【Full Understanding】

TCP Algorithm

Basic edition
《Loss base method》

- Mechanism for reliable data delivery
- Efficient transmission by grasping network congestion

Overall table of contents

1． TCP overview

1-1. What are RFCs?

1-2. What is TCP/IP?

1-3. Protocol stack

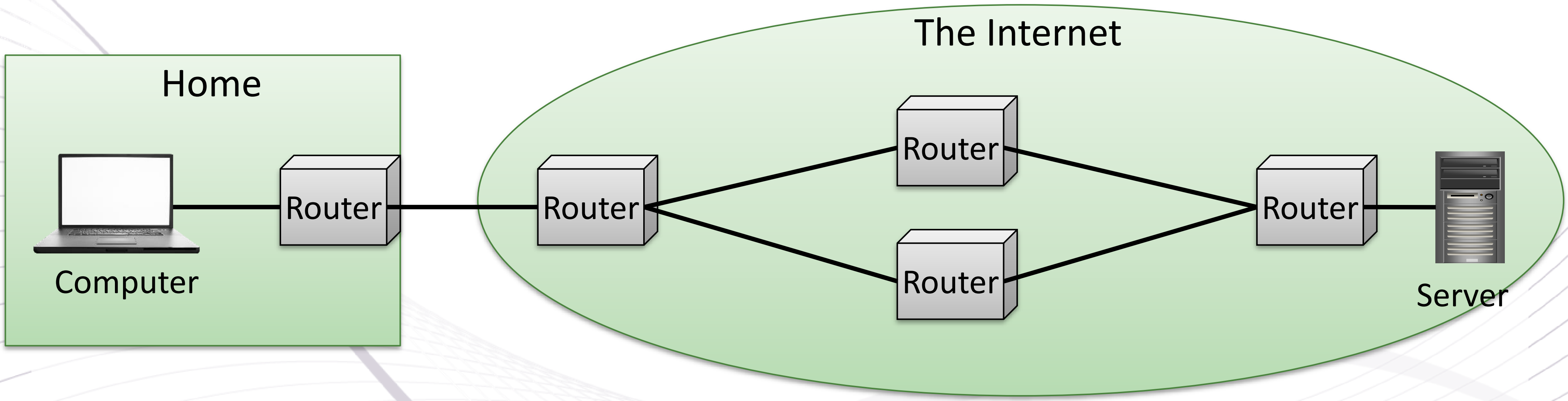
1-4. Packet data (capture)

1-5. TCP version

1-6. TCP data format

1-1.　What are RFCs?

All devices involved in Internet communication must handle data using pre-defined technology.
Therefore, standardization (arrangement) of Internet technology is being carried out. This standardization is called RFC.



- **Organization：IETF（Internet Engineering Task Force)**
- **Standardized provisions：RFC（Request for Comments)**
  1) IP　：　RFC791
  2) TCP：　RFC9293　Released in August 2022
  (Obsoletes: 793, 879, 2873, 6093, 6429, 6528, 6691)

1-2.　What is TCP/IP?

- **TCP/IP**　**Regulations for communication on the Internet.**

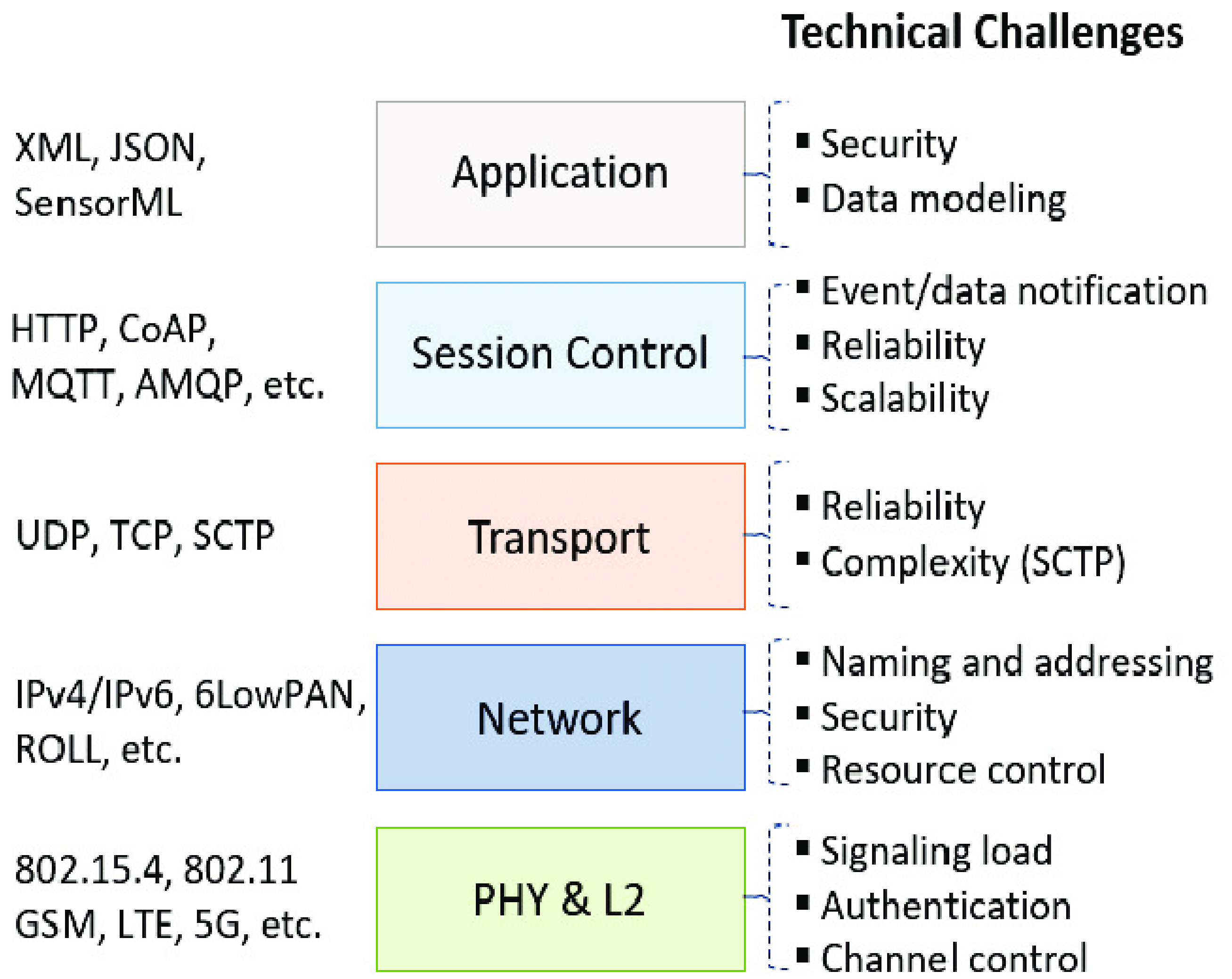
  ①IP　：The role of representing addresses on the Internet.

  ②TCP：The role of delivering data efficiently and reliably according to the state of the communication path.

- **UDP/IP**

  ③UDP：It does not perform communication control just to deliver.

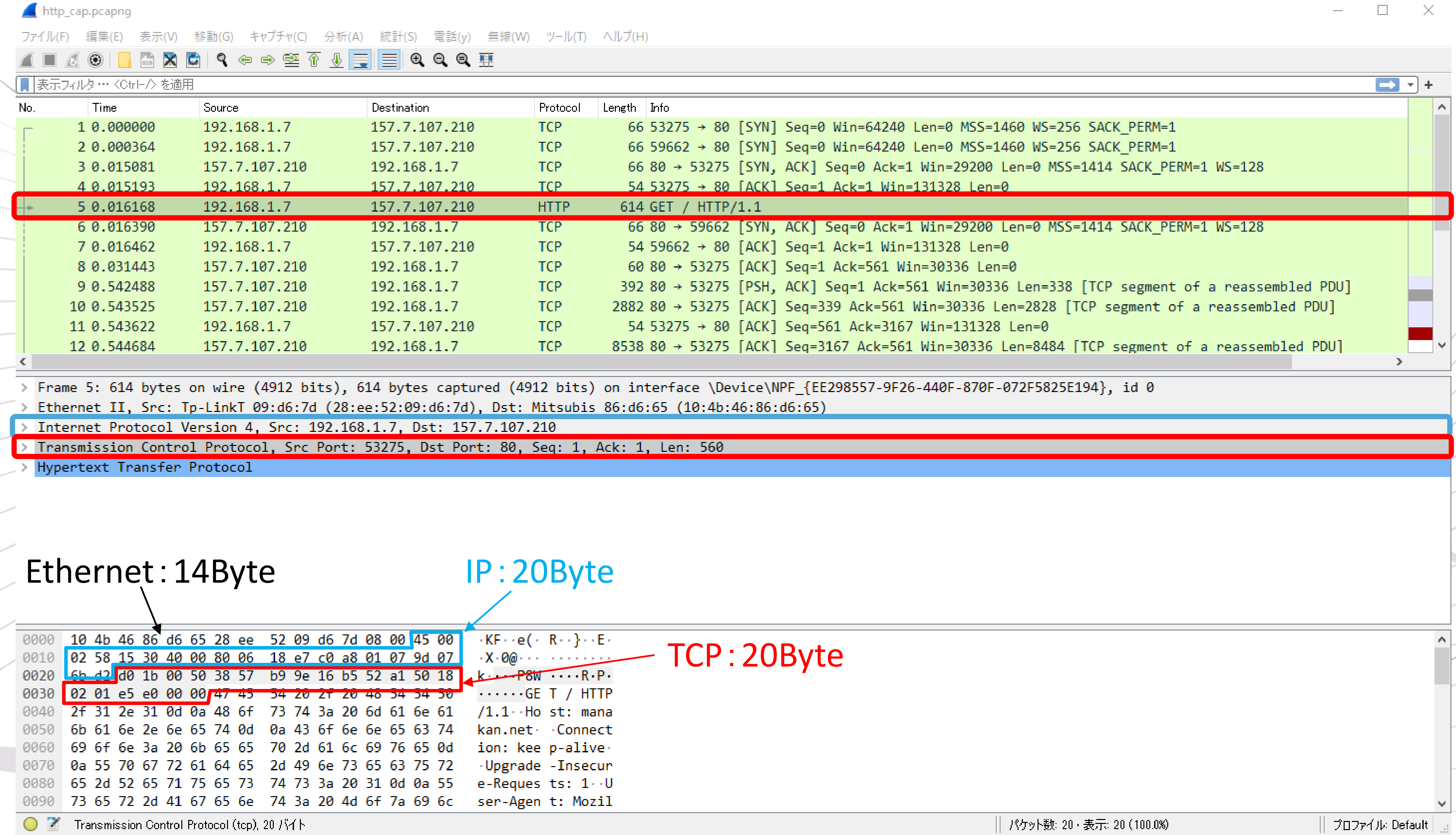
1-3. Protocol stack



https://www.researchgate.net/figure/Protocol-Stack-and-Technical-Challenges_fig1_320453832

- **Match the rules used by each layer at the transmitting and receiving terminals**
- **There is no dependency between layers. Determine the rules to be used for each layer**

1-4.　Packet capture（WireShark）



1-5. TCP Versions

- **Initial standard, most basic TCP**
  -「Reno」「NewReno」*1

Contents of this time

- **For high delay/wideband (long fat pipe)**
  -「BIC」「CUBIC」「H-TCP」*1、「Fast TCP」*2、「Illinois」*3

- **For data centers with low latency and wide bandwidth**
  -「DCTCP」*3

- **For wireless environments such as mobile lines and Wi-Fi**
  -「Veno」「Westwood」*2

《Reference 1: Method》
***1: Loss base method**
***2: Delay-based method**
***3: Hybrid method of the above two**

《Reference 2: Implementation of each OS》
Linux、Android：CUBIC
OS X：NewReno
Windows：CTCP/DCTCP

1-6. TCP Data Format

- **Defined by RFC 9293**

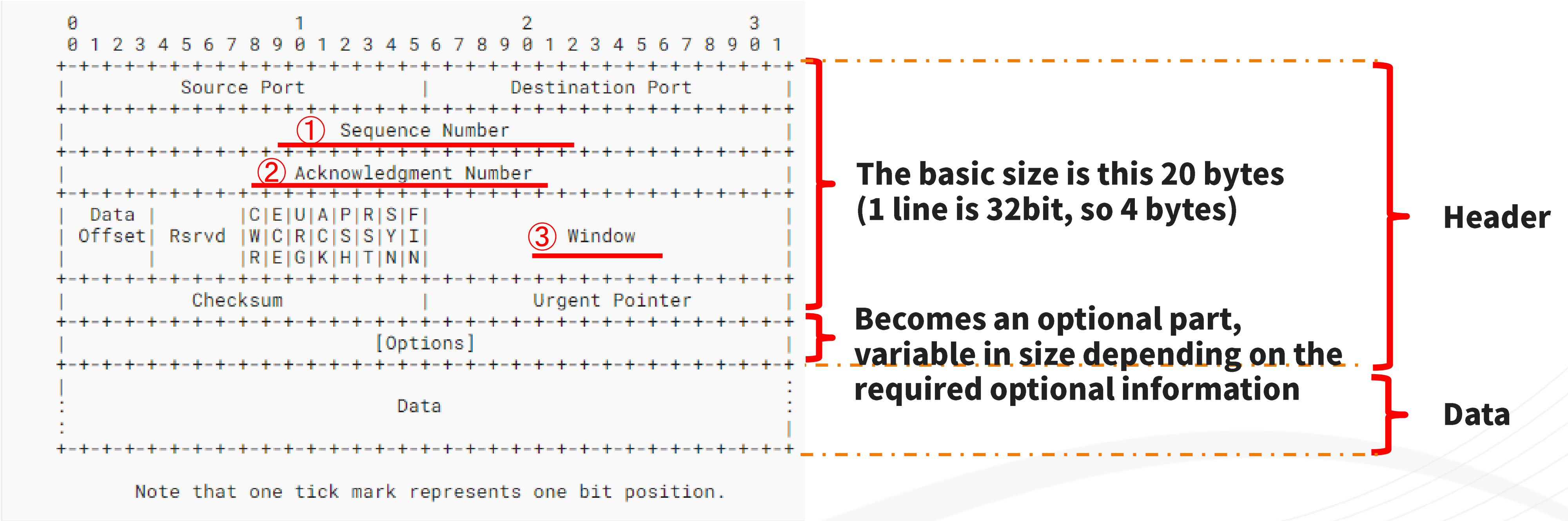


Figure 1: TCP Header Format

①	Sequence Number	The sending terminal attaches a number (Sequence Number) to the sending data and sends it.
②	Acknowledgment Number	Based on the "Sequence Number" received by the receiving terminal, the "Sequence Number" expected to be received next is notified to the sender as an acknowledgment (Acknowledgment Number).
③	Window	Notifies the sender of the amount of data that the receiving terminal can receive (receive buffer size).

2. TCP Algorithm

2-1. Establishing a TCP connection

2-2. Arrival confirmation

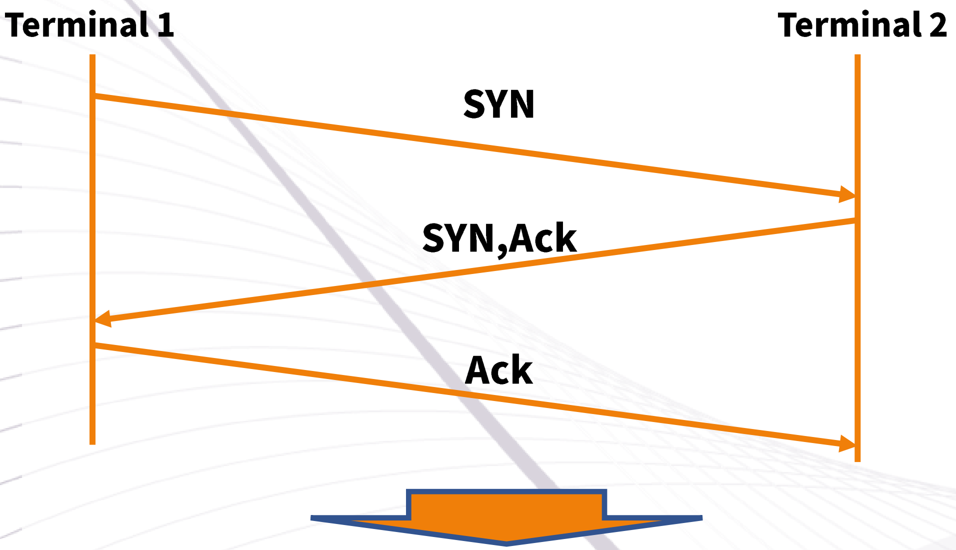
2-3. Window control

2-4. Determining the window size

2-5. Retransmission control

2-1. Establishing a TCP connection

3-Way Handshake

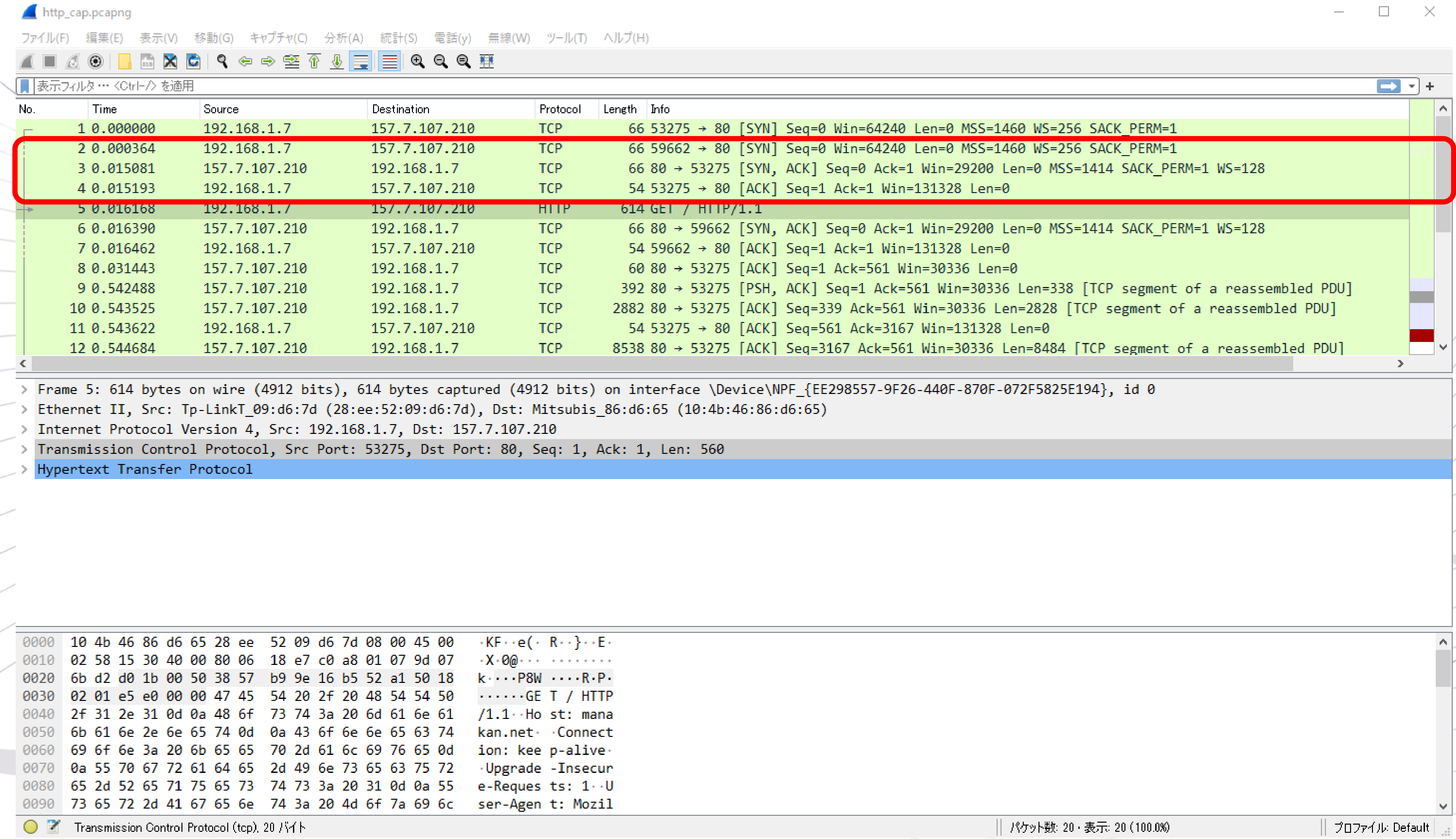


Data is sent from here onwards

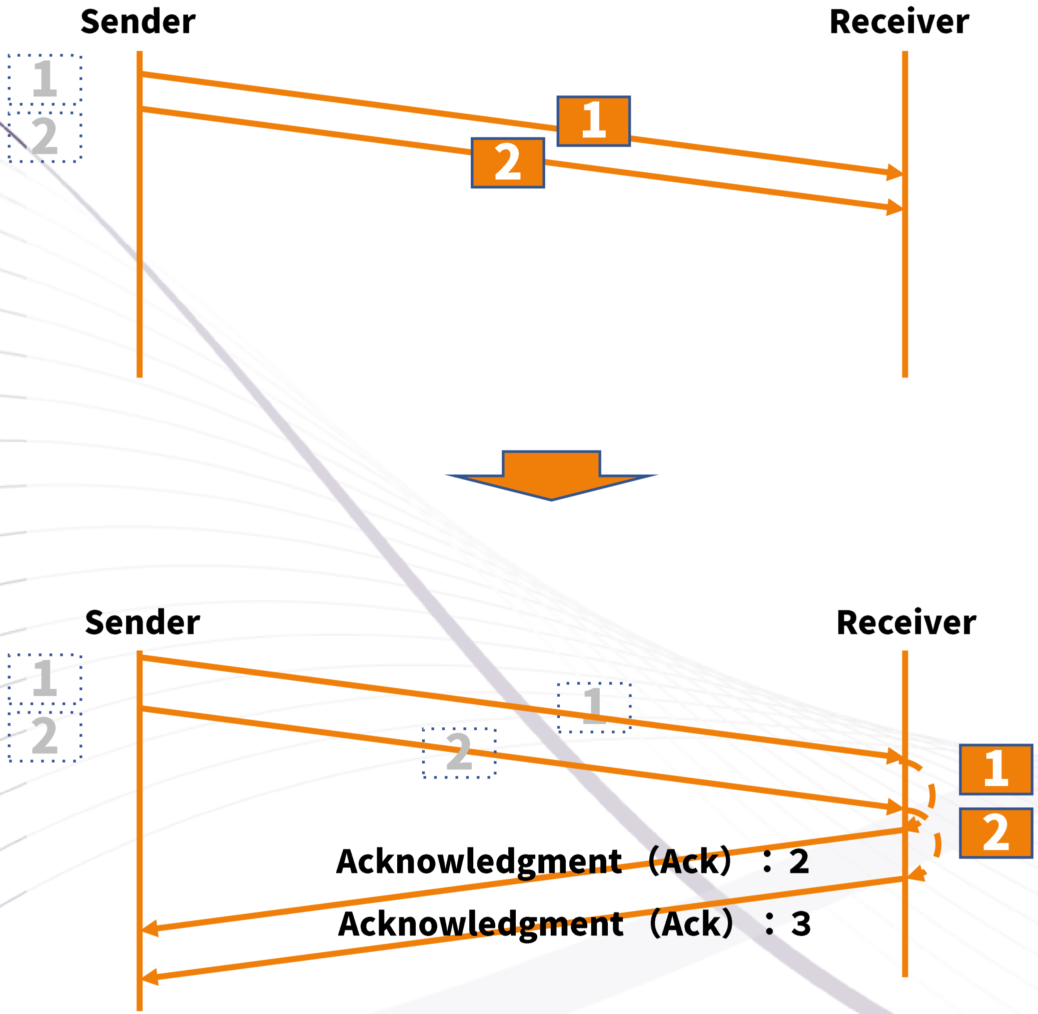
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                        TCP Header Format
```

The availability of optional features such as SelectiveAck and window size options are also determined at this establishment.

Packet capture（WireShark）



2-2. Arrival confirmation



Send data number

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                      TCP Header Format
```

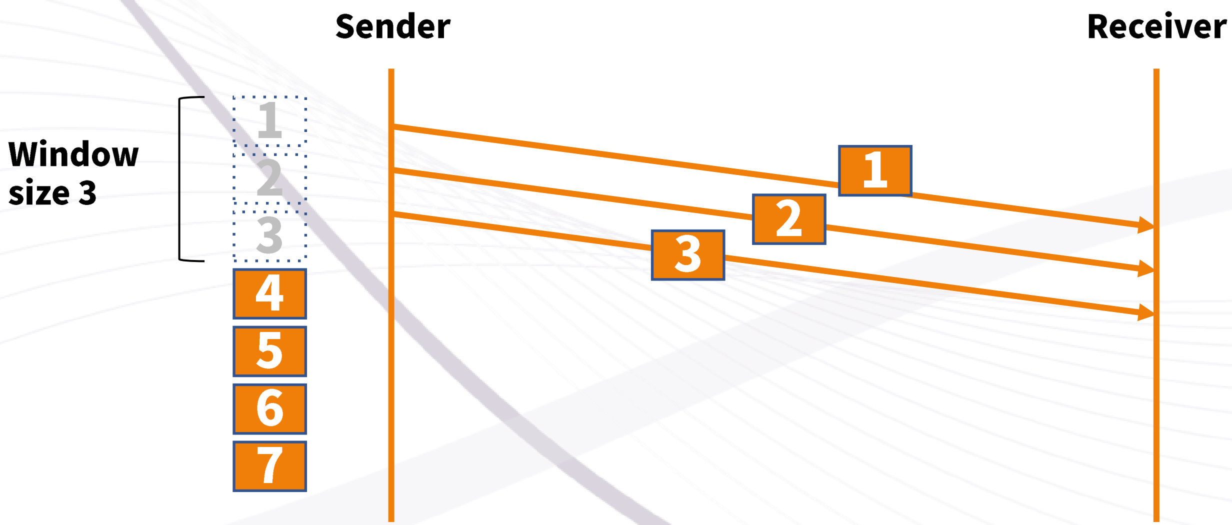
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                      TCP Header Format
```

Arrival confirmation number
(Next number to receive)

2-3. Window Control

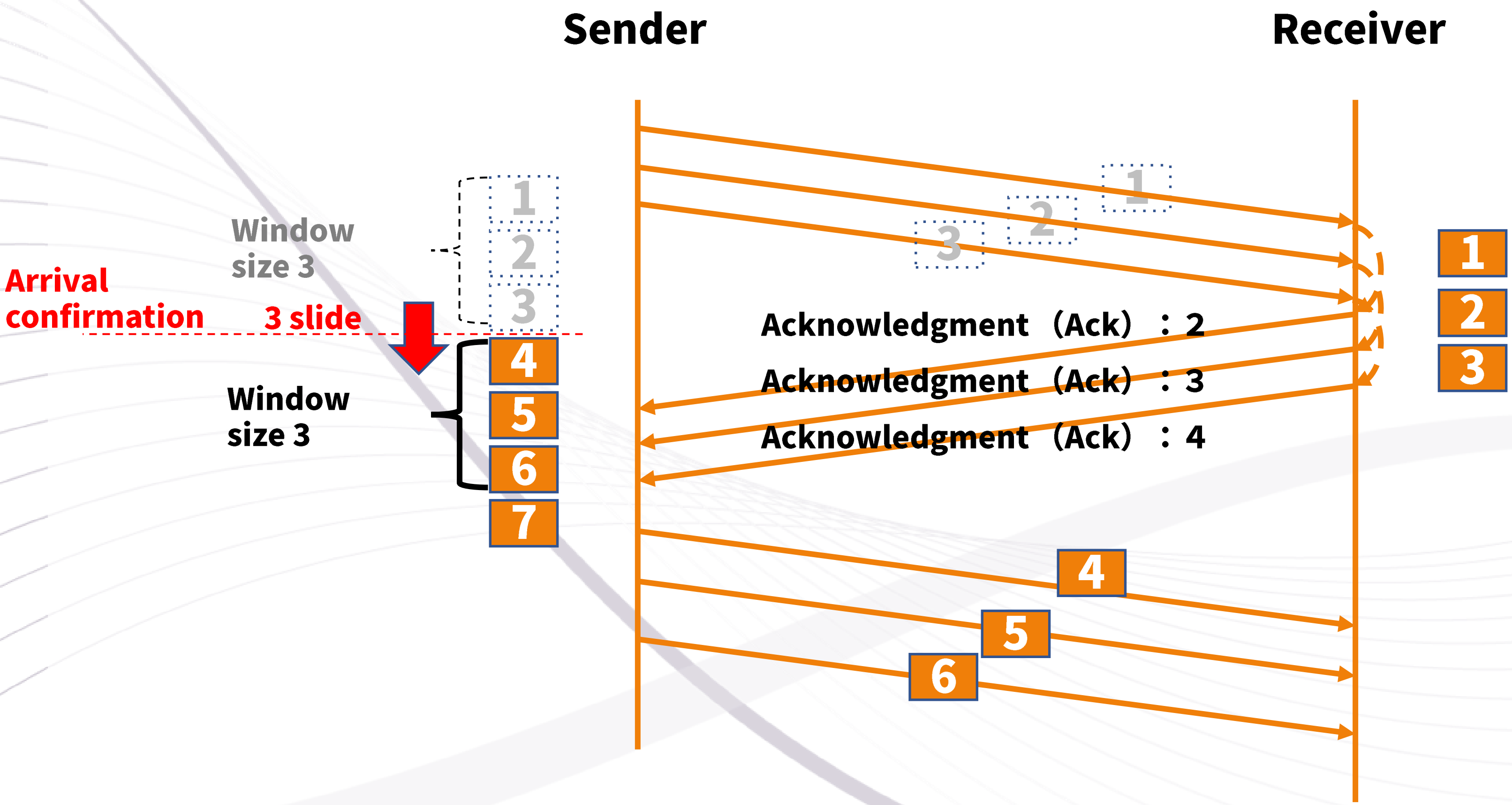
What is a "window"

Amount of data (bytes) that can be sent even though arrival has not been confirmed.



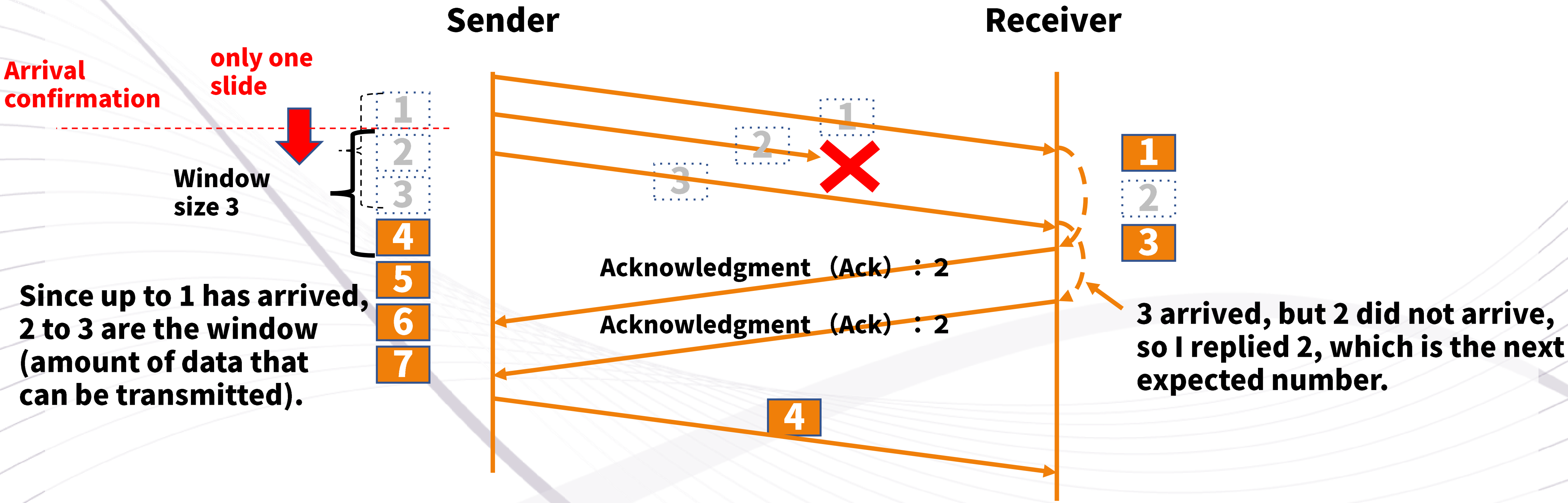
2-3. Window Control

[When everything arrives normally]



2-3. Window Control

[When packet 2 is lost]



2-4. Determining window size

The window size is the size that can be sent, so it is determined by the sending terminal. The decision is made by the smaller value of the following two contents.

1) Flow control

Controlled by the buffer size (window size) notified from the receiving terminal.

2) Congestion control

Controlled by the window size calculated by the sending terminal

Nowadays, the performance of personal computers has improved, so the buffer size of the receiving terminal has little effect, and it is safe to assume that the congestion control calculated on the sending side will become the window size.

2-4-1. Flow control

Receiving terminal notifies sending terminal of current buffer size

- **Notify with TCP packet header "Window (16bit)“**
- **65535 bytes (64 Kbytes) that can be displayed with a maximum size of 16 bits**
- **In recent years, it is common to use window size options that can expand the maximum size**

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                    TCP Header Format
```

2-4-2. Congestion control (congestion window)

- **Congestion control is controlled by a congestion window.**
- **Congestion window is a value that is stored and calculated inside the sender terminal**
- **No information appears in the packet header.**

Due to the high performance of personal computers, the congestion window size calculated at the sender terminal for congestion control is more dominant than the buffer size of the receiving terminal for flow control.

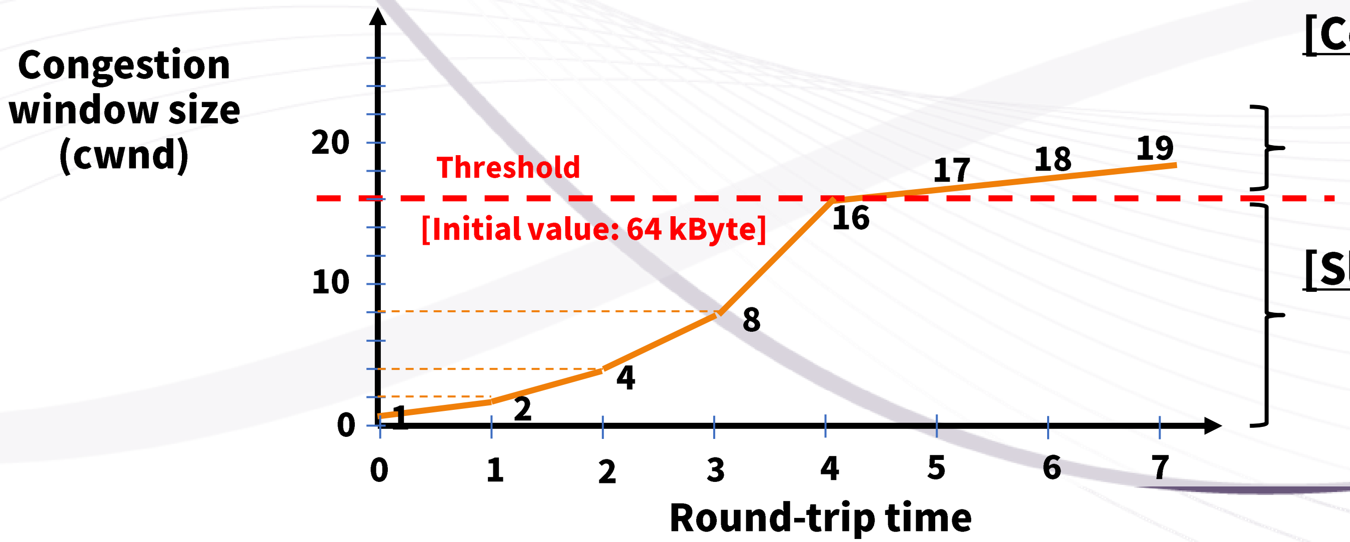
(However, since 64Kbyte of 16Bit is the maximum value without options and is dominant, it is generally expanded by the window size option.)

2-4-2. Congestion control (congestion window)

Congestion window (cwnd): Byte size (description is the number of packets)
The congestion window size is calculated in units of MSS (MaximumSegmentSize).

MSS	=	Ethernet Max Packet Length	−	IP Header Length	−	TCP Header Length
	=	1500 Byte	−	20 Byte	−	20 Byte（No Option）
	=	1460 Byte				

There are two phases, [slow start] and [congestion avoidance], and the method of calculating the congestion window size is different.

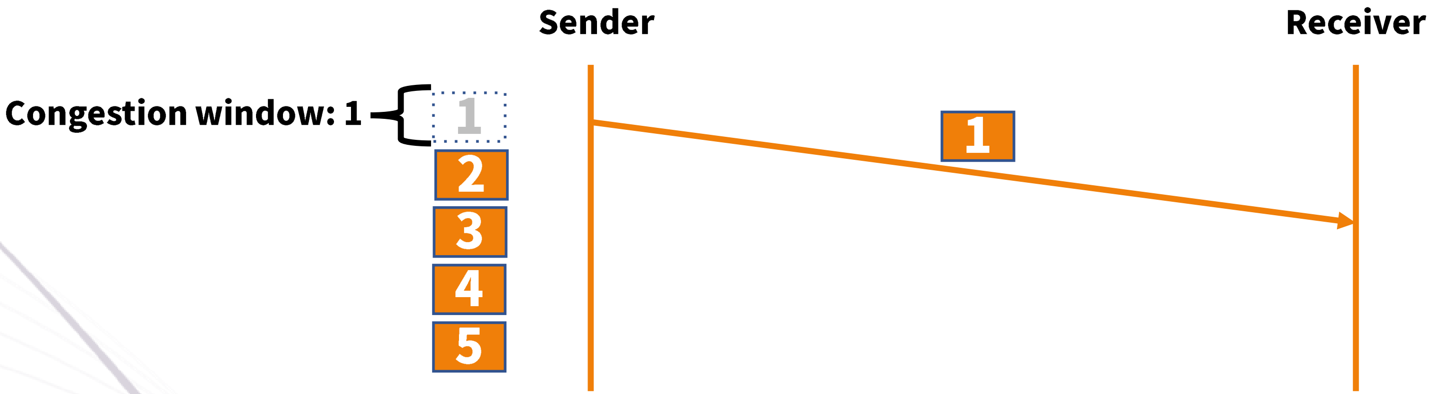


[Congestion avoidance]
1 MSS size (linear) increase per round-trip time
[Calculation]
cwnd = cwnd + MSS/cwnd

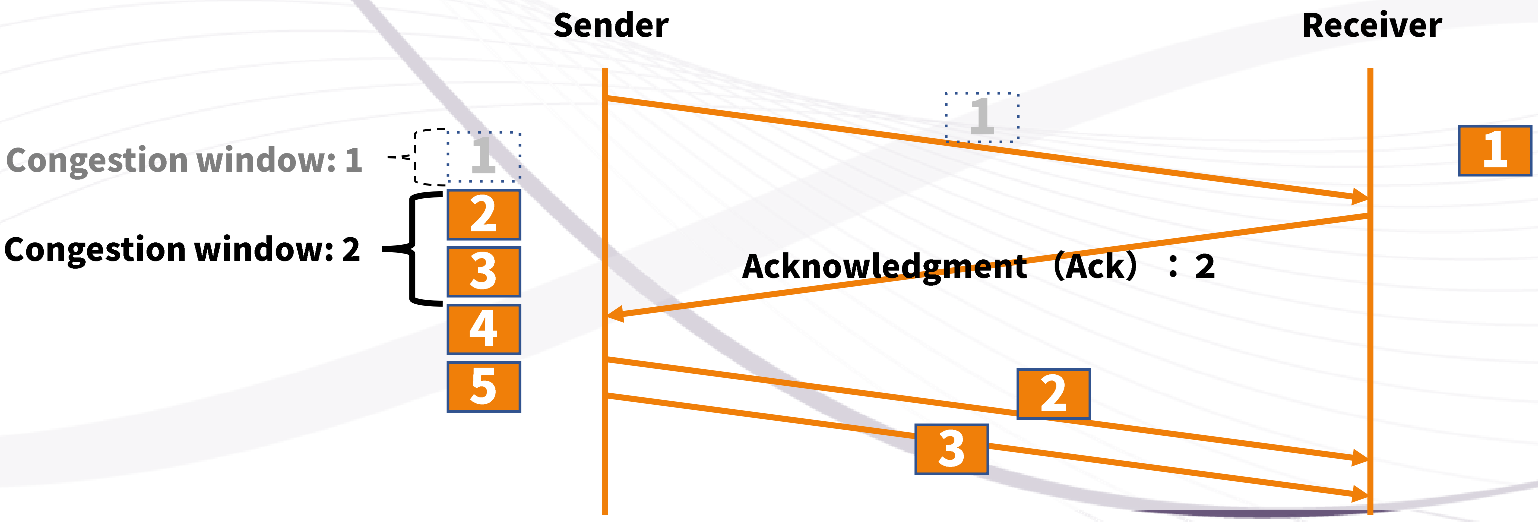
[Slow start]
Exponential increase per round trip time
[Calculation]
cwnd = cwnd + MSS

2-4-2-1. Slow start

[When establishing a connection]



[When acknowledgment 1 is received]



[Calculation]

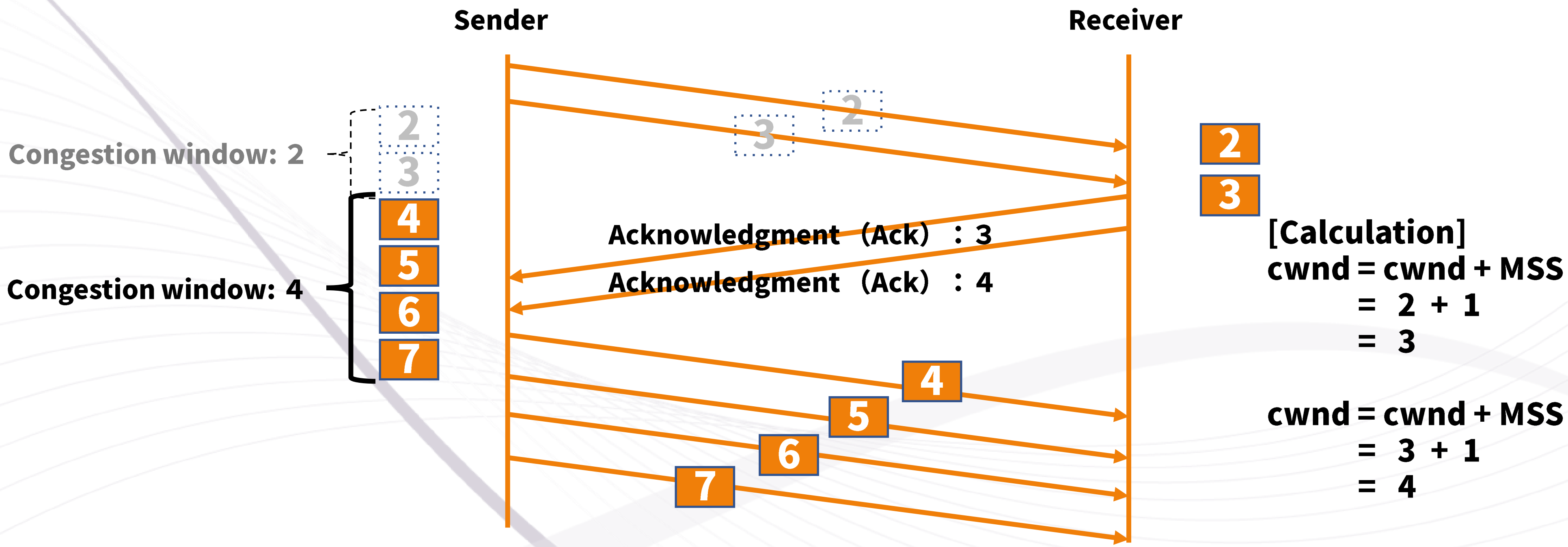
cwnd = cwnd + MSS

= 1 + 1

= 2

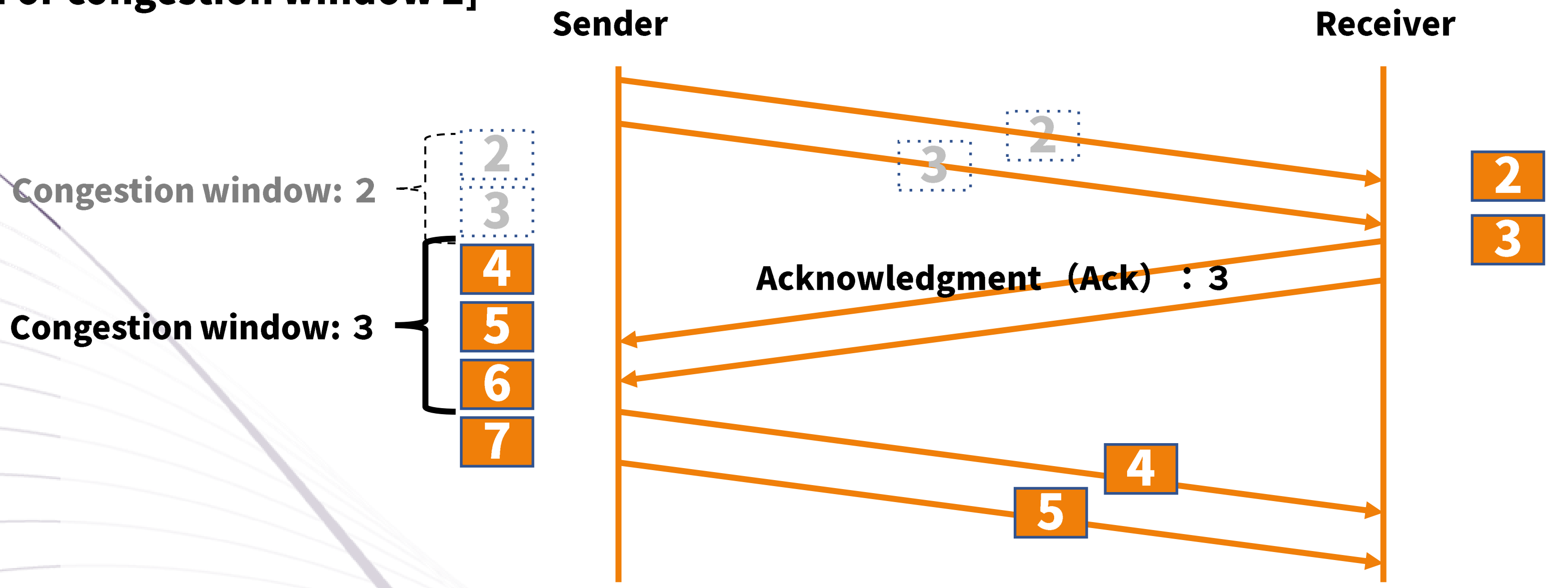
2-4-2-1. Slow start

[When receiving up to acknowledgment 3]



2-4-2-2. Congestion avoidance

[For congestion window 2]

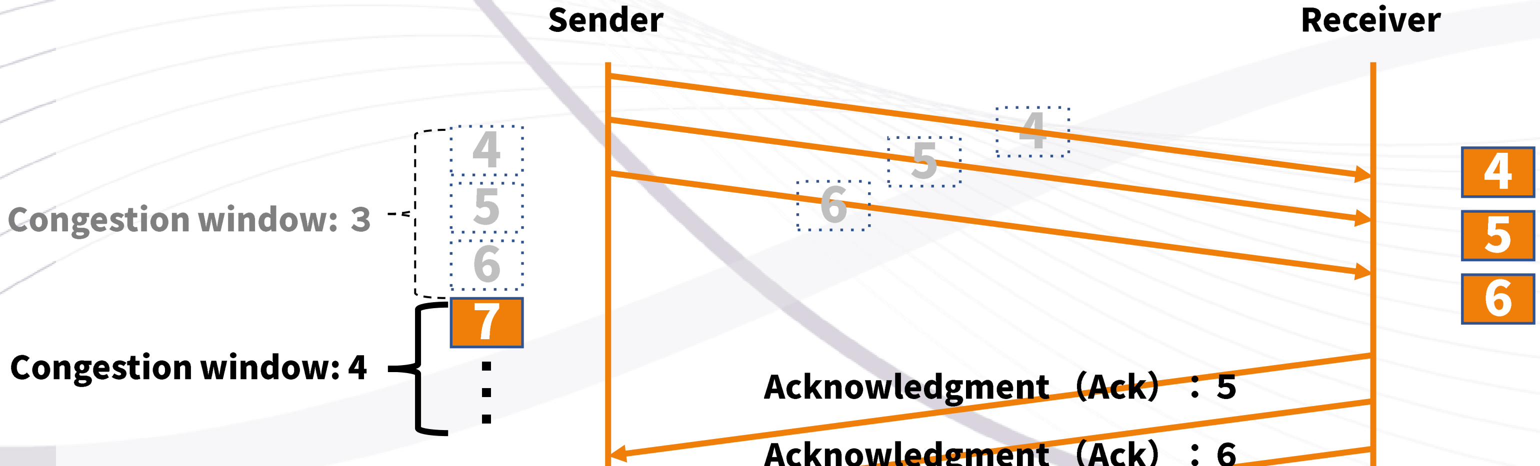


[Calculation]

cwnd = cwnd + MSS/cwnd
= 2 + 1/2
= 2 1/2

cwnd = cwnd + MSS/cwnd
= 2 1/2 + 1/2
= 3

[For congestion window 3]



[Calculation]

cwnd = cwnd + MSS/cwnd
= 3 + 1/3
= 3 1/3

cwnd = cwnd + MSS/cwnd
= 3 1/3 + 1/3
= 3 2/3

cwnd = cwnd + MSS/cwnd
= 3 2/3 + 1/3
= 4

If 3 acknowledgments, which is the congestion window size, are received, the congestion window becomes 4.

2-5. Retransmission control

The Internet is built on the assumption that packets are lost within the network.

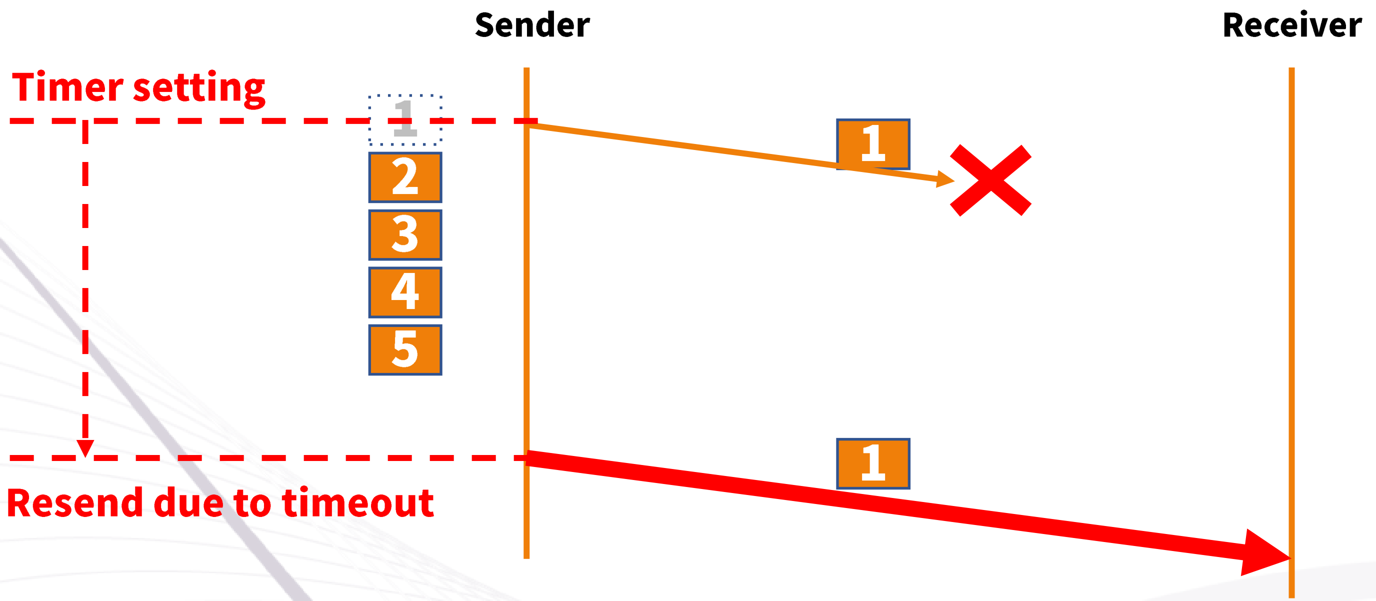
Therefore, TCP has a retransmission control function to ensure data delivery.

Regarding the resend function, the following two functions are implemented.

1) Retransmission due to timeout

2) Retransmission due to duplicate Ack

2-5-1. Retransmission due to timeout

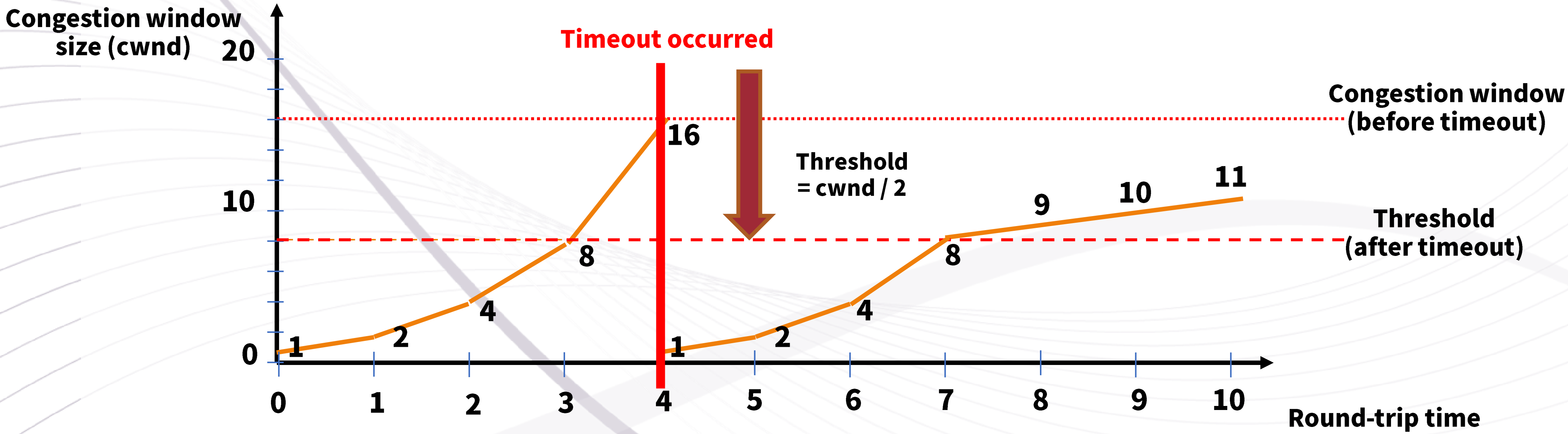


The initial value of the timer time is fixed, but it is updated from time to time based on the actual packet response time.

2-5-1. Retransmission due to timeout

When retransmission occurs due to timeout, the following window control is performed, and the fluctuation of the congestion window size is shown below.

(1) Congestion window (cwnd) = 1 MSS size
(2) Slow start threshold = cwnd / 2

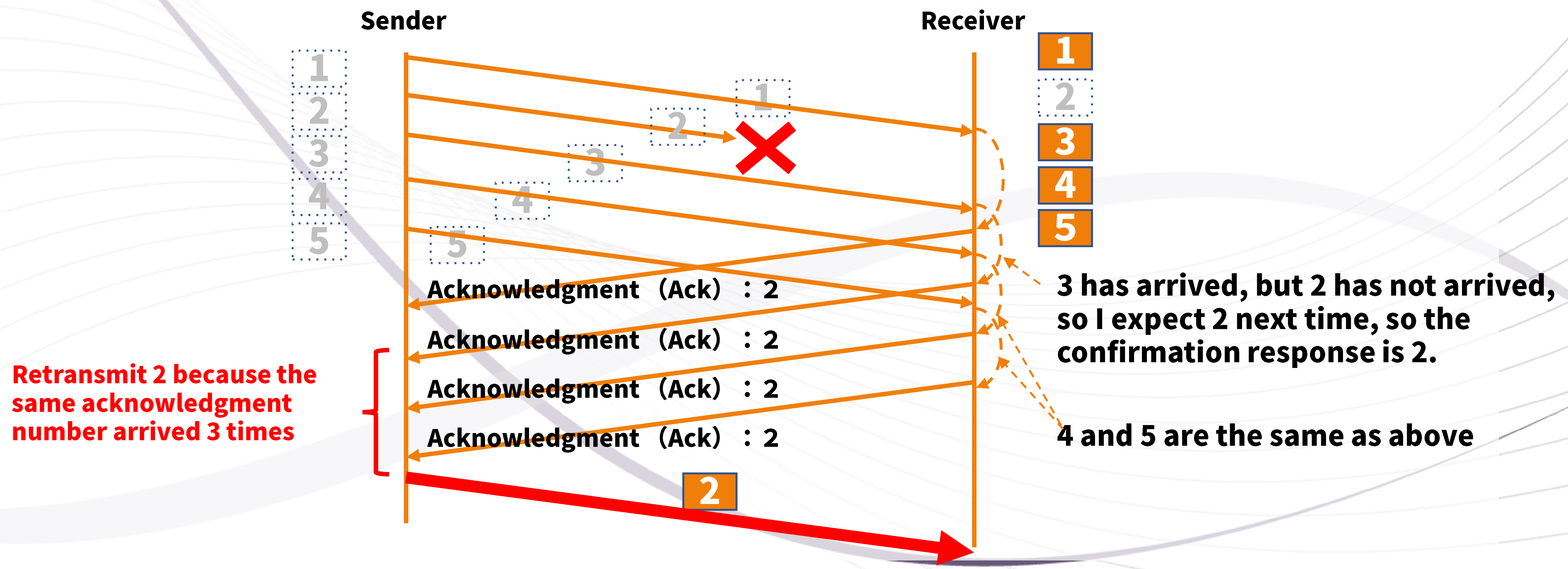


A timeout indicates that the network is very busy. Therefore, it is necessary to carefully transmit and widen the window little by little so as not to cause unnecessary packet loss.

2-5-2. Retransmission due to duplicate Ack

"Fast Recovery Algorithm“
Retransmission by duplicate Ack is retransmitted when the same sequence number as before arrives three times.

[If packet 2 is lost]

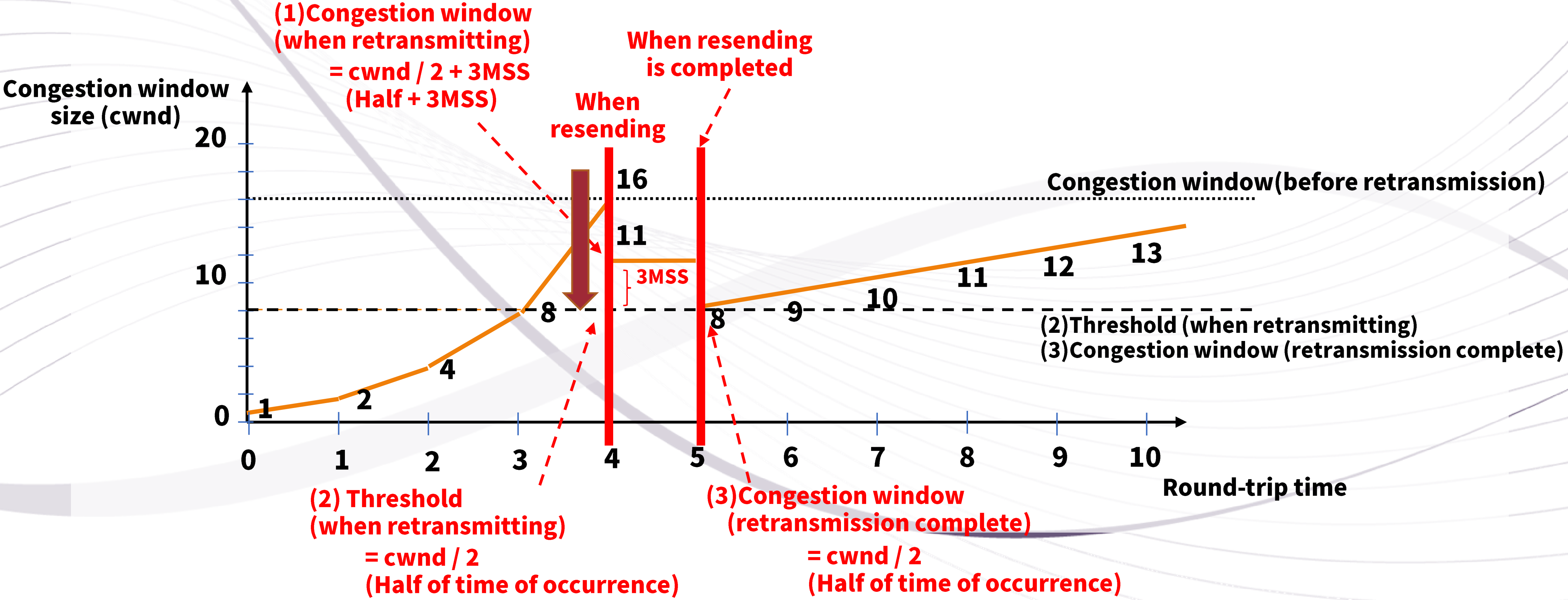


2-5-2. Retransmission due to duplicate Ack

Fast Recovery function: The congestion window is not the first 1 MSS size, but the following calculation is implemented to efficiently transmit data by retransmission.

[When retransmission occurs]
(1) Congestion window (cwnd) = cwnd / 2 + 3 MSS
(2) Slow start threshold = cwnd / 2

[When resend is completed]
(3) congestion window (cwnd) = slow start threshold



3. Summary [TCP Algorithm]

●Arrival confirmation

● Window control
・Flow control
・Congestion control
- Slow start
- Congestion avoidance

●Resend control
・Retransmission due to timeout ← [Large congestion]
・Retransmission by duplicate Ack ← [Less congestion]