retransmission function, it is suitable for video play service and is considered to provide the best effort service. In particular, for playing video that is transmitted in real time, the transmitting side must separately include the image frame order in the data. The receiving side must store it in a separate buffer, according to the frame order and process it in order. In the case of a streaming method, instead of a download method, if there is a delay for more than a certain time, or if only the B and P frames are received without the I frame, the application layer needs to maintain the real-time property by destroying the data.

The UDP protocol is mainly used for the following purposes.

- A process requiring simple request-response communication, without flow and error controls
- A process with internal flow and error control techniques
- Multicasting transmission technology
- Management processes, like SNMP
- Route update protocols, such as the routing information protocol (RIP)

<Table 66> below shows well-known ports for widely used UDP services. It supports the NTP service that corrects the host time through the network, the BOOTP and DHCP services that are network IP management protocols, the TFTP service that is simpler than FTP, the RIP and OLSR services that are routing protocols, and the Kerberos service that is a computer network authentication protocol.

〈Table 66〉 Well-known UDP Ports

Service	TCP port	Service	TCP port
NTP [12,13]	123	Syslog	514
BOOTP Server, DHCP Server [14,15]	67	RIP [18]	520
BOOTP Client, DHCP Client [14,15]	68	RIPng	521
TFTP [16]	69	Timed (Time Server)	525
Kerberos [7]	88	OLSR [19]	698

B) UDP protocol

A UDP packet, called the user datagram, has a fixed-size 8-byte header made of four 2-byte (16-bit) fields, as shown below. Each field of the UDP header is used for the purposes shown in <Table 67>.

<Table 67> User Datagram Header Fields

Field	Length	Description
Source Port	2Bytes	The port number of the source packet in the range of 0-65,535
Destination Port	2Bytes	The port number of the destination packet
Length	2Bytes	The length of the entire packet, including the UDP header and data fields
Checksum	2Bytes	A value to detect a data error
Data octets	Variable	Storage of data to be transmitted

The source port refers to the port of the transmitting process, and the destination port refers to the port of a specific destination host. The length is the sum of the UDP header (minimum 8) and the data size. The checksum is the result of adding 16-bit 1's complement to the pseudo-IP header, UDP header, and data. The checksum field is filled with zeros before the calculation.

As described above, unlike TCP, the UDP protocol is simple to implement because it receives data from the application layer and sends it down to the IP layer. It also analyzes the UDP user datagram, that is received from the IP layer, and delivers it to an appropriate application program.

04 SCTP (Stream Control Transmission Protocol)

A) Characteristics of SCTP

The stream control transmission protocol (SCTP) is a new transport layer protocol designed by combining UDP and TCP advantages for multimedia communication [3]. The service characteristics of the SCTP are described in the following table. The SCTP association is a broader concept than the end-to-end TCP connection. In an SCTP association, one or more IP addresses can be connected at each endpoint. It is called multi-homing, and it enables network-level fault tolerance. SCTP is called association in order to highlight its multi-homing nature. While a stream is transmitted and received to an end-to-end connection in TCP, multiple streams can be transmitted and received per association in SCTP.

〈Table 68〉 SCTP Service Characteristics

Features	Description
Process-to-process communication	• Provides process-to-process communication
Multi-stream	• A multi-stream service, called association, is allowed in each connection.
Multi-homing	• The transmitting/receiving host defines multiple IP addresses on each terminal for an association. • Only one of the IP addresses is the primary address, and all others are alternative addresses.

Full duplex communication	• Full duplex communication to deliver data to both directions simultaneously
Connection-oriented	• SCTP is a connection-oriented protocol and is called association.
Reliability	• The reliable transmission protocol uses acknowledgment to confirm the arrival of data.

The number systems used by the SCTP, include the transmission sequence number (TSN), the stream identifier (SI), and the stream sequence number (SSN). The data units in TCP are bytes, whereas data chunks are data units in SCTP. A transmission sequence number (TSN) is used to number data chunks. The SCTP's TSN plays a similar role to the TCP's sequence number. Moreover, since multiple streams may exist per SCTP association, a 16-bit stream identifier, starting from 0, is used to distinguish them.

The SSN is used to distinguish data chunks belonging to the same stream.

B) SCTP

Since it is not feasible to discuss everything about SCTP, this section briefly describes the SCTP transmission service, as shown in the following figure. SCTP functions can be classified into the following eight types.

- Association start and end
- Sequenced delivery
- User data fragmentation
- Acknowledgment
- Congestion avoidance
- Chunk bundling
- Packet validation
- Path management

① Association start and end

An association begins with an SCTP user's request. When end-point A sends an INIT chunk to end-point Z, in order to prevent an SYN attack (synchronization attack), end-point Z sends the INIT ACK chunk containing cookie information. Upon receiving it, end-point A copies the received cookie and sends the COOKIE ECHO chunk back to end-point Z to inform that it has been properly received, and end-point Z sends the COOKIE-ACK chunk again, to complete the four-way handshake. Generating a transmission control block (TCB), which is the main system resource, only when the normal COOKIE ECHO chunk is received at end-point Z, solves the denial of service (DoS) problem caused by an SYN attack, due to generating a TCB as soon as the SYN segment is received during the TCB connection establishment.

② Sequenced delivery

In SCTP, a stream means a series of user messages. The SCTP user can set the number of streams that can be supported when setting the association, and these streams are identified by the stream identifier SI. Moreover, assigning the SSN to the user message in the stream, enables the delivery of data chunks in normal order. Other streams can be transmitted even if one stream is blocked to deliver the next sequence of user messages.

③ User data fragmentation

As in TCP, user messages are limited by a path maximum transmission unit (MTU) in SCTP, and a fragmentation

technique is provided to resolve it. Each fragment that has been divided and transmitted is reassembled in the SCTP layer to become one user message.

④ Acknowledgment and congestion avoidance

SCTP transmits a data chunk by assigning a TSN to it, regardless of the stream. The receiving side separately secures the transmission reliability from the sequenced stream transmission by acknowledging (ACK) all the received TSNs. Acknowledgment and congestion control functions are related to packet retransmission if an acknowledgment is not received within a given time. Packet retransmission is performed according to a congestion avoidance procedure, similar to TCP congestion control.

⑤ Chunk bundling

A chunk with multiple user data or SCTP control information may follow an SCTP packet's common header. An SCTP user can request to bundle multiple user messages in a single SCTP packet. The SCTP chunk bundling function is responsible for assembling multiple chunks into one SCTP packet and decomposing it.

⑥ Packet validation

The SCTP common header has a verification tag (VT) and a 32-bit checksum field. The VT is a value selected at each endpoint when setting up the association. During the set association, SCTP packets must be transmitted using the same VT, and if an SCTP packet with a wrong VT is received, it is discarded. Moreover, generating a CRC32c checksum and transmitting the SCTP packet prevents data corruption in the network.

⑦ Path management

The transmitting SCTP user can use the path management function to manipulate the transport address, SCTP port number, and IP address, used in the SCTP packet destination. When the association is established, the primary path of each SCTP endpoint is defined and used to transmit and receive SCTP packets. However, if there is a problem with the current path, and available transmission address among the addresses, defined by multi-homing, can be designated as an alternative path. The path management and the packet verification that uses the VT and checksum are performed simultaneously.

SCTP packets have a general header and set of blocks called chunks. There are two types of chunks: control chunks and data chunks. Control chunks are delivered before data chunks.

The general header defines the endpoint of each association of the packet, guarantees a packet belonging to a specific association, and preserves the integrity of the contents of the packet, including the header itself.

〈Table 69〉 SCTP Protocol Header Components

Type	Description
Source port no.	• Same as TCP and UDP
Destination port no.	• Same as TCP and UDP
Verification tag	• Used as the identifier for the association and repeated in all packets during the association.
Checksum	• The size of a checksum is 32 bits to enable CRC-32 checksums. • (16-bit checksums are used in UDP, TCP, and IP.

Control information and user data are carried in chunks, and three fields of Type, Flag, and Length are common in all

chunks. The information field indicates the type of the chunk, and the Type field is defined up to 256 bytes of the chunk, but only some are defined, and the rest are reserved for future use. The Flag field defines special flags that may require special chunks.

In some chunk types, the data chunk is identified as the control chunk. <Table 70> shows major chunk types.

〈Table 70〉 Chunk Types

ID value	Chunk type	ID value	Chunk type
0	Payload data	6	Abort
1	INIT	7	SHUTDOWN
2	INIT ACK	8	SHUTDOWN ACK
3	SACK (selective ACK)	9	ERROR
4	HEARTBEAT	10	COOKIE ECHO
5	HEARTBEAT ACK	11	COOKIE ACK

XV. Application Layer Technologies, Including Web Application

▶▶▶ Recent trends and issues

With the advent of mobile platforms, such as Android and iOS, since the mid-2000s, society has rapidly transformed into the mobile era, as applications useful for work and personal leisure have emerged. Some programs use standard application layer protocols, such as SMTP, POP3, IMAP, and FTP, and some others, like SNS, use their own application layer protocols, such as KakaoTalk, Line, and Facebook. Developers can only develop user-friendly application programs when they have a good understanding of the concept and operation method of the application layer protocol operating on the transport layer.

▶▶▶ Learning objectives

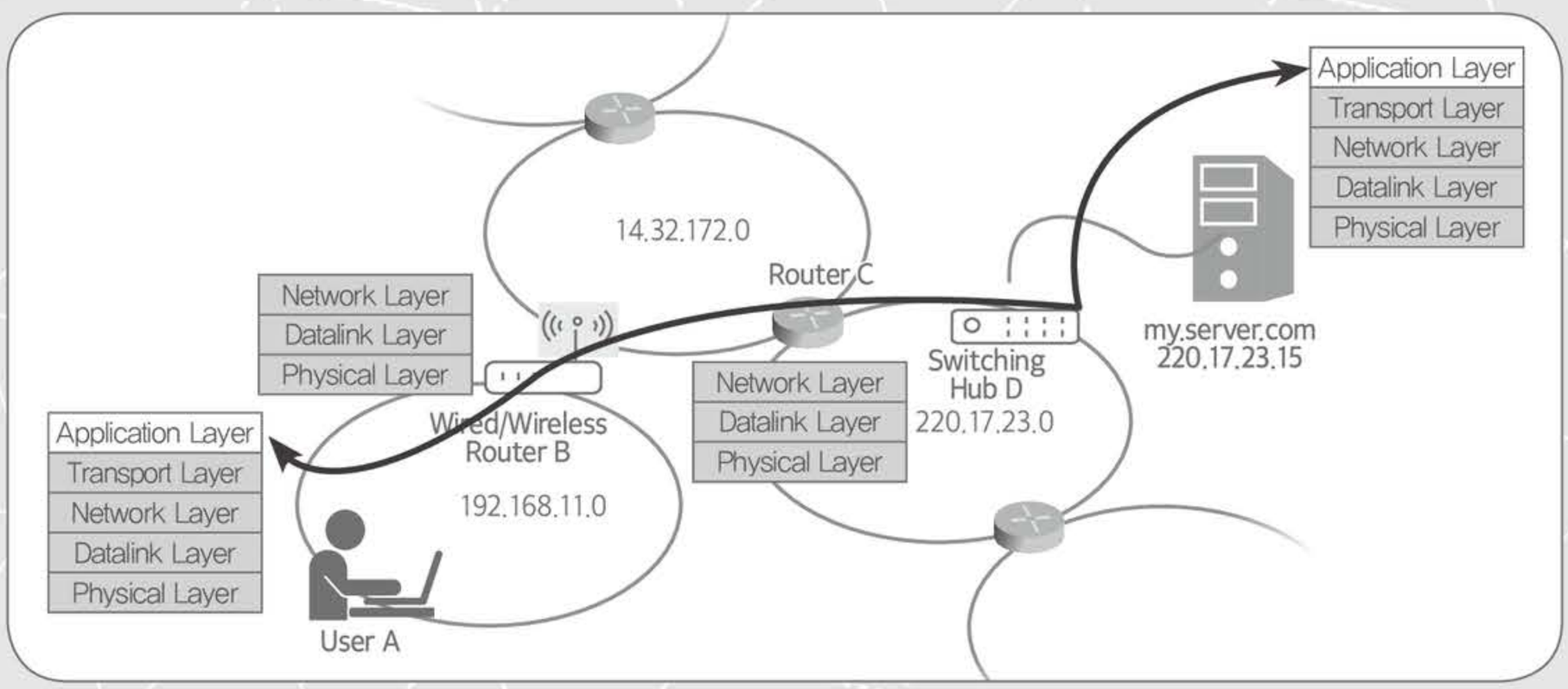
1. To be able to understand application layer protocols and design or utilize them.
2. To be able to understand HTTP protocols for data exchange between clients and webservers and utilize them.
3. To be able to understand FTP protocols for file exchange between clients and webservers and utilize them for program development. Moreover, they will be able to use open-source libraries in order to increase the productivity of application program development.
4. To be able to understand JSP-based server programming techniques and client programming techniques for web application program development and develop software..

▶▶▶ Keywords

Application layer protocol, HTTP, GET method, POST method, FTP, Control connection, Data connection, General transport mode, Passive transport mode, Apache 2.0 license, Web application program, server programming technique, client programming technique, JSP, HTML, JavaScript, AJAX, Scriptlet expression, Scriptlet, Instruction, Standard action, Expression language, Tag extension

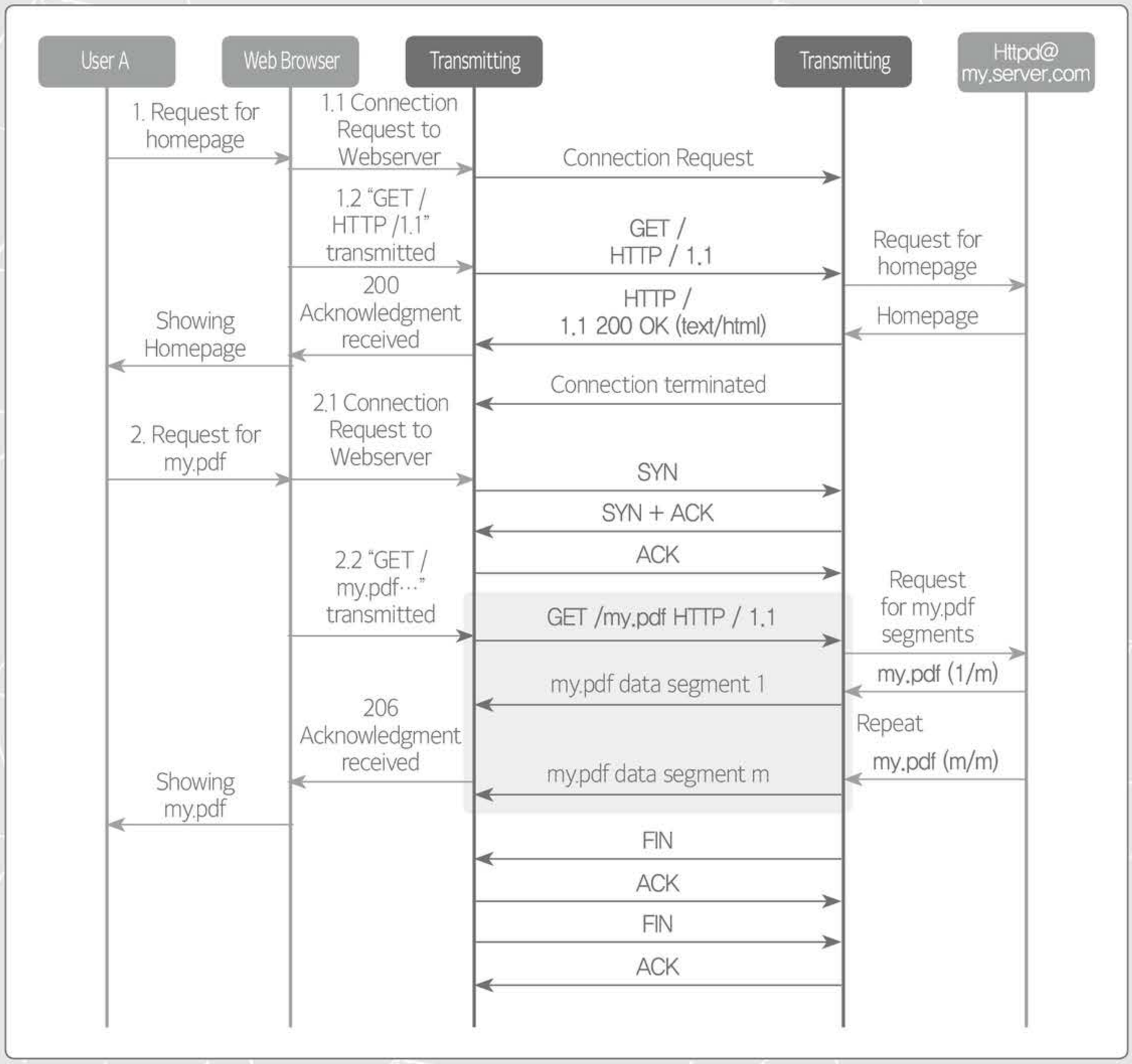
✚ Preview for practical business

[Figure 138] shows the operational path of the application program over the Internet. It shows that application layer programs, like transport layer protocols, operate on the service end-to-end. In other words, like transport layer protocols, such as TCP, the nodes, like wired/wireless routers positioned between end-points, do not affect the operation of transport layer protocols. Although users can generate and exchange them as needed, the most widely used application programs tend to use the internationally recognized standard application layer protocols in order to ensure compatibility.



[Figure 138] Operational Path of Application Layer Protocol

[Figure 139] shows the operational procedure between the application layer and transport layer protocols, according to user A's web use scenario described in "I. Network concept". It shows user A logging into the webserver, my.server.com, and downloading the my.pdf file. Although this section focuses on the operation of the application layer protocol, HTTP, it also shows interaction with the transport layer protocol for better understanding.



[Figure 139] Virtual Scenario of Application Layer Operation

This section describes the part related to the HTTP protocol in [Step 1] and [Step 3-4] of the network operational procedure in user A's web usage scenario ① in "I. Network concept". A web browser and an application program are connected to the web server program, httpd, through a TCP connection and plays the role of fetching the web content that is desired by user A,then converting it into a user-friendly form. The web browser uses the HTTP/1.1 protocol for communication with the webserver. The HTTP/1.1 protocol is a text-based application layer protocol [8]. Text data created, according to the HTTP/1.1 protocol, is delivered to the other application layer program through the transport layer protocol.

The application layer program uses socket programming to transmit and receive user data through the transport layer. Socket programming is a programming technique that is crucial for implementing the application layer protocol, but it is out of the scope of this guide's contents. Refer to other books for more details.

01 What is Application Layer Protocol?

Anyone managing a network system or developing a network application program often encounters application layer technology. The application layer protocol is used to exchange necessary information between application programs. Widely used application layer protocols include FTP [1] for file transfer, telnet [2] for terminal connection, SMTP [3], POP3 [4], and IMAP [5] for e-mail transmission and access, DNS [6] for mapping hostnames and IP addresses, and SNMP [7] for network management. HTTP [8], which is used for data transmission between web clients and webservers, is also an application layer protocol. The BitTorrent protocol is also a widely used application layer protocol, although it is a P2P (peer-to-peer) protocol, not a method of accessing network infrastructure, as conventional communication protocol does.

02 HTTP

HTTP (Hypertext Transfer Protocol) is an application layer protocol for a hypermedia information system that connects text with video and audio files. It has been used for transferring web contents since the early 1990s [8]. The first version, HTTP/0.9, was a simple protocol for low-level data transfer through the Internet, but MIME-type messages have been defined and used since HTTP/1.0 (RFC 1945). HTTP/1.1 is the current version in use and is designed to be more reliable than HTTP/1.0.
Like most standard application protocols, the HTTP protocol implements a request/response protocolthat is suitable for a client-server environment. Since web browsers, such as Microsoft's Internet Explorer, Google Chrome, Mozilla Firefox, and web servers, such as Apache HTTP Server, implement the HTTP protocol, users generally have no difficulty using the web without fully knowing the HTTP protocol implementation. However, when developing an Internet application or a new application protocol, it is helpful to know text-based application layer protocols, such as HTTP, since they are good design and implementation tools.

The HTTP protocol uses a request/response protocol. The client sends a request message, in the form of <request method, URI, protocol version, MIME type information message>, to the server, and the server responds with a status line that includes protocol version, success or error code, MIME type server information, etc. to the client as a response message. Here, the current HTTP/1.1 version is mainly used.
The method, designated in Request-Line of an HTTP request message, specifies the HTTP protocol operation. Although RFC 2616 defines eight methods, they can be extended as needed. The most commonly used methods are GET and POST. The GET method is mainly used to search for content in the server after accessing a website. The POST method is mainly used to deliver user input information, such as a form, to a webserver.
Refer to RFC 2616 for more details on the HTTP response message. The Status-Code information of a status line informs how the HTTP request message has been processed. Codes starting with 2, such as 200 (OK) or 206 (Partial Content), indicate that the request was processed successfully, while codes starting with 4 or 5 indicate client or server errors, respectively.

03 File Transfer Protocol

FTP is a protocol for transferring a file, or part of a file, from one system to another [1]. Unlike other application layer protocols, FTP has two connections. They are control connection and data connection, which use ports 21 and 20, respectively. The control connection is maintained until the client or FTP session is terminated. The data connection establishes a TCP connection when a file transfer starts. The data port number can be a number, other than 20, depending on the method used.

FTP connection is largely divided into general transfer mode and passive transfer mode (PASV). The general transfer mode requests a connection to a client to establish a data connection, and a problem occurs when it connects with an external server system through a system, such as a firewall. In contrast, the client first requests a connection to the server through a channel other than the basic data channel in the passive transfer mode. Since the client requests to establish an FTP connection, it can pass through the firewall without a problem. It is applicable, even when the client's IP is not visible from the outside because the client network uses a proxy server.

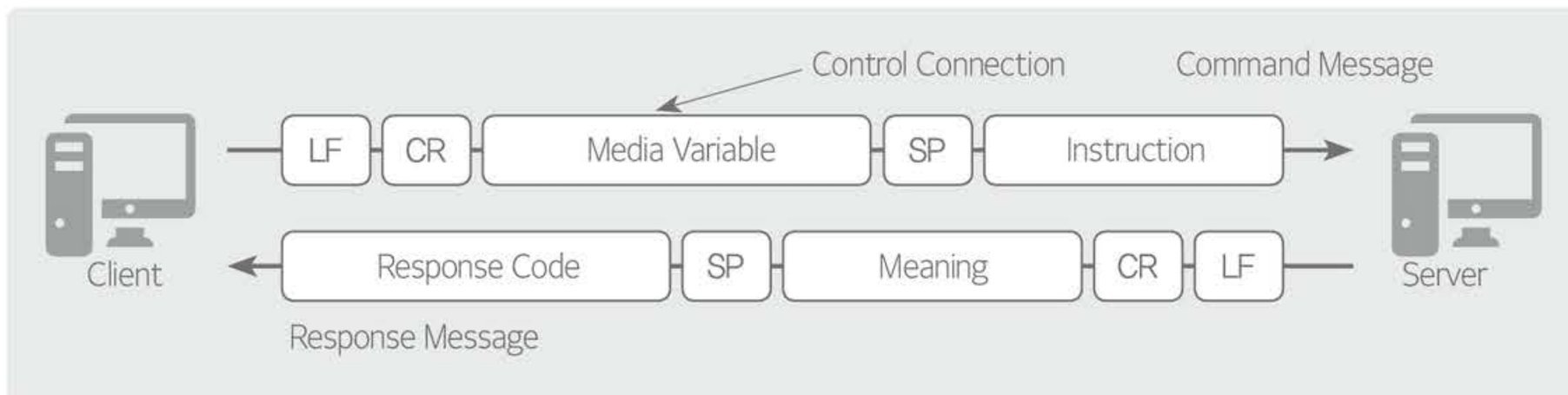
〈Table 71〉 Comparison of General Transfer Mode and Passive Transfer Mode

Type	General Transfer Mode	Passive Transfer Mode
Concept	Data transfer, by a server connecting to the client's specific port	Data transfer, by a client connecting to the server's specific port
Purpose	General FTP use	FTP use from a secured client
Application	Support of Active and Passive of general FTP tools	FTP connection from a web browser
Port used	Control port: 21 Data port: 20	Control port: 21 Data port: 1024 or higher

This section describes the FTP command processing and available FTP commands.

FTP command processing

- FTP uses the control connection to facilitate communication between a server control process and a client control process



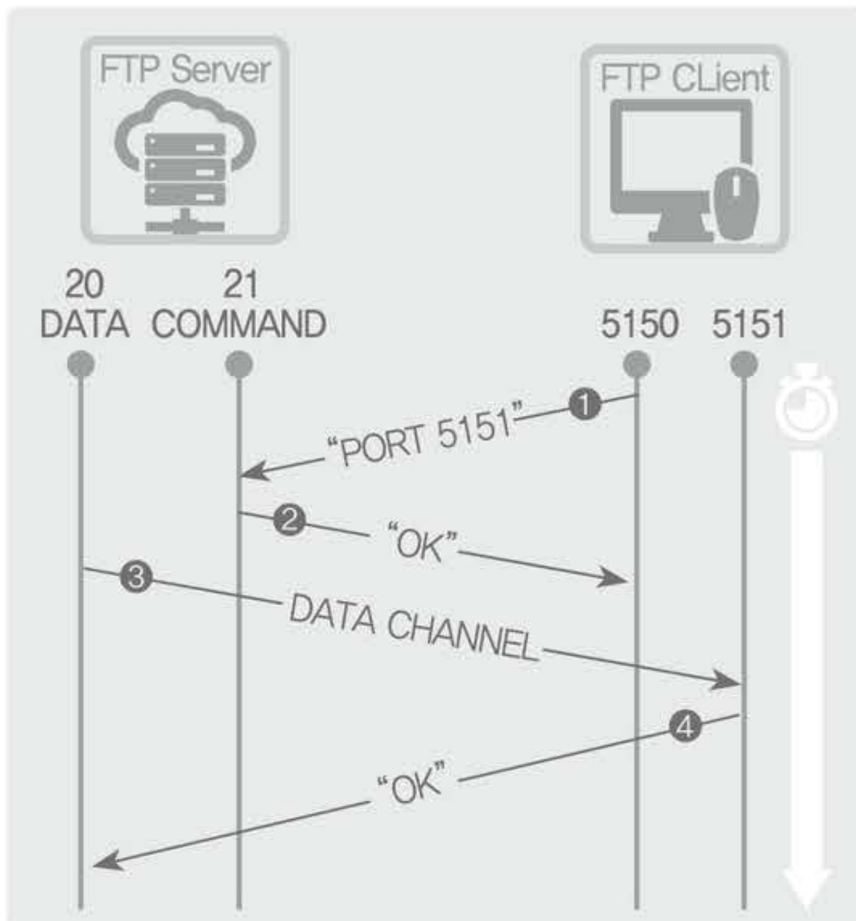
- It is processed interactively and is divided into a command that a user gives to the server as well as a response to it.

Command types included in FTP command messages

Command Form	Command Type
Access command	USER, PASS ACCT, REIN, QUIT, ABOR
File management command	CWD, CDUP, DELE, LIST, NLIST, MKD, PWD, RMD, RNFR, RNTO, SMNT
Data forming command	TYPE, STRU, MODE
Port definition command	PORT, PASV
File transfer command	GET, PUT, MGET, MPUT, RETR, STOR, APPE, STOU, ALLO, REST, STAT
Other commands	HELP, NOOP, SITE, SYST

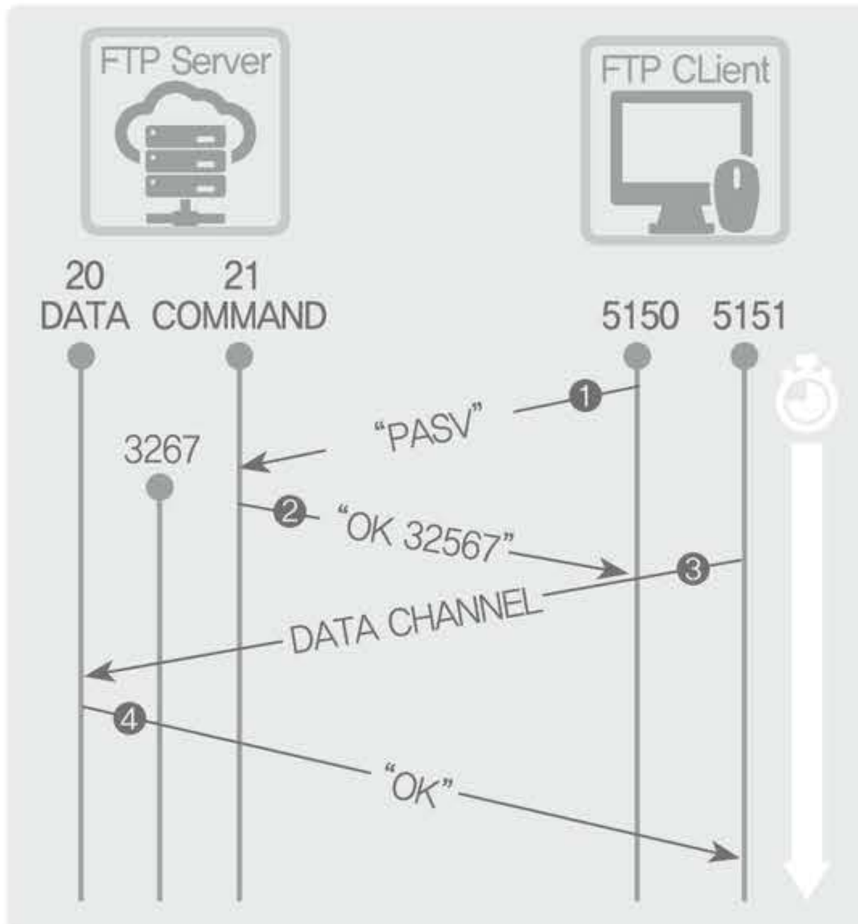
The general transfer mode and passive transfer mode operate in the time sequence shown in the figure below.

General transfer mode operation



1) The client connects to the server's port 21 and informs the server of the second port it intends to use.
2) The server responds with ACK.
3) The server's port 20 attempts to connect to the second port informed by the client.
4) The client responds with ACK.

Passive FTP mode operation



1) The client attempts to connect to the command port.
2) The server informs the second port it intends to use.
3) The client opens another port and attempts to connect to the port informed by the server.
4) The server responds with ACK.

XVI. Mobile Communication and Multimedia Communication Technology

Recent trends and issues

As high-speed data transmission becomes possible and various convergence technologies emerge, cutting-edge technologies are also appearing in the network field. With the advent of technologies that guarantee traffic quality in Internet networks, multimedia and Internet telephony services have been launched. In recent years, Internet of Things (IoT) network-based technologies are expanding together with standardization organizations and groups. Moreover, software-based networks, that are differentiated from the existing network structure, are being introduced. This chapter discusses the recent network trends and major issues.

Learning objectives

1. To be able to understand the concept of multimedia networks and Internet telephony and explain the related protocols.
2. To be able to explain IoT network-based technologies.
3. To be able to explain the concept of software-based networks.

Keywords

Lossless compression, lossy compression, QoS, SIP, H.323, RTP, RTSP, RTCP, IMS, CSCF, HSS, 3GPP, SDN, Openflow, Control plane, Data plane, NFV, Virtualization function, MQTT, CoAP

01 Multimedia Network

A) Graphic compression type

Video data compression, which accounts for most of the traffic in a multimedia network, can be divided into lossless compression (reversible compression) and lossy compression (irreversible compression). Lossless compression, also called reversible compression, refers to restoring a compressed image, without information loss of the original data when decompressing. It is characterized by a lower compression rate than lossy compression.

Lossy compression, also called irreversible compression, refers to a compression method that the restored data, when the compressed data is restored, does not match the original data from before compression because some data is lost.

① Lossless compression

Since the compression and decompression algorithms are the exact opposite of each other, the compression method preserves the original data's integrity, and no part of the data is lost during processing. Leading lossless compression methods include run-length encoding, dictionary decoding, Huffman coding, dictionary coding, and arithmetic coding.

〈Table 72〉 Lossless Compression

Compression method	Description
Run-length encoding (RLE)	• A method expressing consecutively repeated symbols, in terms of the symbol and repetition count (run-length)
Dictionary coding	• A method that creates a dictionary while scanning a message, and if an entry character string in the dictionary is found in the message, the code value (index value) of the entry is sent instead of the string
Huffman coding	• A method that encodes data in a binary pattern and allocates shortcodes to symbols that occur more frequently and longs codes to symbols that occur less frequently
Arithmetic coding	• A method that maps the entire message to a small section in [0,1] and encodes the small section in a binary pattern

② Lossy compression

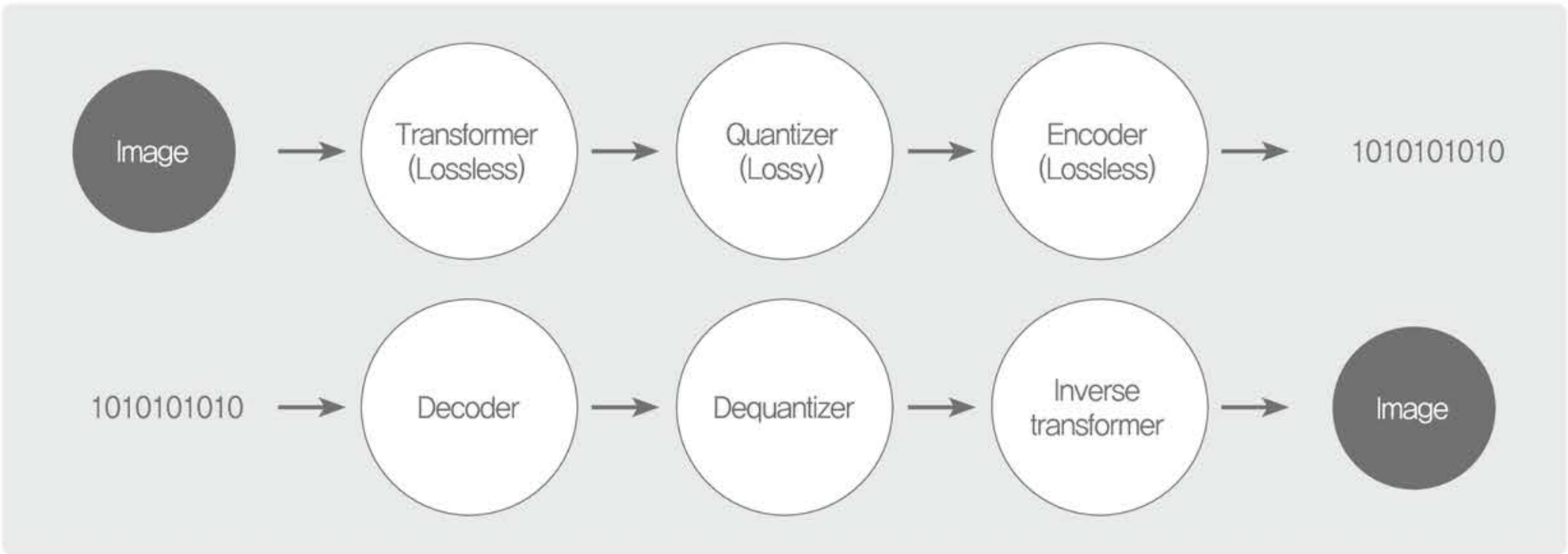
The lossy compression method compromises some accuracy in order to increase the compression rate, by allowing the loss of redundant or unnecessary data. There are two types of lossy compression methods: prediction coding and transform coding. The prediction coding method is used for digitizing the analog signal. Instead of separately quantizing the PCM (Pulse Code Modulation) samples, it quantizes the difference. The difference values are smaller than the actual sample values, so fewer bits are required. The transform coding method transforms a signal, from one domain (mainly time and space domain), to another domain (mainly frequency domain), then compresses it.

B) Multimedia data

Multimedia data includes text, image, video, and audio data. The text has the form of plain text and non-linear hypertext, and the basic language is Unicode for expressing symbols and uses a lossless compression method.

In multimedia,an image is called a still image and refers to a photo, fax page, or video frame. As shown in [Figure 1], an image is transmitted as binary data, through a transformation process, a quantization process, and an encoding process. The data is converted back into an image, through the reverse process.

In the transformation process, the JPEG uses DCT (Discrete Cosine Transform) in the first stage of compression, and the decompression uses the inverse DCT method. The transformation and inverse transformation apply 8 X 8 blocks. The quantization process integerizes the real number of the DCT transform output and converts some values to zero. The coding process arranges data in a zigzag order after quantization and before encoder input, and lossless compression is performed using run-length decoding and arithmetic coding [1].



[Figure 140] Image Data Transformation Process

Video data is composed of several frames. Since each frame is an image, a video file requires a high transmission rate. Video data can be compressed with a spatial compression or a temporal compression method. JPEG performs spatial compression of each frame, and each frame is compressed independently. Temporal compression removes redundant frames and configures each frame into I-frames, which are independent frames, and B-frames and P-frames, which reference other I-frames in order to compress data temporally.

Analog audio signals are digitized using an analog-to-digital converter (ADC), and ADC consists of two processes: sampling and quantization.

C) QoS (Quality of Service)

Methods for guaranteeing QoS in a multimedia network include the RSVP (resource reservation protocol) and the TOS (type of service) field utilization. As shown in [Figure 141], the RSVP method processes data transmission with fixed network bandwidth allocation from the source to the destination. The TOS field utilization designates the TOS field class in each packet to determine the processing priority.

[Figure 141] QoS Utilizing RSVP

RSVP is one of the IETF signaling protocols, and data flow is allocated by a static bandwidth by signaling from end-to-end. It basically uses the method of reserving the queue space [2]. The RSVP message starts from the source device and is delivered to all the receiving devices on the path. It is a path message that contains the necessary information for the receiver and a message sent by the receiver upon receiving the path message. A RESV message reserves the resource of a router that moves toward the source device (upstream) to support RSVP. If a router on the path does not support RSVP, packets are delivered, based on the best-effort delivery method.

The priority method, using the TOS field of IP datagram, checks the 1-byte TOS field, as shown in [Figure 140], and determines the processing priority, according to the set value. The TOS class is divided into eight levels from class 0 to 7, and the higher the number, the higher the processing priority.

Precedence, the first field, indicates the priority or importance of the packet, and the TOS fields indicate whether the network maintains a balance between throughput, delay, reliability, and cost. The last MBZ (must be zero) field is a field that is not currently used.

The TOS field is expressed as a 4-bit binary number. <Table 73> shows the meaning of each value.

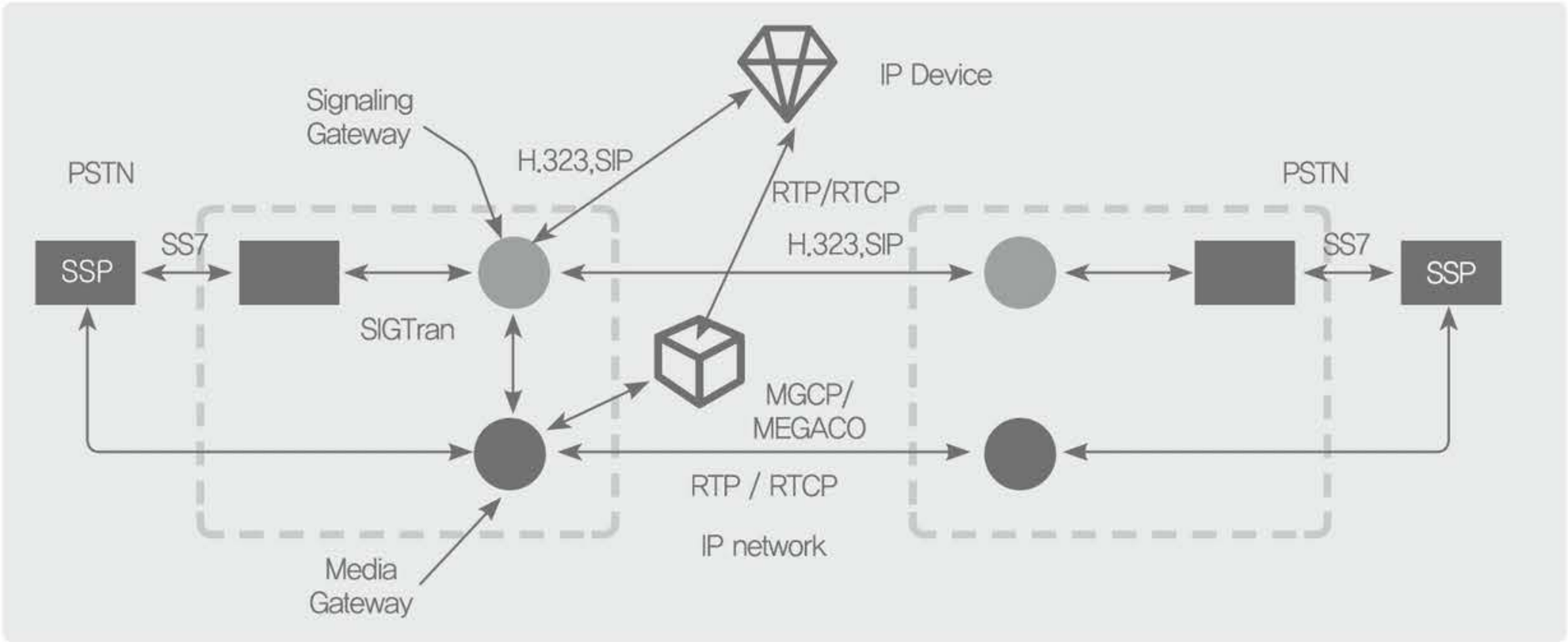
〈Table 73〉 TOS Field Values [4]

Binary Value	Meaning
1000	Priority on delay minimization
0100	Priority on processing volume
0010	Priority on reliability
0001	Priority on cost minimization
0000	General service

02 Internet Telephony Concept and Call Signal Protocol

A) What is VoIP (Voice over Internet Protocol)?

VoIP is a communication technology that facilitates voice calls through packet data, based on the IP network. [Figure 142] shows the VoIP system structure. The main service components are the media gateway and the signaling gateway. The media gateway exchanges multimedia data, delivers the changed date to the corresponding network, and controls it through the media gateway control protocol (MGCP). The signaling gateway performs call signaling, using protocols, such as H.323, SIP, MGCP, and MEGACO, and converts the signals of the PSTN and IP networks, so they can be used with each other.



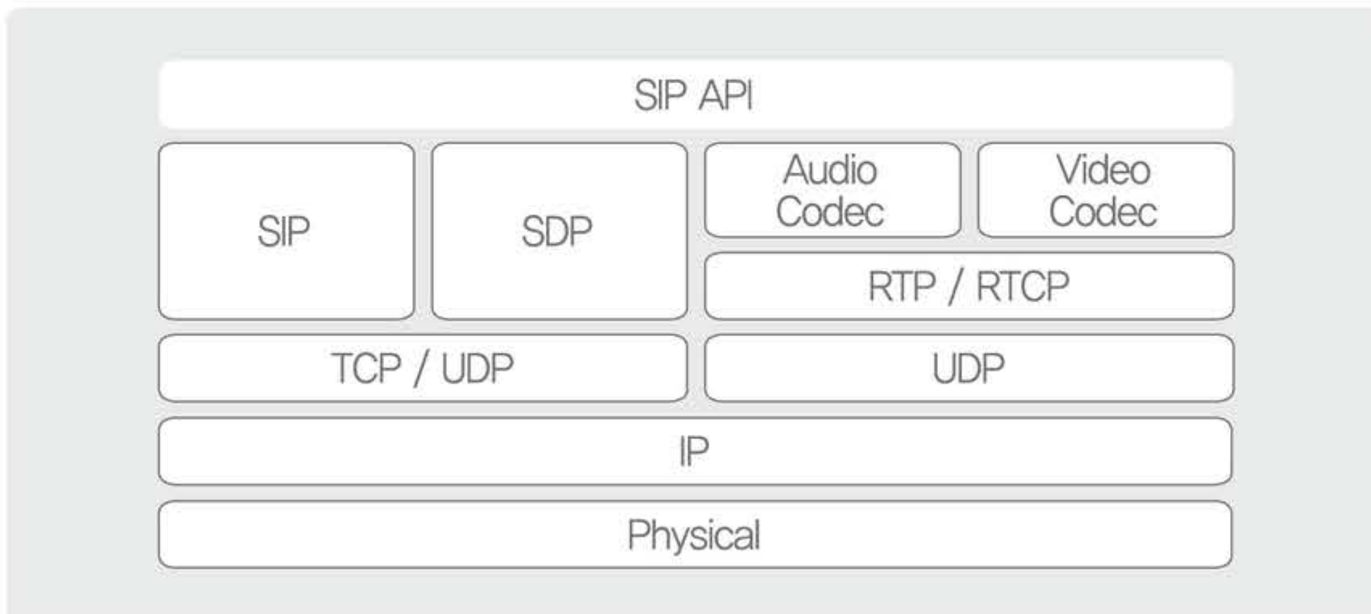
[Figure 142] VoIP System Structure

B) VoIP call signal protocol

VoIP call signaling protocols include SIP (Session Initiation Protocol) and H.323. The SIP protocol is an application layer signaling protocol that specifies a procedure for intelligent terminals that are communicating on the Internet in order to identify each other, find their locations, and create or delete a multimedia communication session between them. The H.323 protocol is an ITU-T standard for providing voice, data, and video services on a LAN that does not guarantee quality. Startup VoIP service providers use it widely because it can provide multimedia services without changing the existing network's underlying structure.

① SIP

SIP is an application layer signaling protocol, used to establish, modify, and terminate multimedia sessions. It operates independently of the lower-layer transport protocols and has scalability as an HTTP text-based protocol.
It controls sessions to enable call control in a circuit-switched data network in a packet-switching network and multimedia applications on the Internet, which is a packet network. It can support text-based addressing methods in URL and E-mail format and message parsing or extension.
[Figure 143] shows the SIP protocol stack located on top of the transport layer protocols TCP and UDP.<Table 74> shows the SIP protocol components.



[Figure 143] SIP Protocol Stack

〈Table 74〉 SIP Protocol Components

Type	Description
SIP	• Definition of RFC 3261 and SIP
SDP	• Session Description Protocol, RFC 4566/3264 • Setting multimedia session parameters
Audio Codec	• G.711A, G.723.1, and G.729A • Voice coding, many specifications for compatibility with various systems
Video Codec	• H.263, MPEG-4, and H.264 • Video coding commonly H.263 and H.264 as the latest version
RTP/RTCP	• Realtime transport (control) protocol, RFC 3550, and RFC 3551 • Real-time communication protocol

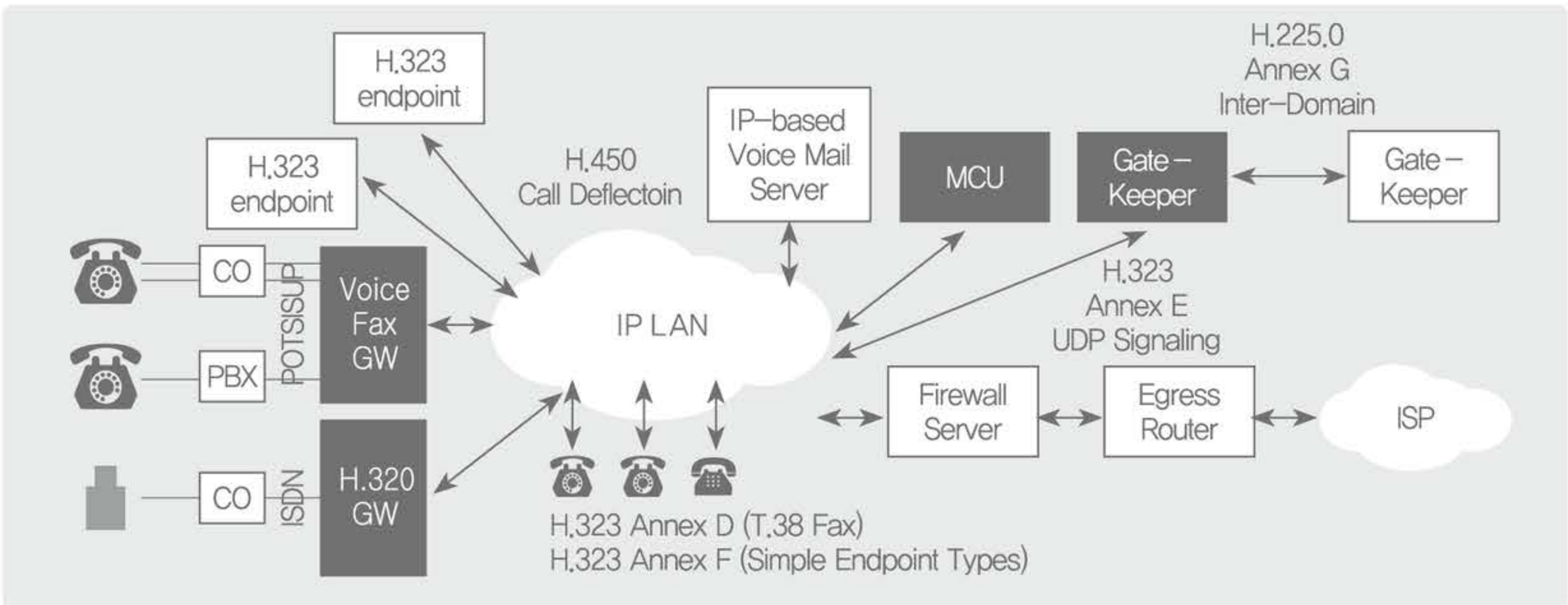
② SIP message structure

SIP uses an SIP uniform resource identifier (URI), similar to an e-mail address, to identify users to provide services, without being dependent on IP addresses. A SIP message contains the START LINE to describe the method type to request and SIP URI, the HEADER to set the value to control the session, the BODY containing the type set in Content-Type, and CR/LF a blank space between HEADER and BODY.

C) H.323

H.323 is an ITU-T standard for providing voice, data, and video services over LAN where quality is not guaranteed. Startup VoIP service providers use it widely because it can provide multimedia services without changing the existing network's underlying structure.

[Figure 144] shows the H.323 network architecture. The service components include terminals, such as general telephones, fax machines, and PCs equipped with multimedia equipment that are used by end users. Other components are a gatekeeper that converts between E.164 and the IP address, performs redirection and authentication for calls, and manages the components and bandwidth, and a gateway that performs the logical connection mapping (encoding, protocol, and call control) of different networks (IP network, PSTN, ISDN, ATM, etc.).



[Figure 144] H.323 Network Architecture

D) VoLTE (Voice over LTE)

VoLTE is a technology that provides data and voice through the LTE network. It provides voice services in the LTE environment, the next-generation wireless broadband network technology.

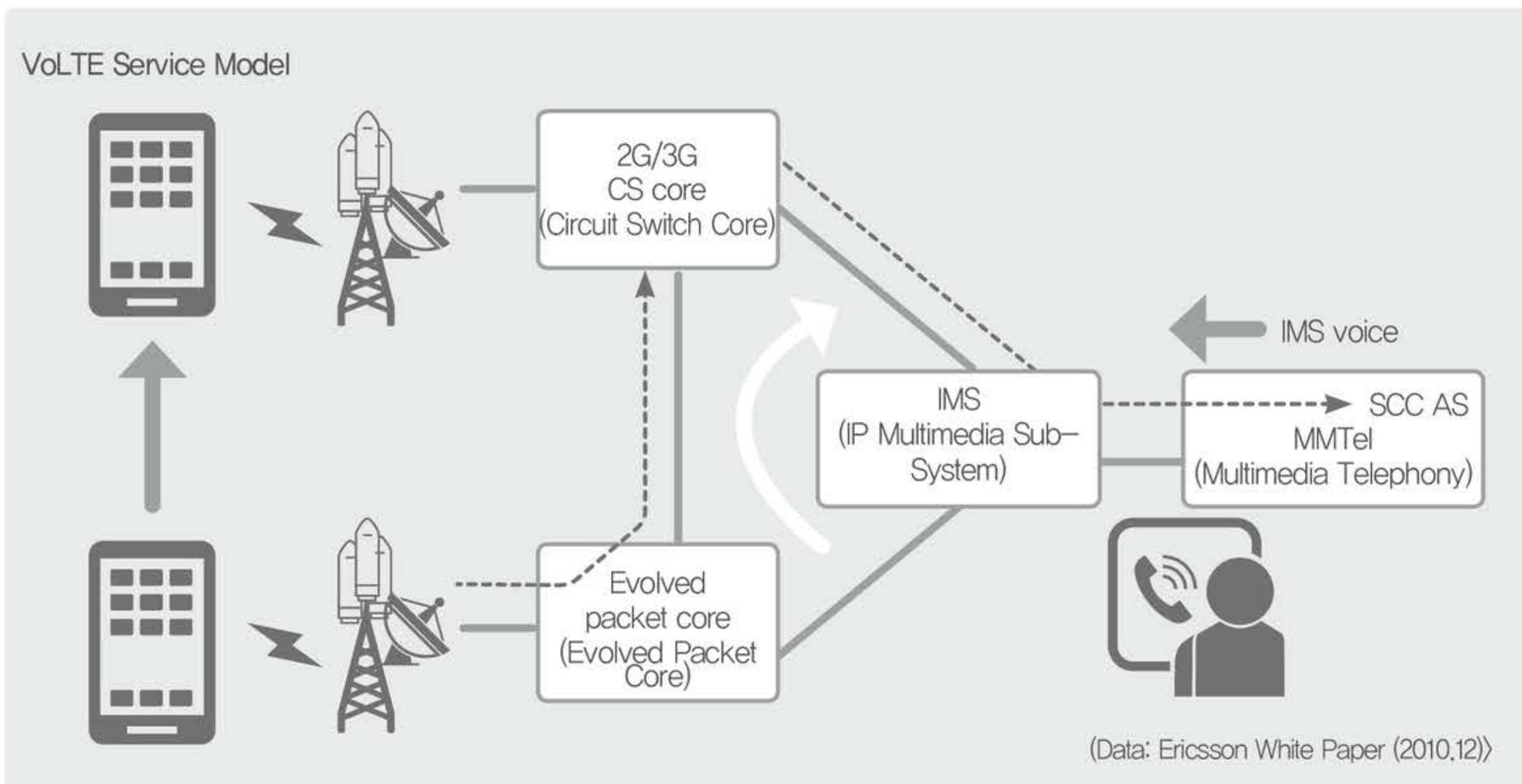
Although VoLTE refers to the voice service domain, since LTE technology is based on an all-IP, data-only network, the VoLTE service is implemented in the form of digital data. Therefore, it transmits voice, text, video chat, and multimedia content in the SIP, which is the main calling solution for Internet telephony services. It provides voice and text services, which exist as data, through the IMS (IP multimedia subsystem) architecture.

These VoLTE technology solutions were not implemented flawlessly at the beginning. It was because the LTE infrastructure was not yet fully equipped, so voice services could not be transmitted over the WCDMA/GSM networks freely. The problem was that while roaming and interconnection were required between LTE and 3G networks to provide seamless communication services, there was no standard agreement on roaming and interconnection between new and old networks. The following two steps can be considered to resolve the problem.

Firstly, the circuit-switched fallback solution establishes calls, mainly over the WCDMA/GSM circuit network, and LTE terminal vendors and mobile telecommunication carriers perform minimal roaming. However, multimedia communication service is not implemented at this stage.

Secondly, if the coverage is established to some extent after the initial stage of LTE network construction, it evolves into the stage of implementing a multimedia communication service, through an appropriate solution, such as 3GPP MMTel (Multimedia Telephony).

[Figure 145] shows how VoLTE services are implemented over 3G networks.



[Figure 145] VoLTE Service Model on 3G Networks

03 Media Transport Protocols

Media transport protocols include RTP (Real-Time Transport Protocol), RTCP (Real-Time Control Protocol), and RTSP (Real-Time Streaming Protocol). RTP is designed to process real-time traffic over the Internet. It is mainly used to transmit video or audio data in real time over the Internet. RTCP is a protocol to control the RTP (Real-time Transport Protocol) for streaming video or audio over the Internet. RTP and RTCP are specified in the IETF standard document RFC 1889. RTSP is designed for use in entertainment and communication systems to control streaming media servers. RTP is used to establish and control media sessions between endpoints, and it does not actually transmit media streaming data. Most RTSP servers transfer actual audio/video data to the transport layer, using the RTP protocol.

A) RTP

RTP is designed to process real-time traffic over the Internet. It is mainly used to transmit video or audio data in real time over the Internet. It is performed over the UDP (user datagram protocol) of the transport layer.

In the RTP protocol, the transmitting side that sends the media file, converts the media unit data compressed with the codec into an RTP packet and sends it to the receiving side, using UDP. Since it is transmitted using UDP, RTP does not guarantee that packets are delivered within a specified time and are not lost. Therefore, the video and audio applications must process packets, using information included in the RTP packet header. RTCP can ensure the QoS of RTP and synchronization of media streams.

B) RTCP

RTCP controls RTP, a protocol for streaming video or audio over the Internet. It is an IETF standard defined together with RTP in RFC 1889. RTCP packet types include the sender report packet, receiver report packet, source description message, bye message, and application-specific packet.

C) RTSP

RTSP is an application layer protocol that transmits real-time media. The communication protocol developed by IETF in 1998 is defined in RFC 2326. It is a communication protocol for transmitting and receiving audio and moving pictures in real time,as well as a standard protocol for controlling streaming data on the Internet. It is used to remotely control media servers. Its commands are similar to VCR operations, such as "PLAY" and "PAUSE", and it accesses the server, based on time information.

D) IMS

IMS was first proposed by the 3GPP (3rd Generation Partnership Project) group that develops international standards in the wireless communication field. It is an infrastructure for providing IP multimedia services and uses SIP protocol-based call control as a core technology.

① IMS concept

IMS is a communication platform defined by 3GPP to control multimedia sessions and provide services, based on the

SIP. It is a core network for providing integrated services in wired/wireless multiple access network environments.

② IMS service goals

IMS provides multimedia content, such as voice, audio, video, and data, based on the IP protocol, and it supports quick service development and change. It uses general-purpose Internet-based technology to improve the price competitiveness of services and enable easy interface with various 3rd party applications, based on an efficient session management function and expansion of business areas, through global linking between services.

③ IMS network structure

The logical structure of an all-IP network is divided into a radio network domain, a GPRS-based packet-switching service domain, and an IP multimedia service domain. The GPRS-based packet-switching service domain can be replaced with an IP network by a packet router outside the 3GPP scope. The wireless domain can be replaced with other wireless network access domains and data network access domains, as Wibro and mobile-LAN that are out of the 3GPP scope. The IMS service domain consists of the call session control function (CSCF) that registers and processes multimedia calls using the IETF's SIP protocol and the home subscriber server (HSS) that integrates the home location register (HLR) function of the legacy mobile network with the function to mobility management and authentication of IP multimedia users.

④ IMS's role in the convergence aspect

- ID and authentication system in the convergence environment, using the IMS infrastructure
- Bidirectional channel for service control
- Service session connection, through the independent service infrastructure

XVII. Latest Technologies

Recent trends and issues

As high-speed data transmission becomes possible and various convergence technologies emerge, cutting-edge technologies are also appearing in the network field. With the advent of technologies that guarantee traffic quality in Internet networks, multimedia and Internet telephony services have been launched. In recent years, IoT network-based technologies are expanding together with standardization organizations and groups. Moreover, software-based networks, differentiated from the existing network structure, are being introduced. This chapter discusses the recent network trends and major issues.

Learning objectives

1. ITo be able to explain IoT network–based technologies.

To be able to explain the concept of software–based networks.

Keywords

Lossless compression, lossy compression, QoS, SIP, H.323, RTP, RTCP, IMS, CSCF, HSS, 3GPP, SDN, Openflow, Control plane, Data plane, NFV, Virtualization function, MQTT, CoAP

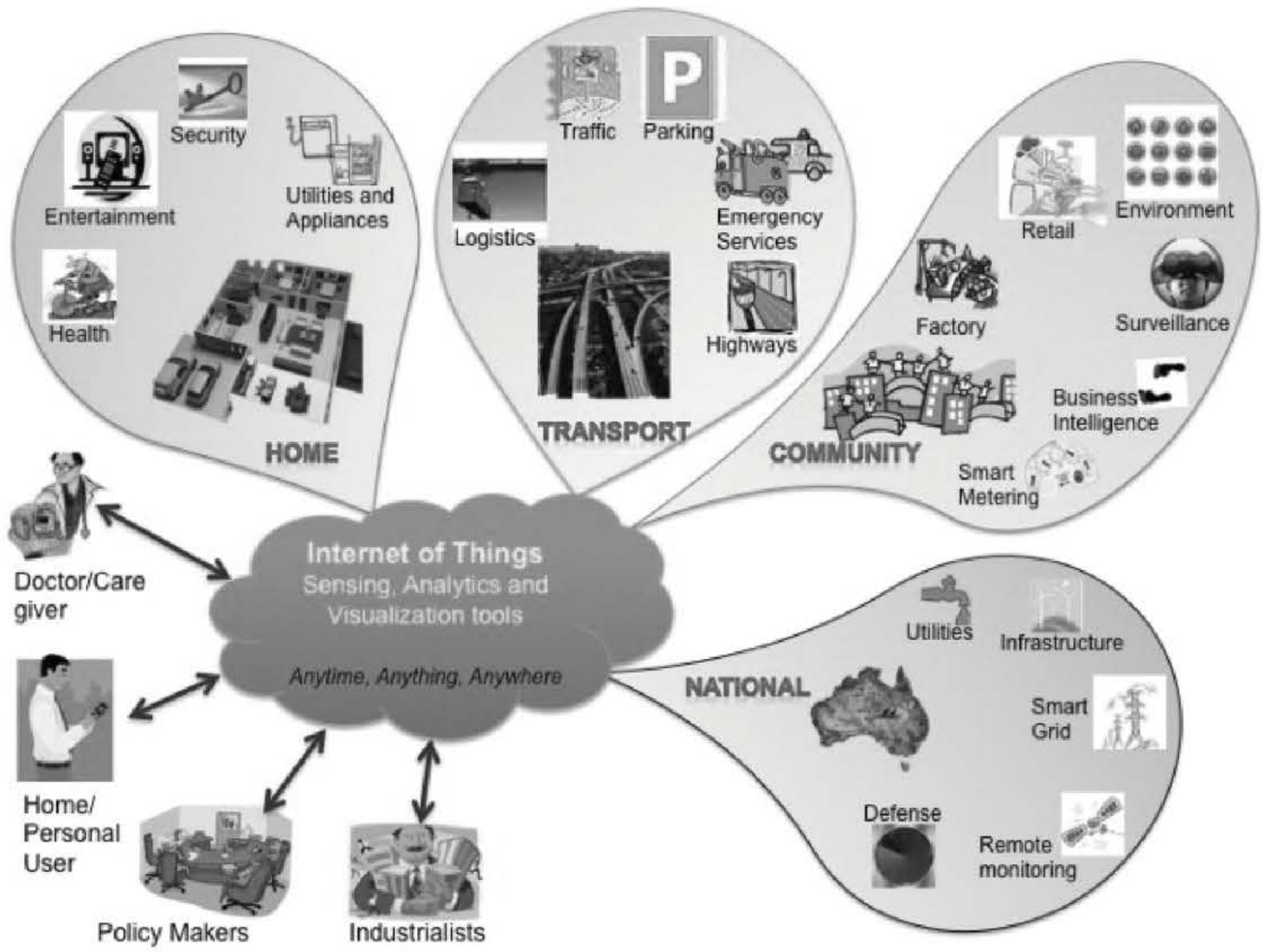
Practical preview

Network technologies are changing with new technologies, such as IoT, artificial intelligence (AI), and cloud computing. The recent topic of the new network technology is the software-defined network (SDN). SDN is a technology for standardizing software that is dependent on existing hardware manufacturers and providing flexible network services. It is the core technology of a software-defined data center (SDDC). Cloud computing is another core technology of SDDC. As such, the network is the infrastructure and core of new technologies. Even in AI, the network performance determines the AI performance. Therefore, the AI network uses InfiniBand, which is a short word for infinite bandwidth. The existing network Ethernet can communicate with one device at a time, but InfiniBand can simultaneously handle communication between multiple devices. It can reliably operate, even if there is a problem with some equipment. This section discusses new network technologies that have recently become a hot topic.

01 IoT Network-Based Technology

A) IoT concept

IoT (Internet of Things) is a concept that embeds sensors and communication functions in various objects to connect to the Internet. It refers to the technology that can perform sensing, networking, and information processing, through cooperation, without explicit human intervention. At the same time, the ubiquitous technology that allows access to information and communication networks, regardless of time and place as a concept that exists anywhere at the same time and is ubiquitous and the machine-to-machine (M2M) technology that enables intelligent communication between objects have evolved further. As such, IoT enables all things to exchange information through the Internet, without boundaries between people and objects.

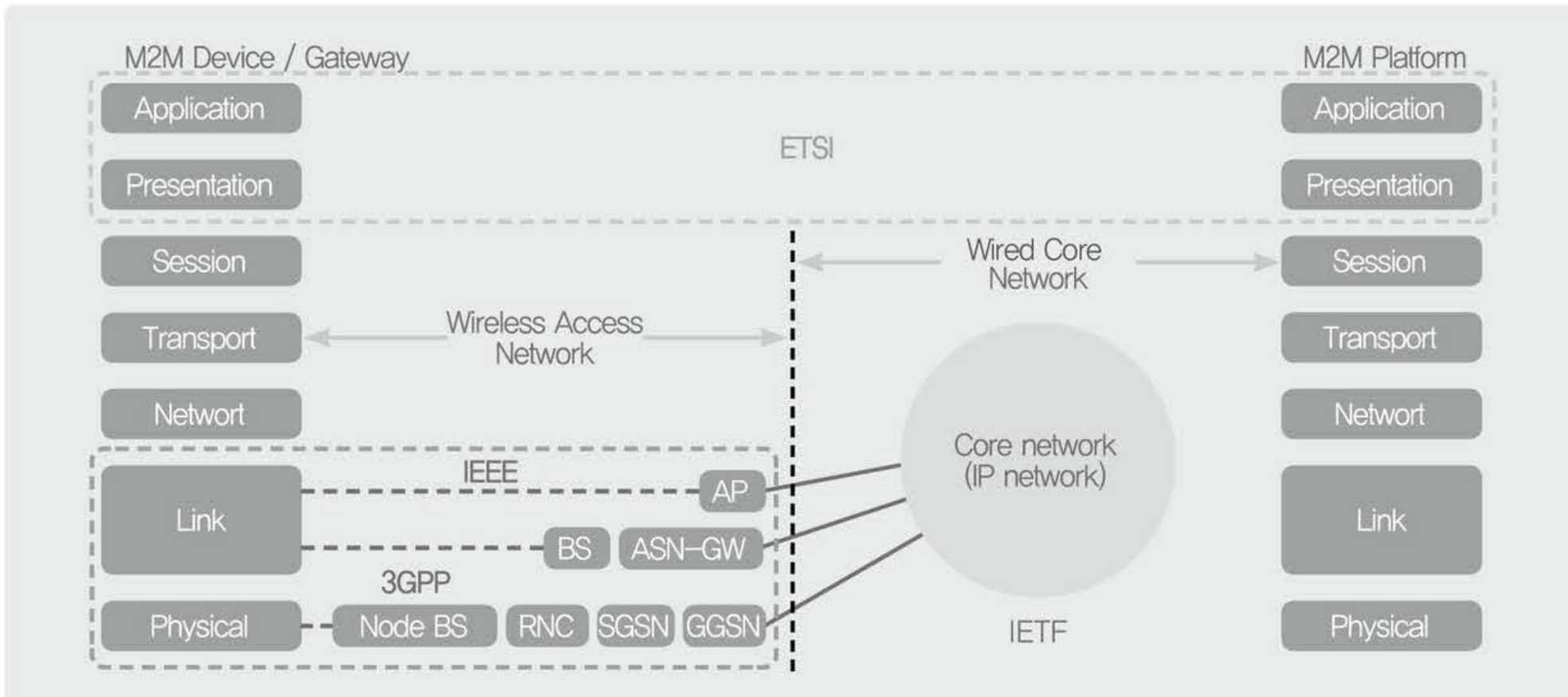


[Figure 146] IoT Applications

B) IoT standardization trend

3GPP and ETSI have established mobile communication-based IoT standards, but since 2012, they have been led by an international consultative body called, oneM2M. The Open Mobile Alliance (OMA) proposed a communication method, based on a server-client structure, by establishing the Lightweight M2M (LWM2M) protocol. oneM2M has released LWM2M, which applies CoAP (Constrained Application Protocol) and HTTP for data transfer, as the latest standard.

[Figure 147] shows the standard scope of each standardization organization for protocol layers.



[Figure 147] Standard Scope of Each IoT Standardization Organization [10]

C) Core IoT technologies

IoT's main technologies are a sensing technology, a wired and wireless communication and network infrastructure technology, and an IoT service and interface technology.

① Sensing technology

Sensors collect object information. Various sensors, such as temperature, humidity, heat, gas, illuminance, and ultrasonic waves, are installed in the object to obtain information about the surrounding situation. Sensors are needed to minimize power consumption in order to operate for a long time. Physical sensors have evolved into smart sensors with a standardized interface and information processing capability, and they include a virtual sensing function that extracts specific information from the sensed data. As the sensor layer's role has increased, the hardware platform market is evolving to facilitate connecting various sensors, control, and communication, using open-source hardware (OSHW).

② Wired/wireless communication and network infrastructure technology

Although objects generally use wired means, such as Ethernet or power line communication (PLC), to access the Internet, wireless communication is more efficient when considering the installation convenience and mobility. Short-range wireless communication, such as WLAN, Bluetooth, ZigBee, and UWB, and mobile communication, such as 3G and LTE, are available. Moreover, new technologies for sensor networking are constantly being proposed.

- BLE (Bluetooth Low Energy): Also called Bluetooth Smart, it is low-power, short-range wireless communication that is not compatible with existing Bluetooth but is compatible with Bluetooth 4.0.
- Z-Wave: It is a protocol for devices that use intelligent mesh network topology, do not have a master node, and require low power and low bandwidth.

③ IoT services and interface

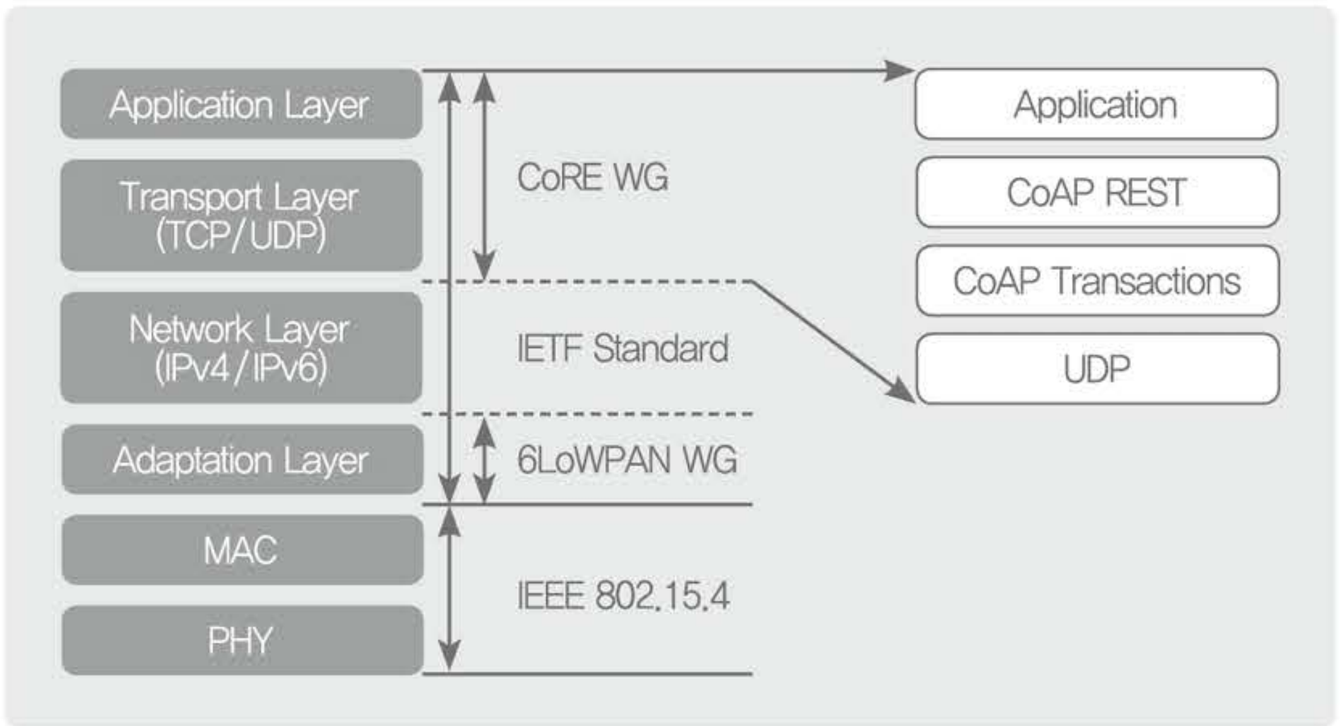
It is necessary to use ontology-based semantic web technology, cloud computing for large-scale distributed processing, and open API for accessing various services, to automatically analyze and share data collected from the sensor of an object.

D) Main IoT protocols

Protocols for IoT require lightweight the most, and since they have been commonly used for some time, protocols that feature compatibility and scalability are preferable. In particular, CoAP and MQTT are suitable for small devices that need to connect to the Internet and transmit data, due to the nature of IoT, and are proposed as communication protocols in various architectures.

① CoAP (Constrained Application Protocol)

CoAP is a lightweight application layer protocol, developed by the IETF CORE Working Group, for communication between objects. Although it uses the UDP transport layer above the IP layer, it is designed independently of the lower layers and can operate in other networks and transport layers. It has reduced the message size to reduce end-point load and uses binary encoding to facilitate encoding and decoding.

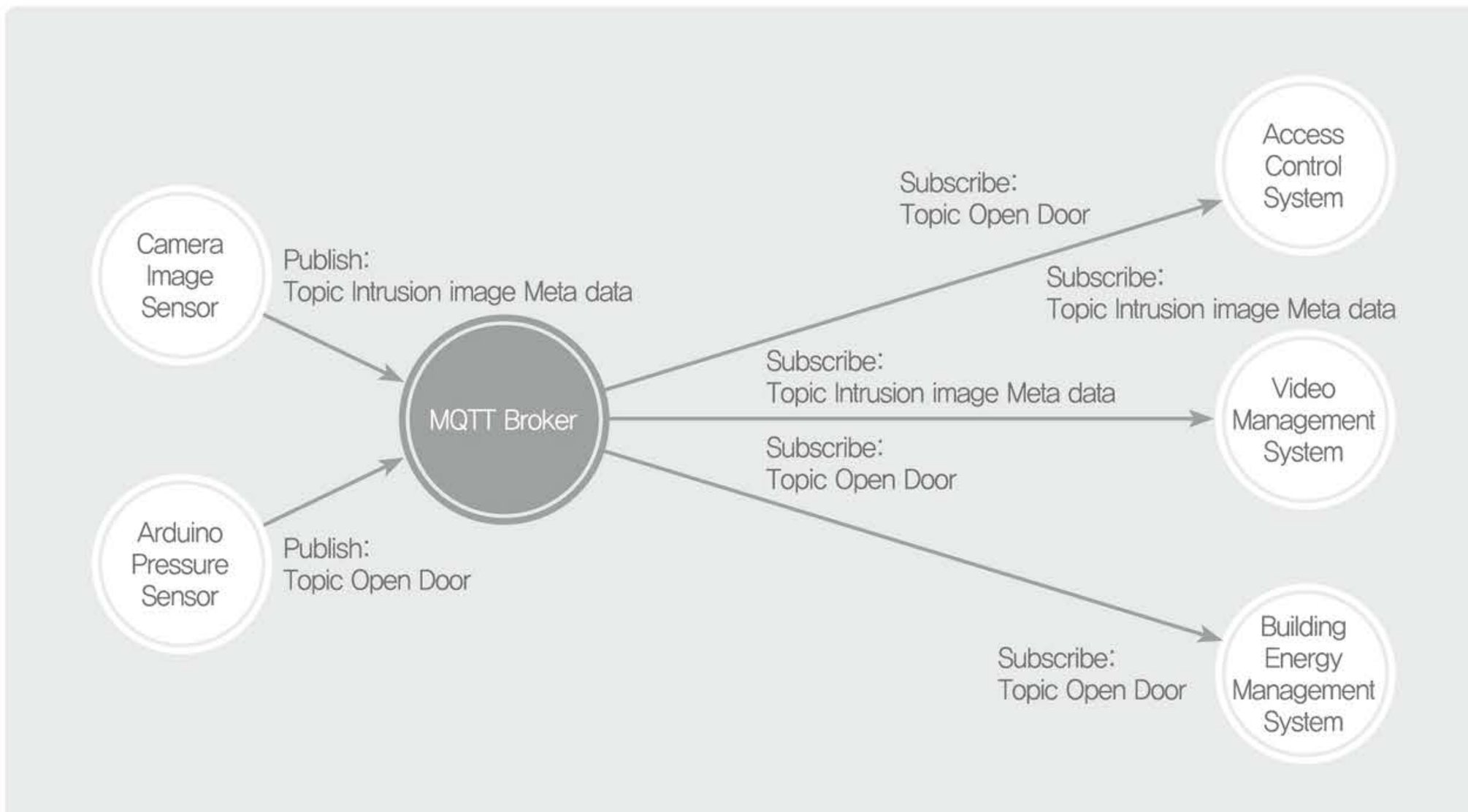


[Figure 148] CoAP Conceptual Diagram

CoAP has attracted attention as the main protocol for IoT, since it communicates with endpoints using Zigbee, rather than fast and stable Ethernet or Wi-Fi.

② MQTT (Message Queue Telemetry Transport)

MQTT is a publish-subscribe-based protocol for delivering lightweight messages at low speed in high-latency and unreliable networks.



[Figure 149] MQTT Applications

MQTT is a flexible data transfer model between multiple endpoints and multiple servers and operates by publishing and subscribing to necessary topics. [Figure 149] shows a conceptual diagram of transmitting the topic of the door opening and closing, by sending pressure sensor data to the access control system, video security system, and building energy control system, using Arduino and the metadata topic that is generated by the camera's image sensor to multiple systems, using MQTT.

Although the protocols based on CoAP and HTTP are widely used currently, the standardization organizations are still suggesting architectures using MQTT and REST. OASIS continues to update the MQTT standard. [14]

〈Table 75〉 Comparison of MQTT and CoAP

Type	MQTT	CoAP
Purpose	Message protocol for IoT	Message protocol for IoT
Topology	N:M type	1:1 type
Configuration	A broker and multiple clients	Server and client
Operation	Publishing and subscribing	Requesting and responding
Data	Event	Status Information
Transmission protocol	Mostly TCP	Mostly UDP
Standard	OASIS standard	IETF CoRE standard

02 Software-Based Network

The technologies that have recently received the most attention in the network field are software-defined networks (SDN) and network function virtualization (NFV). Network openness and virtualization technologies are expected to change the era when certain large equipment makers dominated the global network equipment market and increase 3rd party software providers and equipment makers to enter the market.
Korea has planned smart Internet-related R&D and standardization projects focusing on SDN and NFV to enhance national SDN technology competitiveness, led by ETRI, since 2012. Organizations, such as the TTA Future Internet Project Group (PG220) and the Smart Internet Technology Standardization Council actively promote domestic standards and reflect international standards.

A) Limitations of the existing communication environment and paradigm shift

① Changes in traffic patterns

Current applications generate a large volume of traffic, by accessing various application servers and databases before providing users data. The environment is changing from the limited communication between the client and the server to access between various systems.

② Virtualization technology

Virtualization technology has dramatically increased the number of servers connected to the network, fundamentally changing the assumptions about the host's physical location.

③ Networks getting more complex

Network structures are becoming more complex, as a large number of computers are configured of discrete protocol sets.

④ Difficulties in network design and management

Initial designs use a link oversubscription method that considers the expected traffic patterns and the number of concurrent users. However, it is becoming more difficult to predict the network size, as traffic patterns change dynamically.

⑤ Increasing dependence on equipment manufacturers

In particular, the core network has had difficulties in introducing new services or technologies as certain major equipment manufacturers dominated the communication market.

B) SDN (Software Defined Network)

SDN is a next-generation network technology that can set the network routing and control and complex operation management through software programming. The technology attracted attention when ONF introduced the OpenFlow technology, and international standardization organizations, such as IETF and ITU-T, began to develop standards for SDN-related core technologies. There is heated competition in the industry for securing leadership in

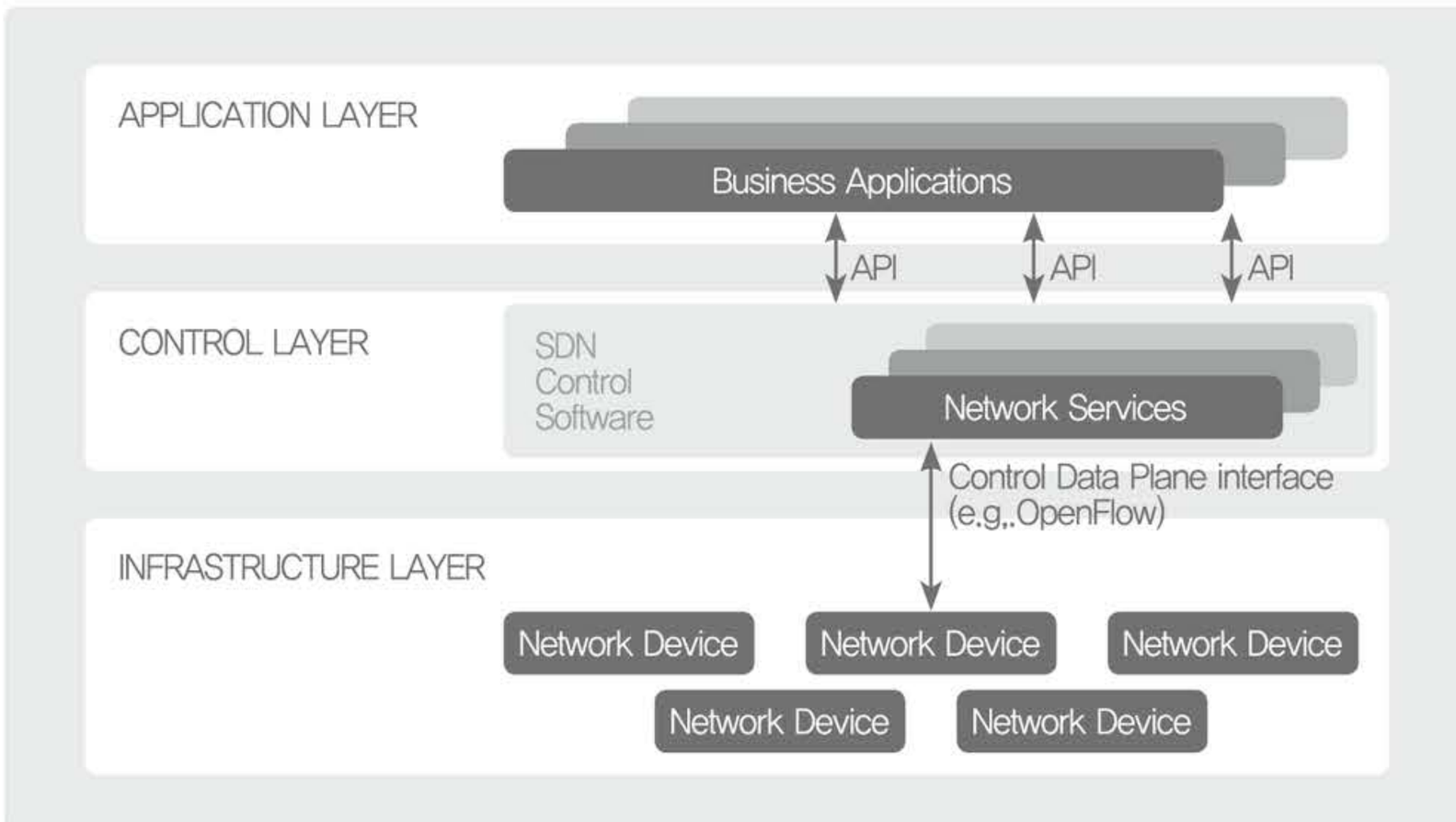
SDN technology standards.

① Introduction of SDN

SDN was first proposed at the Open Network Summit, a conference specializing in SDN and open flow, held at Stanford University in October 2010. SDN and OpenFlow are technologies that program and configure or operate networks, like computers. They have attracted attention as new innovative communication environmental technologies. In March 2011, ONF (Open Networking Foundation) was formed to promote and standardize OpenFlow and SDN.

② Concept of SDN

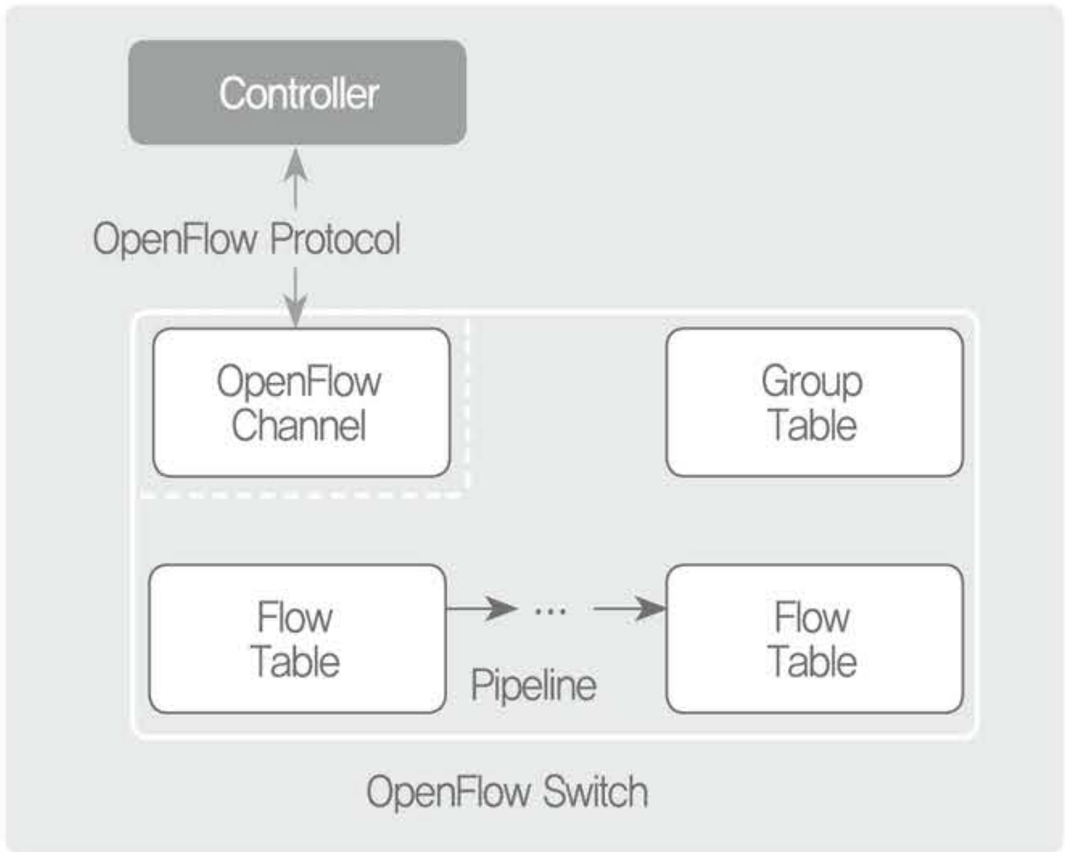
As shown in [Figure 150], SDN enables programming networks by separating the control plane (to guide the network to which it goes) from the data plane (to send packets to a specific destination). The basis of SDN is a switch that uses industry standard control protocols, such as OpenFlow, and can be programmed through an SDN controller.



[Figure 150] SDN Architecture [5]

③ OpenFlow technology

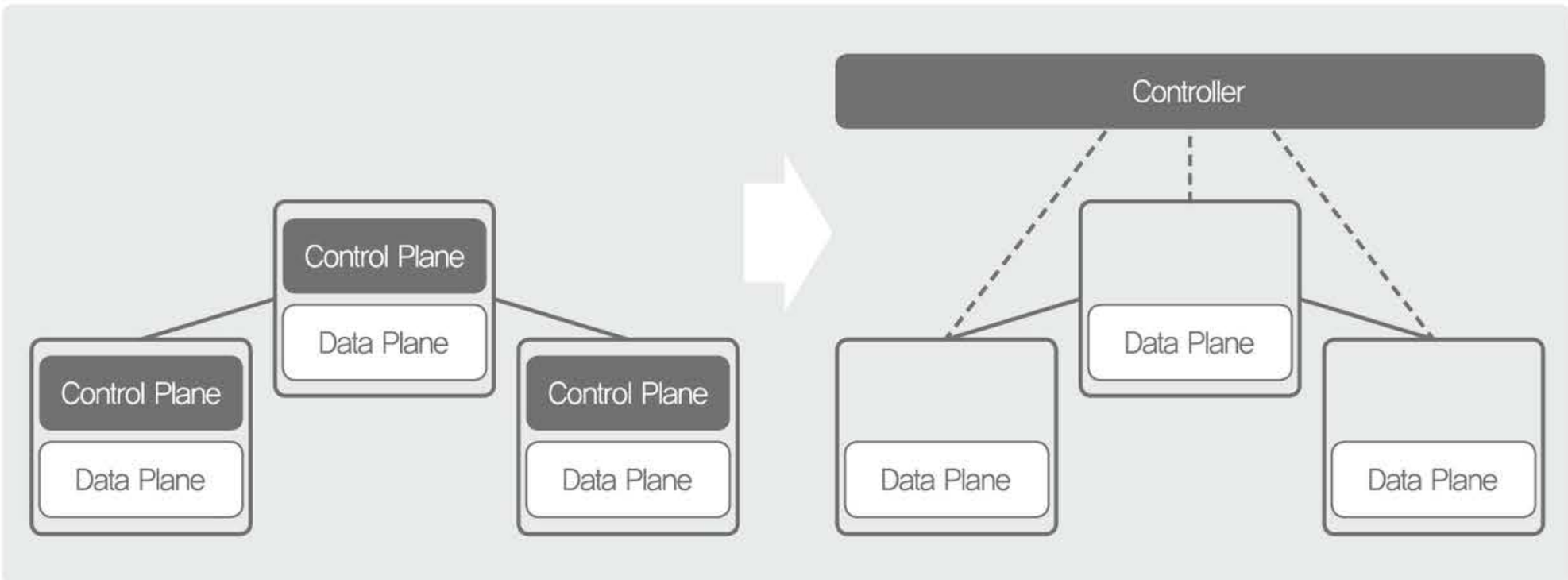
As shown in [Figure 151], the OpenFlow technology separates the functions to control and transfer the packets and controls the network through programming. OpenFlow consists of a controller and a switch. The controller issues commands to the switch, and the switch performs tasks related to data flow, such as sending or modifying packets to the destination, according to the commands.



[Figure 151] Main Components of OpenFlow Switch [6]

④ SDN applications

As shown in [Figure 152], network devices have had a control plane and a data plane implemented in one device until now. However, when SDN is applied, the controller performs the control plane function of each network device, and only the data plane function remains in the actual physical network device.
Here, OpenFlow plays a role as an interface standard technology for SDN operation.



[Figure 152] Network Structure before/after SDN Application

For example, it removes computational operations in medium and large routers and leaves only a simple switching function. Each router simply operates according to a command issued by the controller.

C) NFV (Network Function Virtualization)

NFV has emerged as a new topic, thanks to the high interest of network operators. Moreover, the NFV Group is preparing standards for major telecommunications operators and equipment manufacturers and is promoting a PoC (Proof of Concept) to check the feasibility of the technology.

① Background of NFV

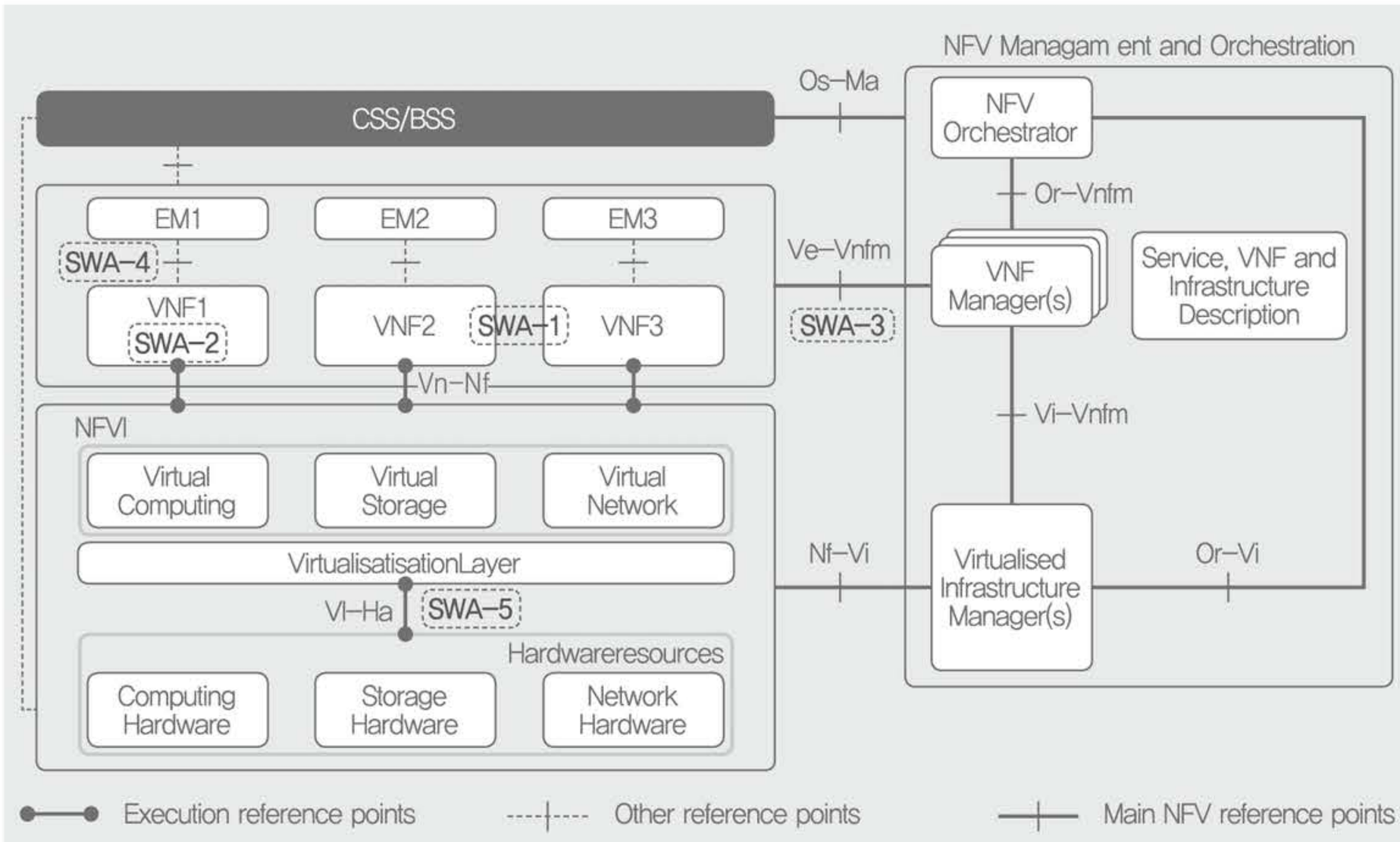
With the increase in network speed and the emergence of various services, securing space and power to accommodate the rapidly increasing network equipment has become a crucial issue for Internet service providers. Moreover, the lifespan of network equipment is shortened, making it increasingly difficult to secure profits after introducing new equipment. NFV was introduced to solve these problems through virtualization technology.

② Concept of NFV

NFV runs on a high-performance x86 platform and allows users to activate the necessary network functions as needed. It creates a VM (Virtual Machine) or service profile and implements virtualization on the network by utilizing the x86 capabilities.

③ NFV architecture framework structure

As shown in <Table 76>, the NFV architecture framework consists of three function groups: VNF group, NFV infrastructure, and management and orchestration.



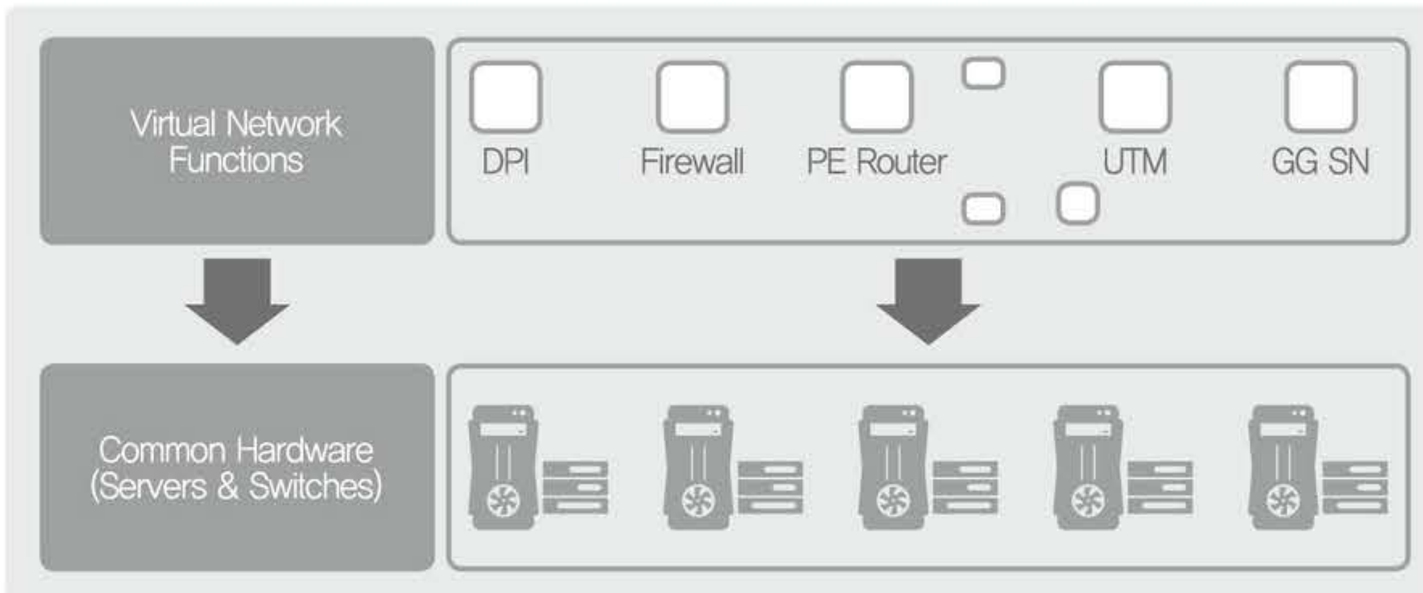
[Figure 153] NFV Architecture Framework (Five VNF Interface Types) [7]

〈Table 76〉 NFV Architecture Framework Configuration

Type	Description
VNFs	• Virtual network functions • Network feature set software to support multiple application programs
NFVI	• NFV infrastructure • Provides physical hardware resources to support computing, storage, and network functions as well as functions to support virtualization and VNF execution
Management & Orchestration	• Provides hardware and software resource management, delivery, and VFN management functions

④ NFV application cases

NFV uses virtualization to implement network equipment functions. It not only operates on specialized equipment, but also implements the network equipment functions in a general high-performance x86 server platform, as shown in [Figure 154].

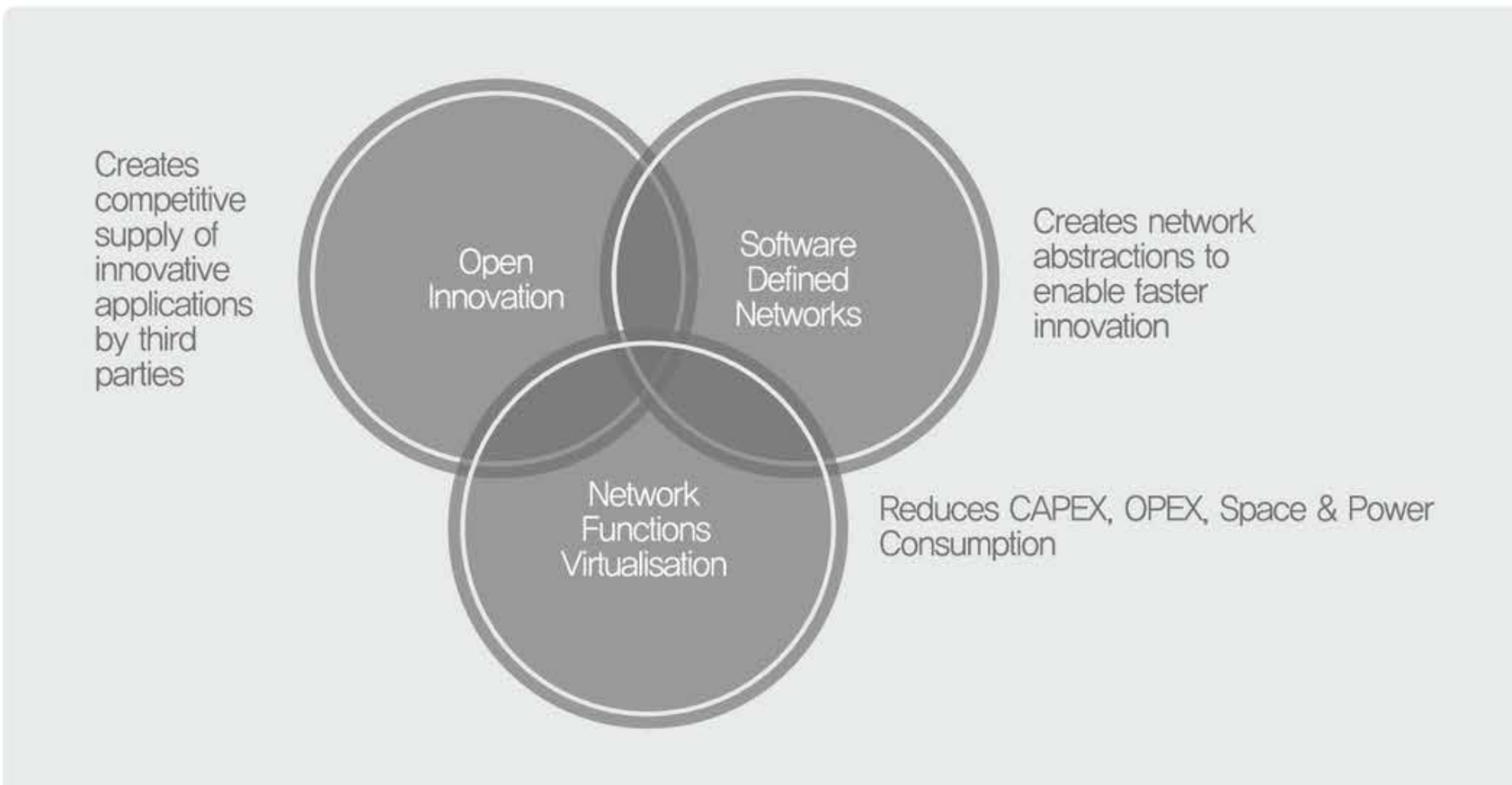


[Figure 154] NFV Implementation

D) SDN and NFV

Initially, SDN was developed mainly by researchers and data center designers, and NFV was developed by Internet service providers (ISPs). However, SDN and NFV technologies are complementary to each other, and ISPs currently combine the two technologies and apply them to networks.

① Relationship between NFV SDN



[Figure 155] Relationship between NFV SDN [8]

SDN enables faster change by creating an abstracted network, and NFV reduces capital expenditures (CAPEX), operating expenditures (OPEX), and space and resource consumption. NFV and SDN have a complementary relationship and can be independently configured.

② Comparison of NFV and SDN

Classification	SDN	NFV
Technology objective	Software implementation of network functions, through a separation of control and data and centralization of network control	Implementation of existing dedicated network devices, using the VM and high-performance x86 server platform
Target location	Although the technology began at campuses, data centers, and cloud services, telecommunication carriers have recently used it.	The technology targeted network devices of communication operators from the beginning.
Target equipment	Network equipment, such as middle to large routers and switches	Network equipment, such as middle to large routers and switches
Implemented functions	Cloud orchestration and networking	Router, firewall, gateway, CDN, WAN accelerator, and SLA compensation
Protocol	OpenFlow centered	None
Leading organization	ONF	ETSI NFV Working Group