TCP/IP **基礎知識**

Mastering TCP/IP　Ch. 2

by r1cky [KUMO]

2.1 History of TCP/IP

- Most Used protocol in computer networks
- Many OSes support it

2.1.1 Military applications

- 1960s, Department of Defence research
- Needed resilleient network to route data despite partial damage
- Achieve data transfer by packets
- Packets allow sharing network routes by many users
- Enable network utilization, lower costs

2.1.2 Birth of ARPANET

- Experimental network constructed in 1969
- Nodes within 4 universities
- Expanded to 34 nodes in 3 years
    - UCLA
    - UCSB
    - SRI (Stanford)
    - Utah U
- Proved real-life efficacy of packet communication

2.1.3 Birth of TCP/IP

- Developed in early-1970s
- Specification decided in 1982
- Became ARPANET's exclusive protocol

2.1.4 Spread of UNIX and Internet

- BSD UNIX used in many universities around 1980
- TCP/IP implemented
- TCP/IP ratified as official connection method for ARPANET in 1983
- Sun Microsystem began offering commercial products in same year
- More institutions connect to ARPANET, later NSFnet

2.1.5 Beginning of Commercial Internet Serivce

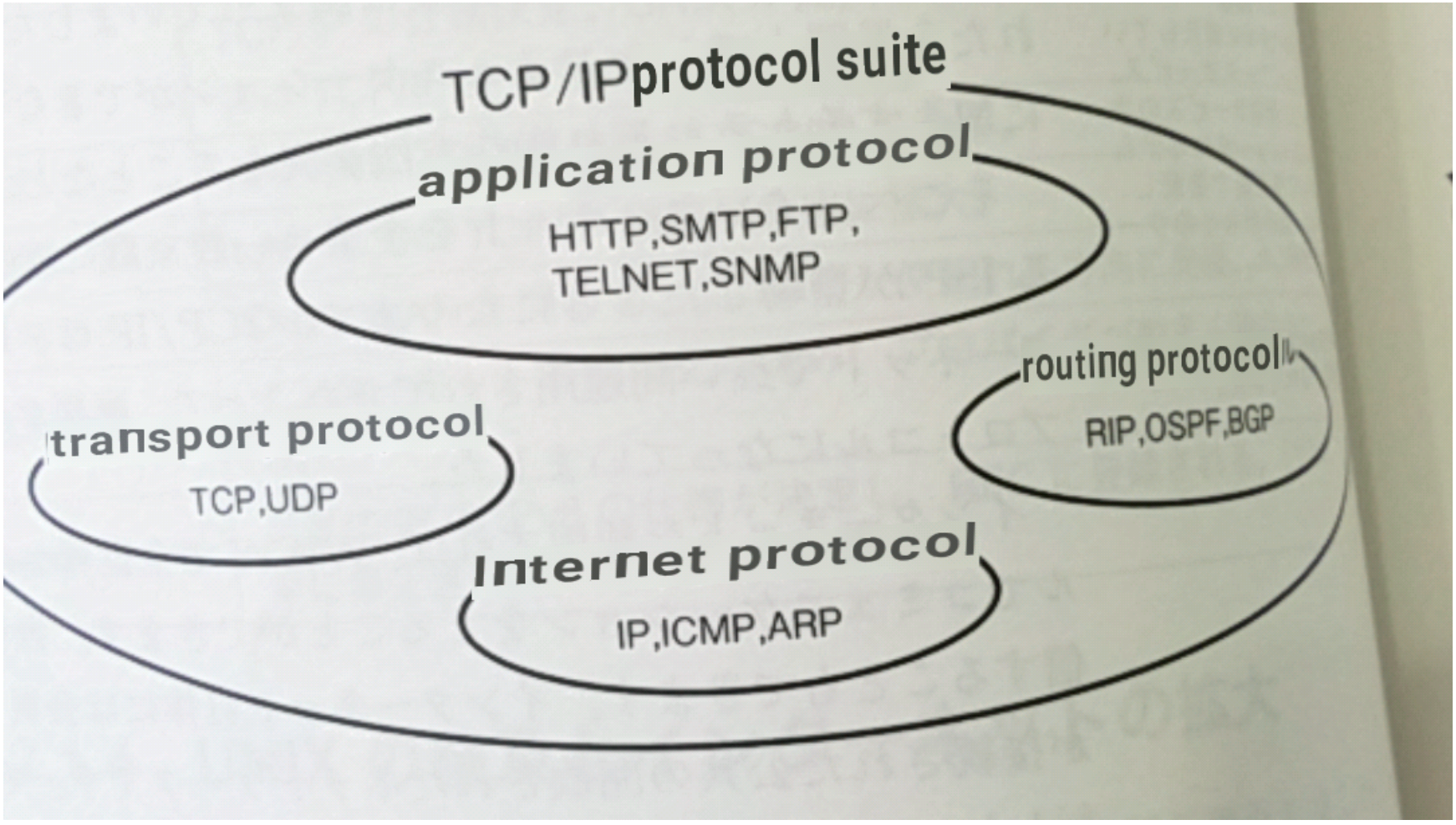
- ISPs and services appeared around 1990s
- PC通信 (Host-2-Client) dominant before then
  - communication limited to members
  - each subsciption offered different usability
- Internet allowed commerical use
- With Internet, you can...
  - collect info with WWW
  - communicate with email

2.2 Standardizing TCP/IP

- ISO standardized OSI protocol in 1990s
- TCP/IP became more popular

2.2.1 What is TCP/IP?

- TCP/IP indicates TCP and IP protocols
- In most cases, TCP/IP is a suite of protocols



2.2.2 The Spirit of Standardizing TCP/IP

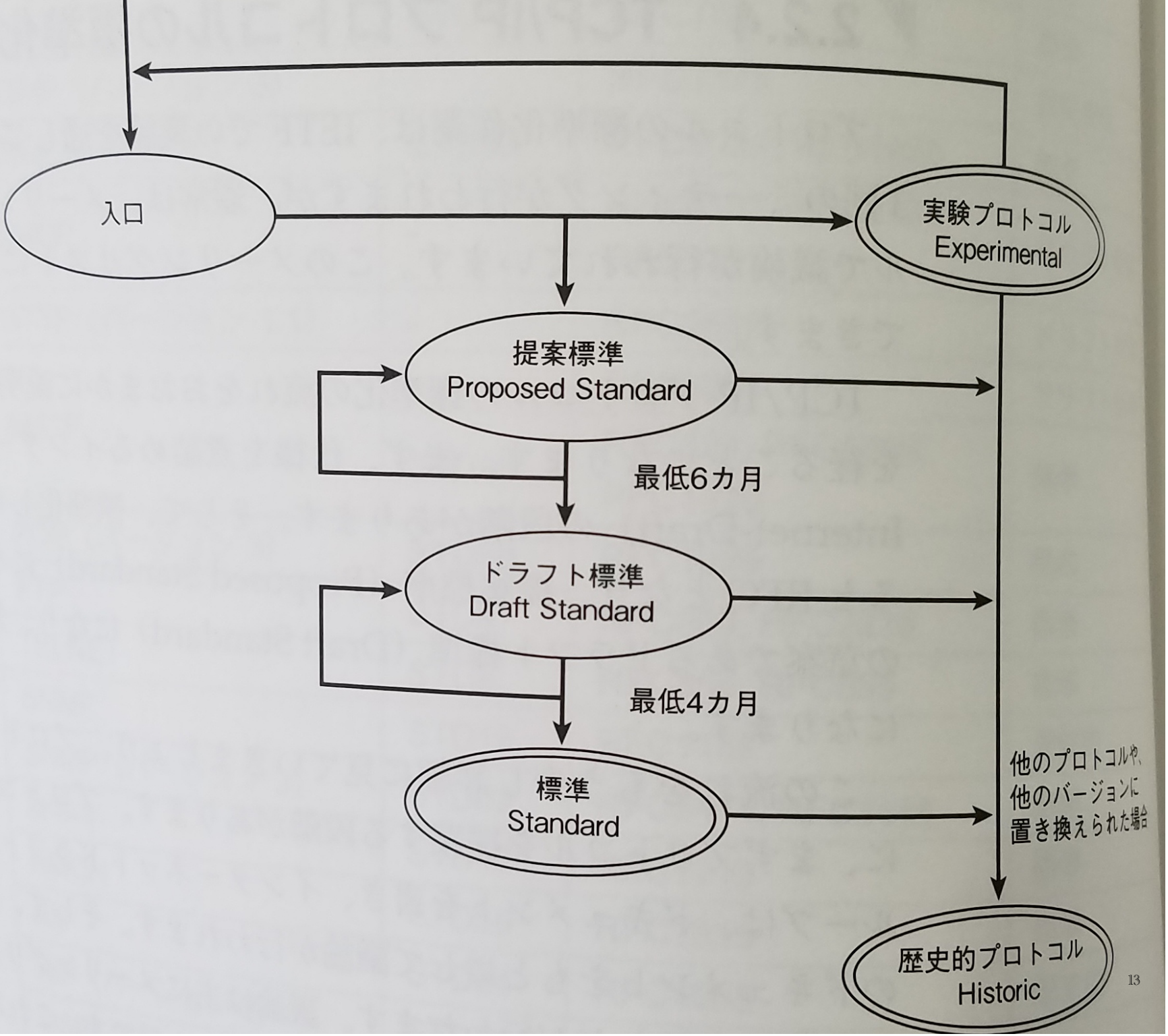
- Openess
  - decided through discusssion in IETF (Internet Engineering Task Force)
  - anyone can participate via mailing list
- Focus on interoperability
  - Almost as if program developed first before specification
  - Specified with implementation in mind
  - Results in highly usable protocol
  - OSI specification wasn't as flexible

2.2.3 TCP/IP Specification in RFC

- Protocols considered for standardization are published as RFC (Request for Comments)
- Also includes FYIs and Experimental
- Ex. RFC791 determines IP, RFC793 determines TCP
- Numbers given in specific order, content unchangeable afterwards
    - Extentions to protocols made in new RFC
- STD (Standard) #s do not change

2.2.4 Process of Standardizing TCP/IP

- Discussions in 3 meetings every year or mailing list
- Stages of proposal
  - I-D (Internet-Draft)
  - Proposed Standard
  - Draft Standard
  - Standard
- I-D valid for 6 months
- IESG (IETF Engineering Steering Group) approves publishing as RFC
- Implement in numerous devices for each stage
- In TCP/IP, standards are already widely adopted


入口
実験プロトコル
Experimental
提案標準
Proposed Standard
最低6カ月
ドラフト標準
Draft Standard
最低4カ月
標準
Standard
他のプロトコルや、
他のバージョンに
置き換えられた場合
歴史的プロトコル
Historic

2.2.5 Obtaining RFC

- All RFC managed under "RFC Editor"
  http://rfc-editor.org/rfc
  ftp://rfc-editor.org/in-notes

2.3 Basics of the Internet

2.3.1 What is the Internet?

- originally meant mutliple networks connected into one
    - could be internal networks among groups/divisions
    - now called Internet Working
- Now, means network spawned from ARPANET

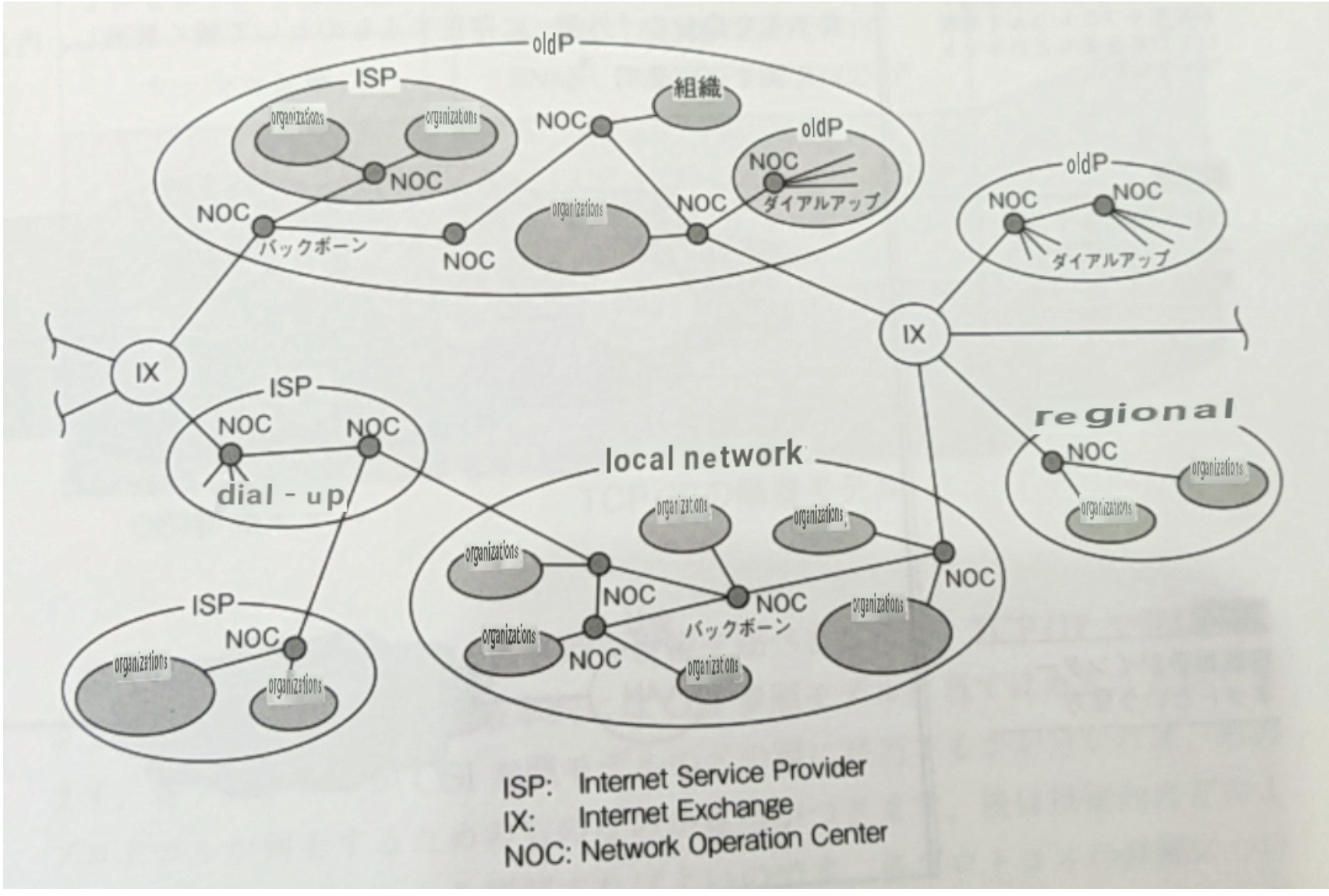
2.3.2 Internet and TCP/IP

- Need protocol to communicate on Internet
- TCP/IP developed originally for Internet

2.3.3 Structure of the Internet

- Internet consists of smaller levels of networks
- Each network connected by Backbone basic network and Stub terminal networks
- Networks connected by NOC (Network Operation Center)
- Networks of differing policies connected by IX (Internet Exchange)

An example of Internet network

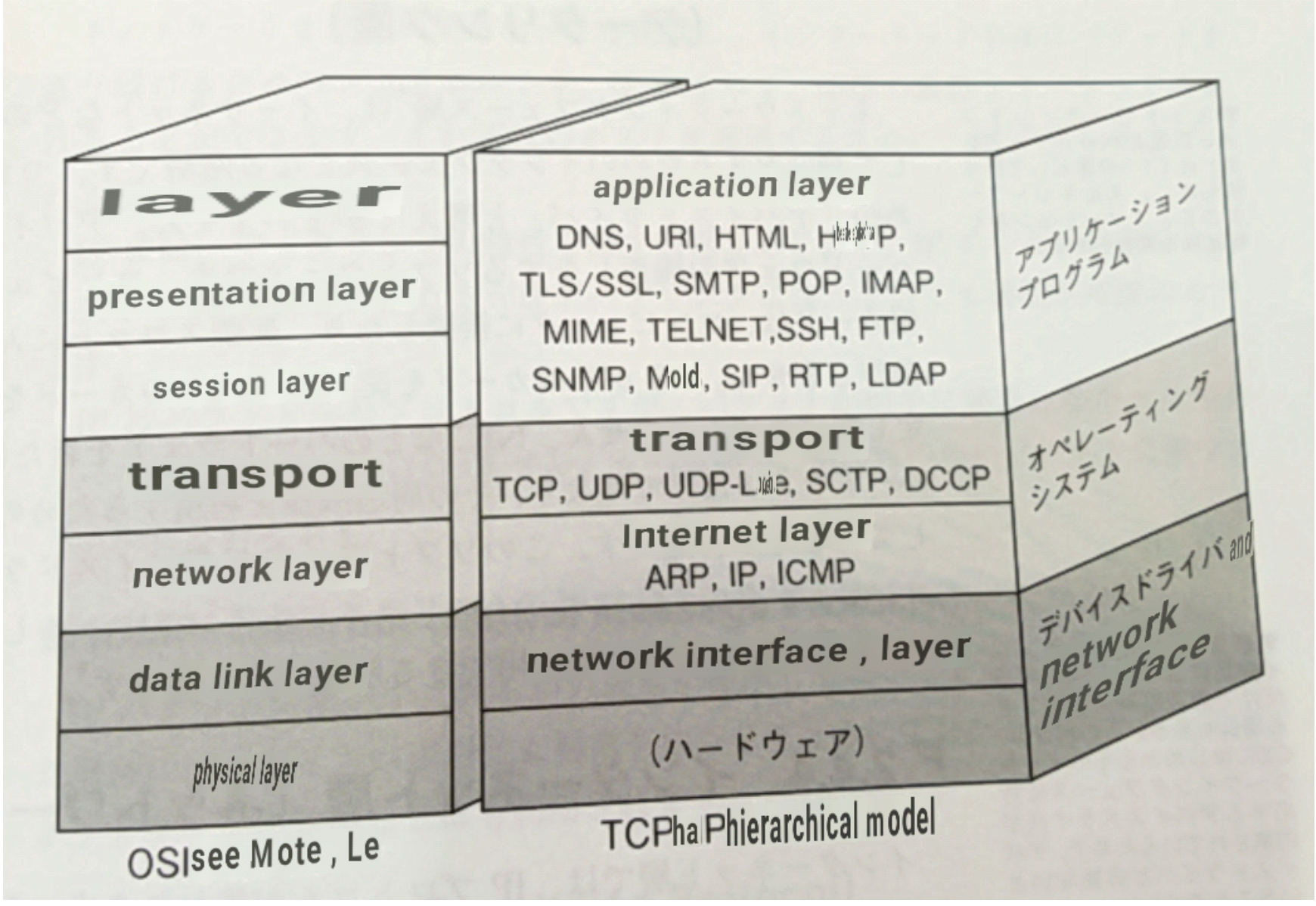


2.3.4 ISP and Local Network

- ISPs provide menu of services
- Local Networks are run by local governing bodies and volunteers
    - cheaper, but with more constraints

2.4 TCP/IP Protocol Layers

2.4.1 TCP/IP and OSI Models



2.4.2 Hardware

- Points to Ethernet, phone lines, etc.
- Can be whatever mechanism

2.4.3 Network Interface

- Think of it as "Network Driver"
- Indicates how OS handles network hardware

2.4.4 Internet/Network

- Handles sending of packets from A to B
- IP (Internet Protocol)
    - IP Addresses identify each device
    - Hides datalink features to communicate regardless
- ICMP (Internet Control Message Protocol)
    - Alerts and diagnozes disruptions in networks
- ARP (Address Resolution Protocol)
    - Obtains MAC Address from IP Address

2.4.5 Transport

- Enables communication within application programs
- TCP (Transmission Control Protocol)
    - Connection-full and Trusty transport protocol
    - Ensures data arrival on both ends
    - Resolves data loss and missequencing
    - Complicated
        - Takes 7 runs to establish connection
        - Not suited for rich media
- UDP (User Datagram Protocol)
    - Connection-less and Trustless transport protocol
    - Doesn't check arrival at other end (Application checks)
    - Suited for rich media

2.4.6 Application

- Implemented in Application Program
- Matches Session and Presentation Layer in OSI
- Most services made in Server-Client model

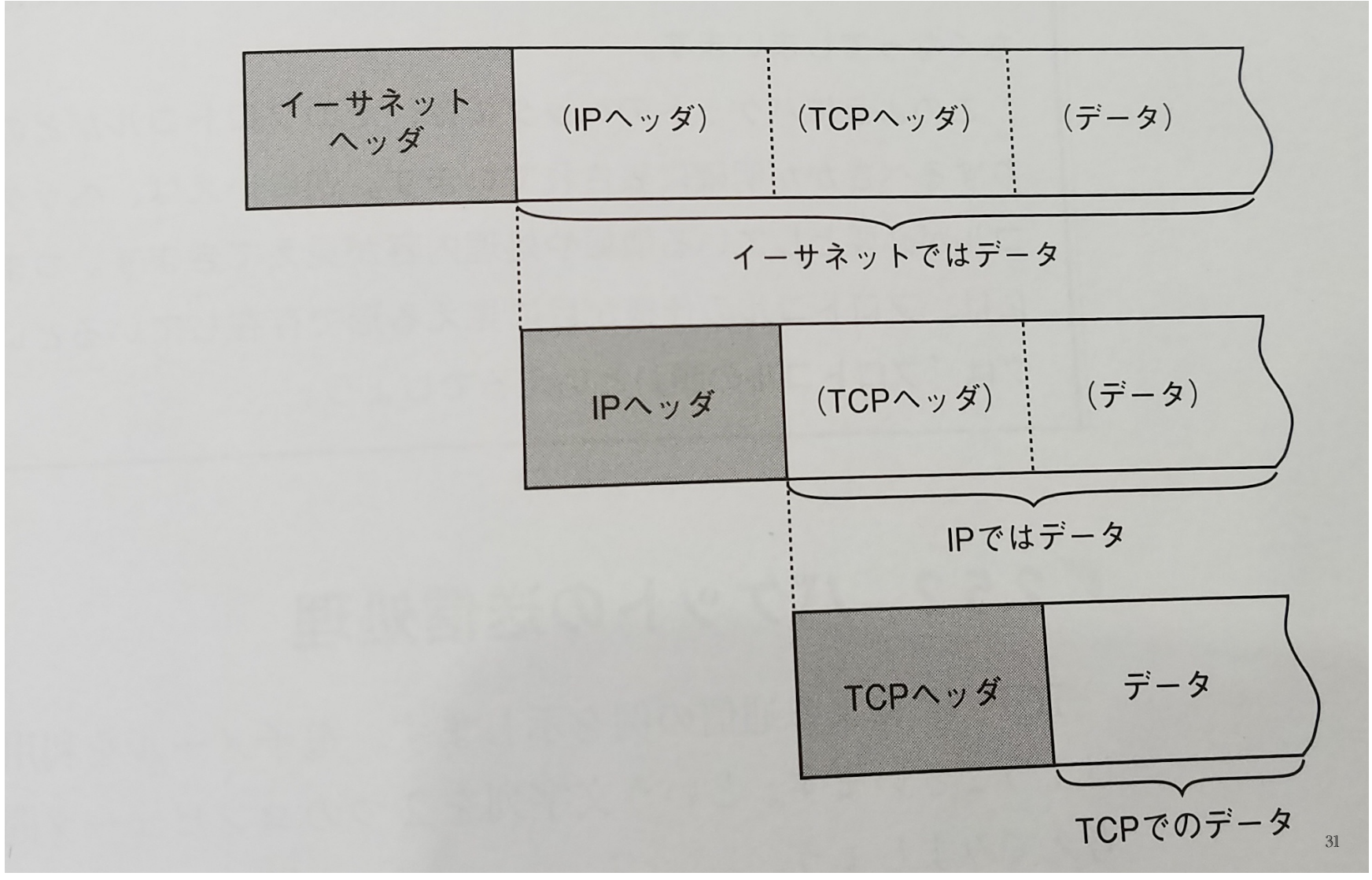
- WWW (World Wide Web)
  - Killer app to spread Internet
  - Mostly in HTTP
- Email
  - Uses SMTP (Simple Mail Transfer Protocol)
  - Initially limited to text, but data type extended by MIME (Multipurpose Internet Mail Extention)

- FTP (File Transfer Protocol)
    - Send file to another computer
    - Send in Binary or Text mode
    - Establish 2 TCP Connections: Control and Data
- Remote Login (TELNET & SSH)
- Network Admin
    - SMNP (Simple Network Management Protocol)
    - Manged devices are called agents
    - MIB (Management Information Base)
        - file structure with device info

2.5 TCP/IP Model and Communication

2.5.1 Packet Header

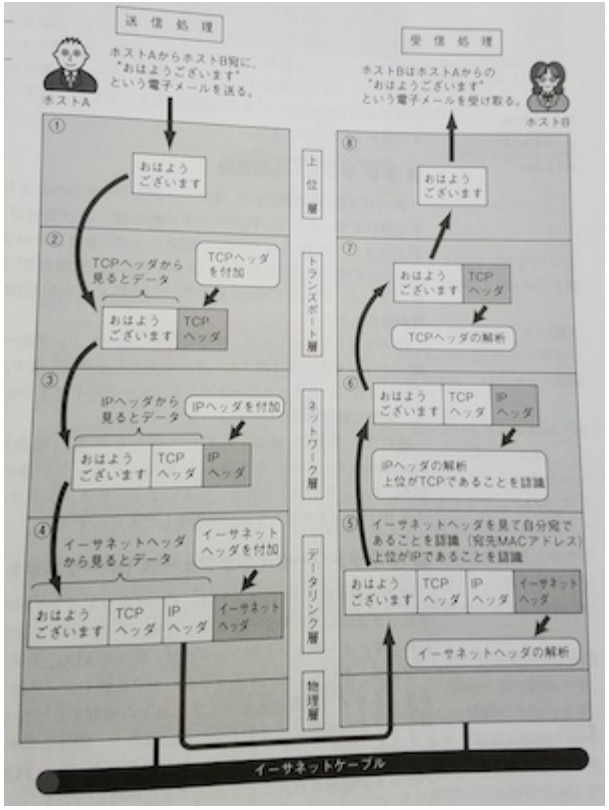
- Add header to data on each layer
- Indicates Addresses and Metadata



2.5.2 Sending Packets

Ex. How email is sent

1. Application Process
    - Create message "Hello"
    - Application program handles text code etc.

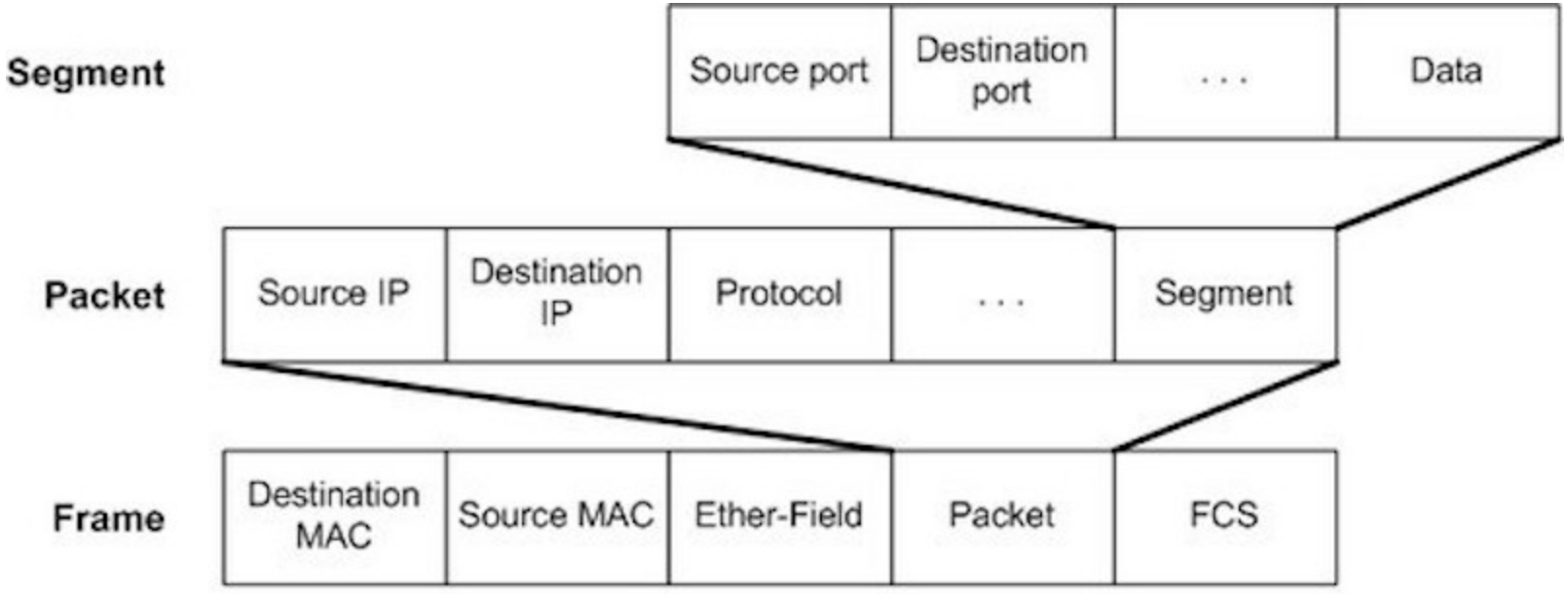

送信処理
ホストAからホストB宛に、
"おはようございます"
という電子メールを送る。
ホストA
受信処理
ホストBはホストAからの
"おはようございます"
という電子メールを受け取る。
ホストB
①
おはよう
ございます
②
TCPヘッダから
見るとデータ
TCPヘッダ
を付加
おはよう
ございます
TCP
ヘッダ
③
IPヘッダから
見るとデータ
IPヘッダを付加
おはよう
ございます
TCP
ヘッダ
IP
ヘッダ
④
イーサネットヘッダ
から見るとデータ
イーサネット
ヘッダを付加
おはよう
ございます
TCP
ヘッダ
IP
ヘッダ
イーサネット
ヘッダ
上位層
トランスポート層
ネットワーク層
データリンク層
物理層
⑧
おはよう
ございます
⑦
おはよう
ございます
TCP
ヘッダ
TCPヘッダの解析
⑥
おはよう
ございます
TCP
ヘッダ
IP
ヘッダ
IPヘッダの解析
上位がTCPであることを認識
⑤
イーサネットヘッダを見て自分宛で
あることを認識（宛先MACアドレス）
上位がIPであることを認識
おはよう
ございます
TCP
ヘッダ
IP
ヘッダ
イーサネット
ヘッダ
イーサネットヘッダの解析
イーサネットケーブル

2. TCP Module Process
    - Handles connections
    - Adds TCP header
        - Port number for application on sender and reciever
        - Sequence Number
        - Checksum

3. IP Module Process
    - Handles previous data as single piece
    - Adds IP header
        - Sender and Reciever IP address
        - Identification for data as TCP or UDP
    - Decides device to send IP packet via routing table
    - Sends data through network interface driver
    - Uses ARP to find MAC Address

4. Network Interface Process
    - Adds Ethernet Header
        - Sender and Reciever MAC Address
        - Ethernet Type
    - Appends FCS (Frame Check Sequence)

2.5.3 Packets through the Datalink



2.5.4 Recieving Packets

5. Network Interface Process
    - Check if packet is sent to itself
    - Look up Ethernet Type Field
    - Type of data handled by Ethernet Protocol
6. IP Module Process
    - Accepts if IP Address is itself
    - Looks up type of Transport (TCP/UDP)

7. TCP Module Process
    - Calculate checksum
    - Check sequence of data
    - Responds with confirmation
    - Hands data to specified port number
8. Application Process
    - Accepts data for email

Thank you 👍

r1cky [kumo]