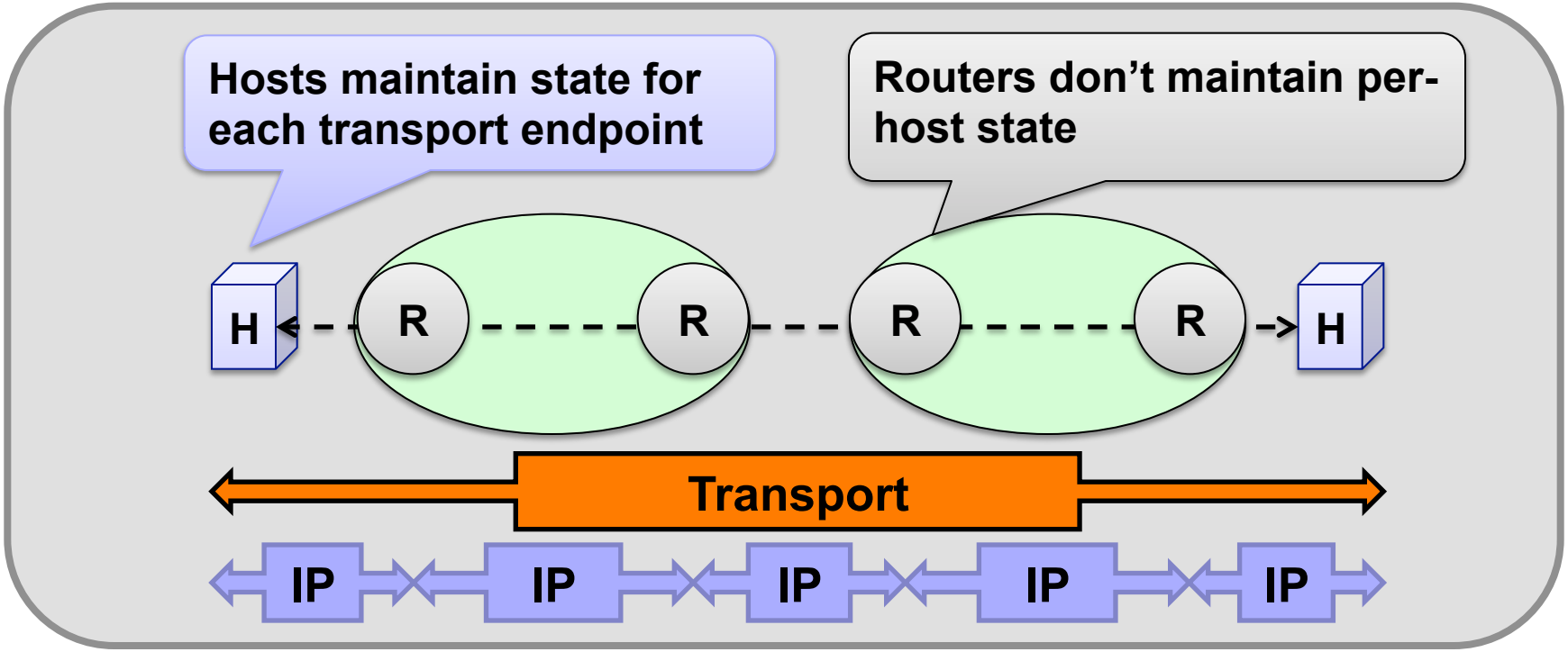
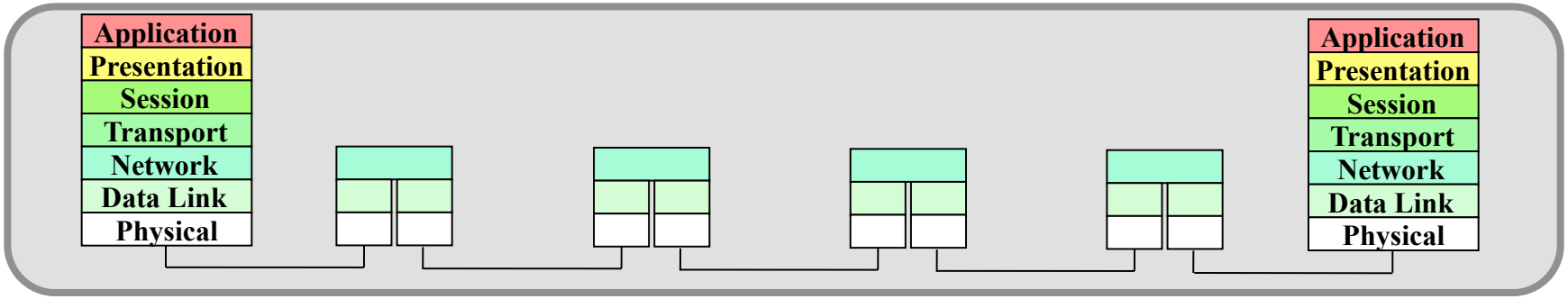
Information Network 1
TCP 1/2

Youki Kadobayashi
NAIST

Transport layer: a birds-eye view





Functions provided by the transport layer

- Communication between processes
  - designation of process
  - identification of inter-process channel
- Interface for upper layer
  - Connection-oriented (virtual circuit)
  - Connectionless (datagram)
- Competition and arbitration of network resource
  - Flow control
  - Congestion control

Next lecture

Transport protocols in the Internet protocol suite

- TCP (RFC793)
  - “Transmission Control Protocol”
  - Connection-oriented
  - Multiple functions for reliablity
- SCTP (RFC4960)
- DCCP (RFC4340)

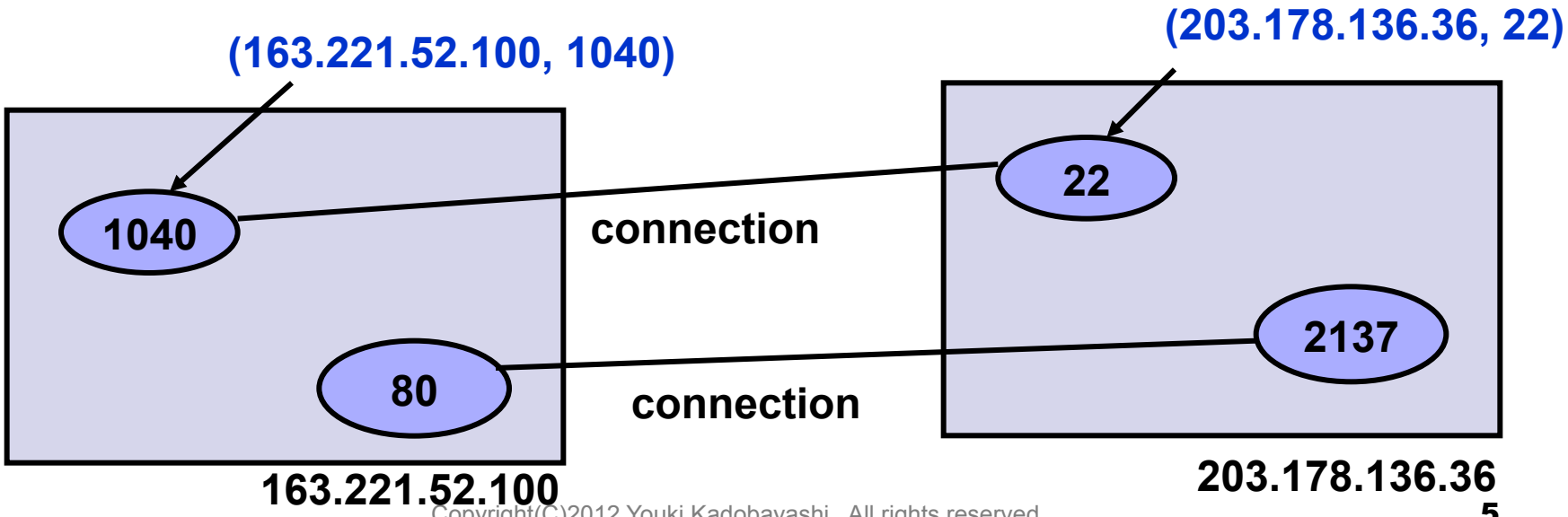
Advanced topic; out of scope

- UDP (RFC768)
  - “User Datagram Protocol”
  - Connectionless
  - IP + Process Identification

For self study

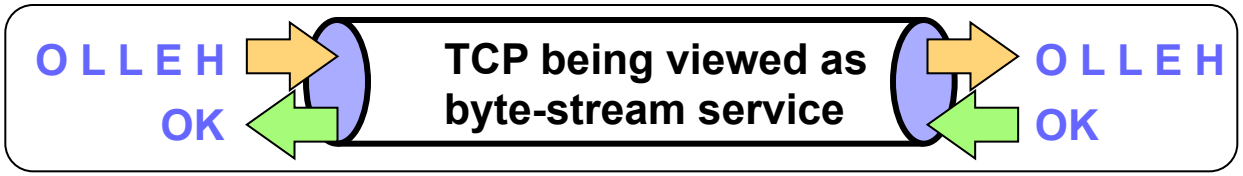
Process and connection identification in TCP

- Designation of Process
  - (IP, port)
- Identification of TCP connection
  - (source IP, source port, destination IP, destination port)



TCP service model (1)

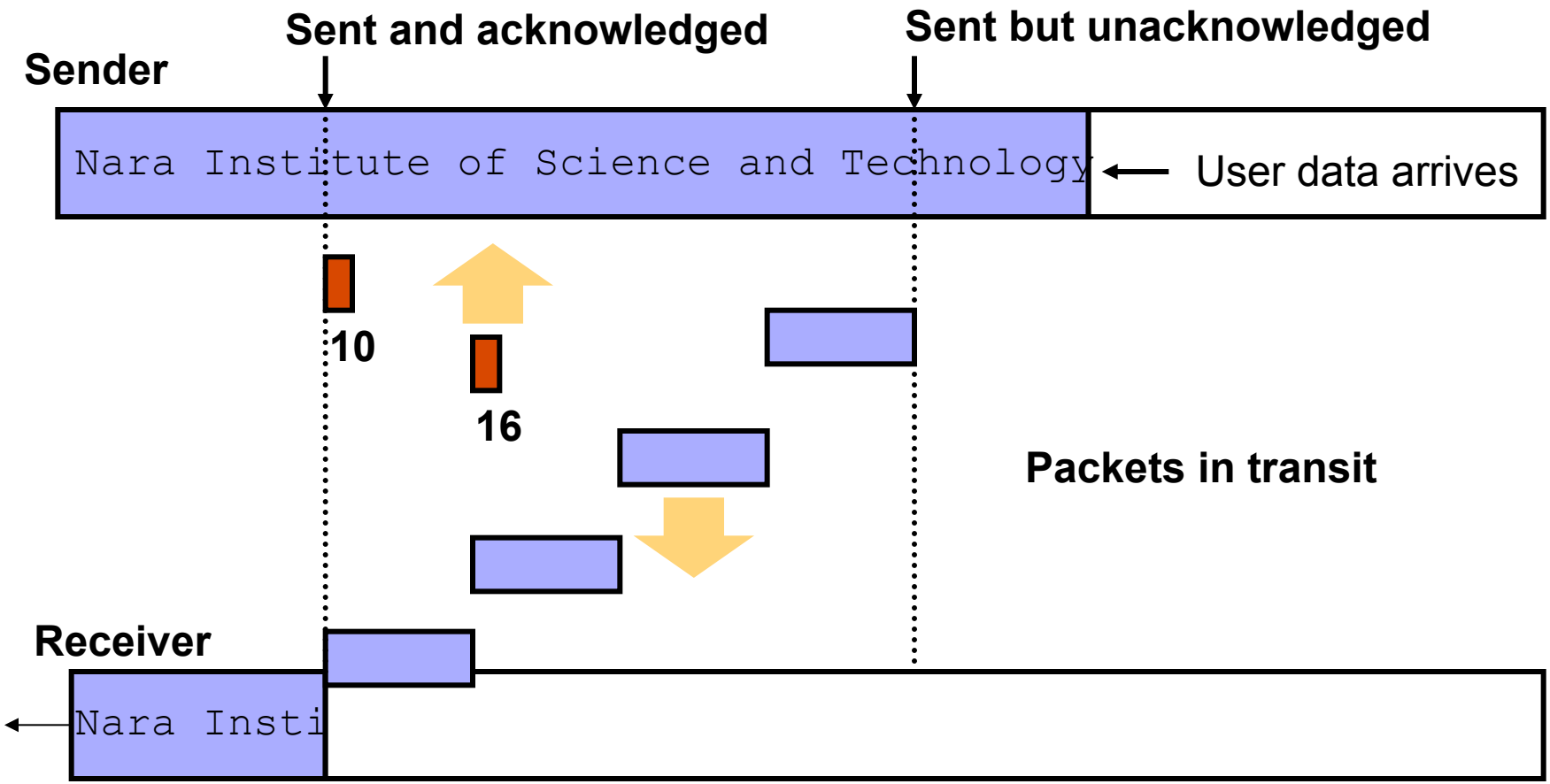
- Connection-oriented
- Byte-stream service
  - Upper layer can't see boundaries between packets
  - no boundary → structuring needed at upper layer
- Full duplex
  - independent two streams in single connection
- Reliability
  - masks packet reordering, duplication, discard and bit error



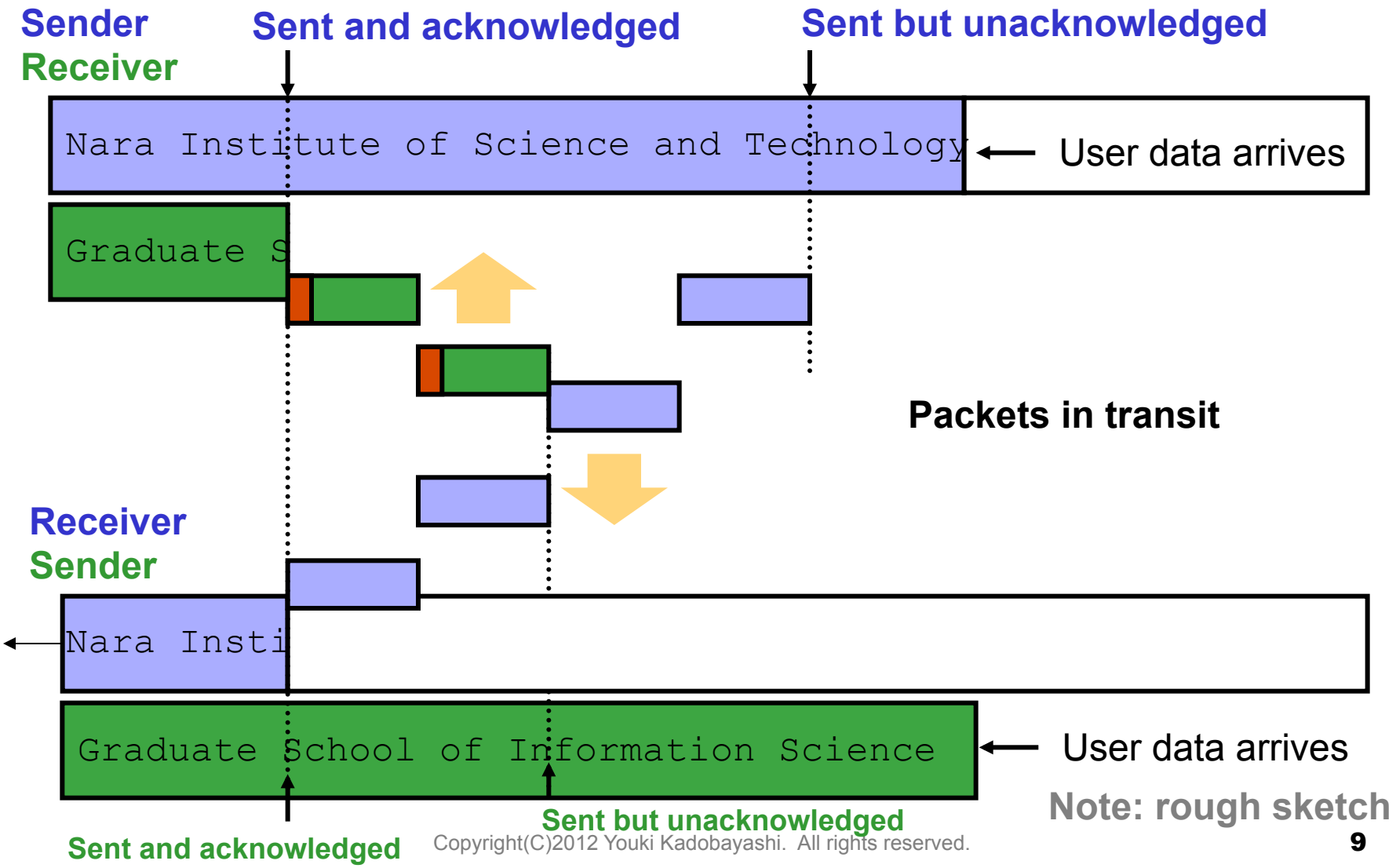
Implementing reliable stream service

- ACK: Acknowledgment
  - Active acknowledgment
    - Explicitly acknowledge the receipt of packet
  - Duplicate ACK
    - Implicitly communicate the packet loss information
- Timeout and Retransmission
  - If sender doesn't receive ACK after fixed time
    → Timeout
  - Retransmission assuming that transmission has failed
    - Exponential back-off

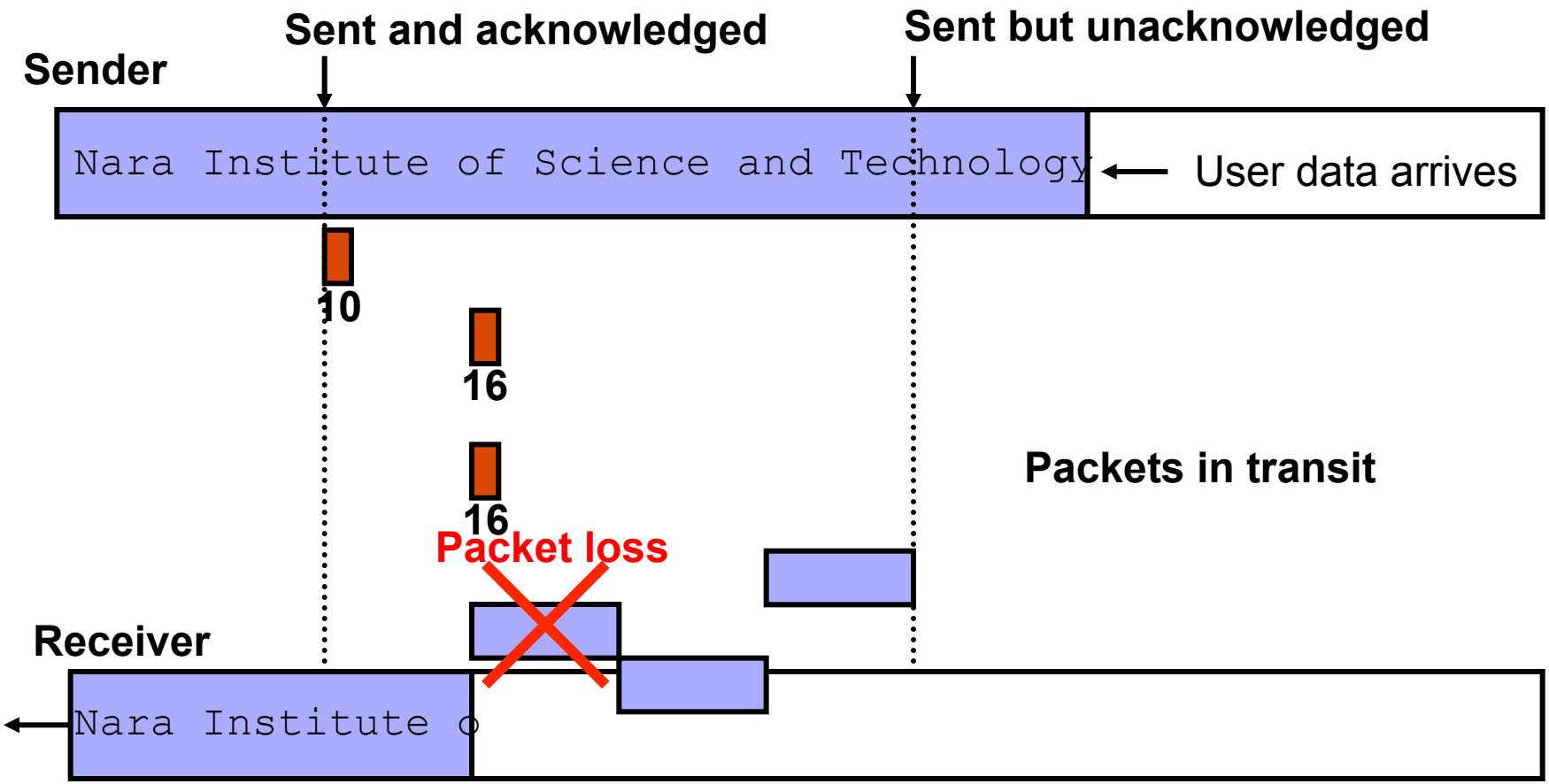
ACK



Piggybacking: Exploiting full-duplex channel



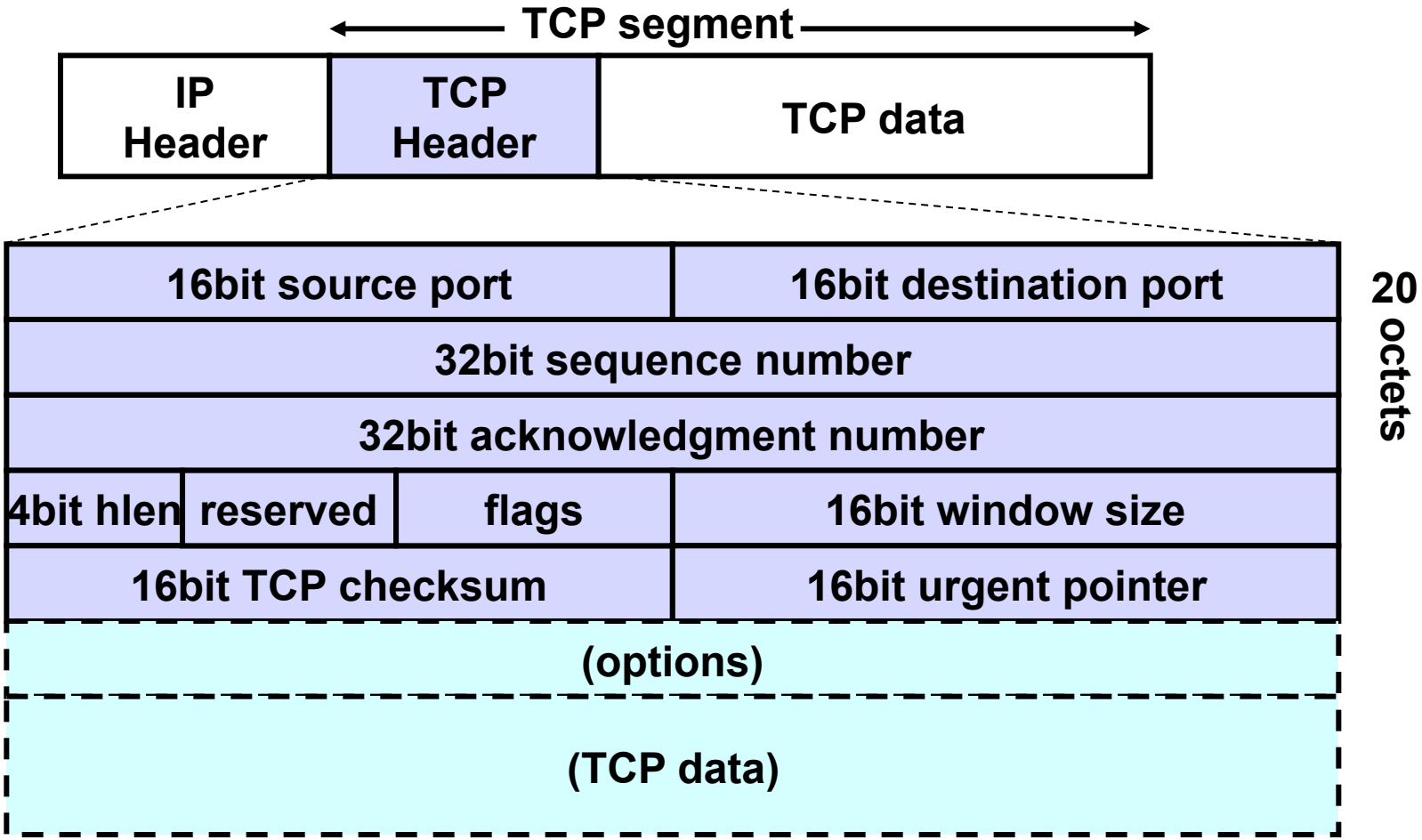
Duplicate ACK



Questions?

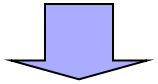
From byte-stream to packets: TCP header

Chop appropriate length from byte-stream buffer & add TCP header



Nagle algorithm

- Q. If you added 20byte+20byte size header to 1byte data, is that overhead big?



- Nagle algorithm (RFC896)
  - There is only one small segment which is unacknowledged in network.
  - In case of short RTT:

    → acceptable overhead; LAN b/w abundant

    → send packets with small buffering
  - In case of long RTT

    → reduce overhead, due to WAN b/w constraints

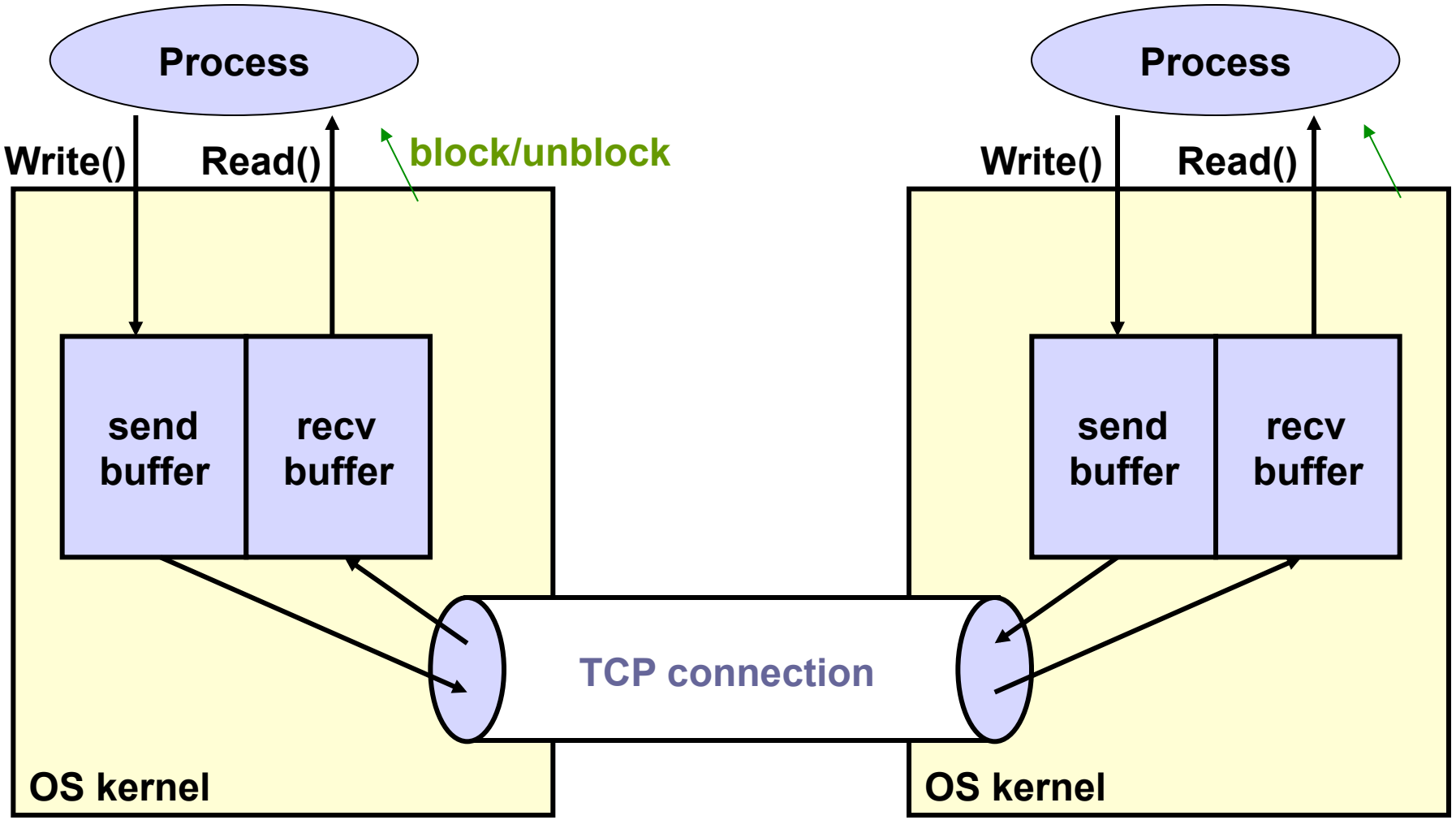
Q.

- In what kind of situation Nagle algorithm has to be turned off?

TCP service model (2)

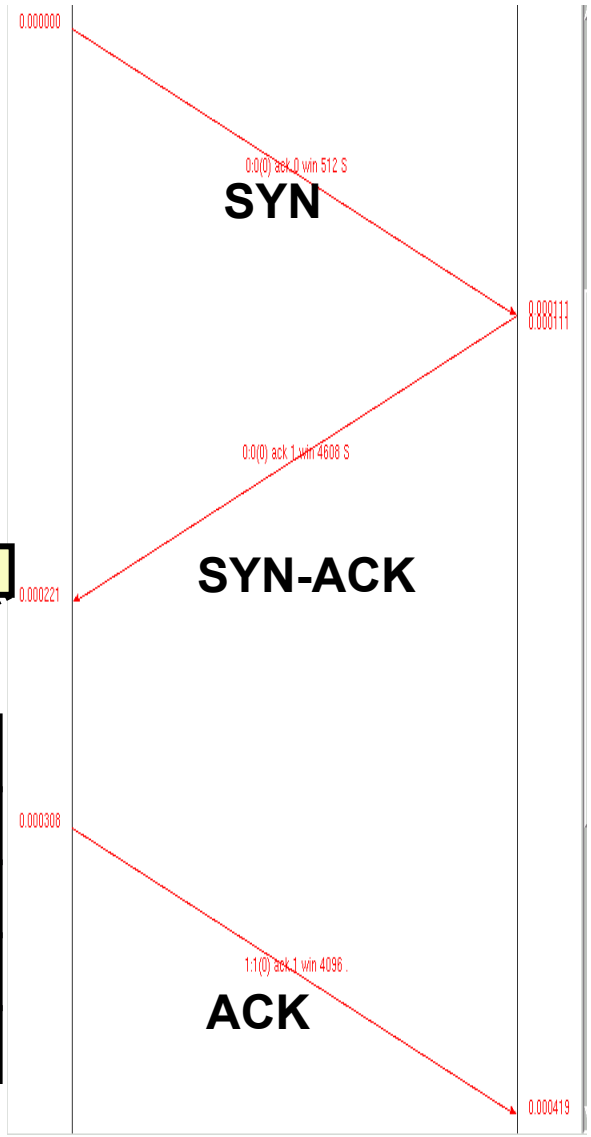
- Buffered Transfer
  - Virtually unlimited writes
  - doesn't need synchronization in application
  - change process state in operating system
- Virtual Circuit
  - connection setup and release
  - disconnection is detectable

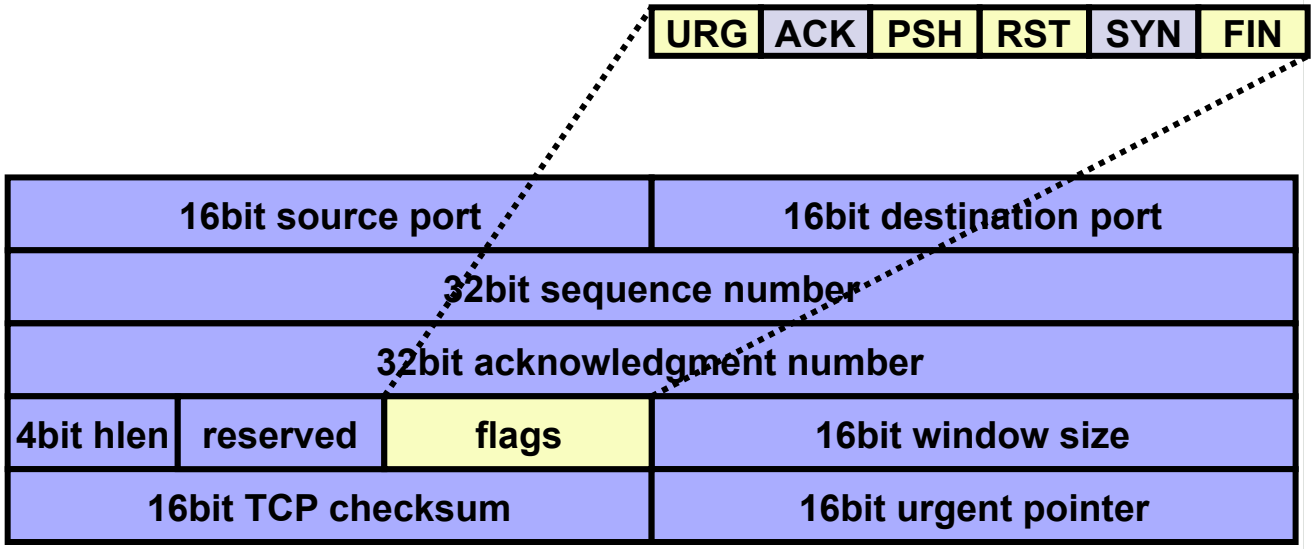
Buffered transfer: implicit speed adjustments



Virtual Circuit(1): TCP connection establishment

- 3-way handshake
- SYN, SYN-ACK, ACK
- Ensure full-duplex communication





TCP connection establishment: a concrete example with tcpdump

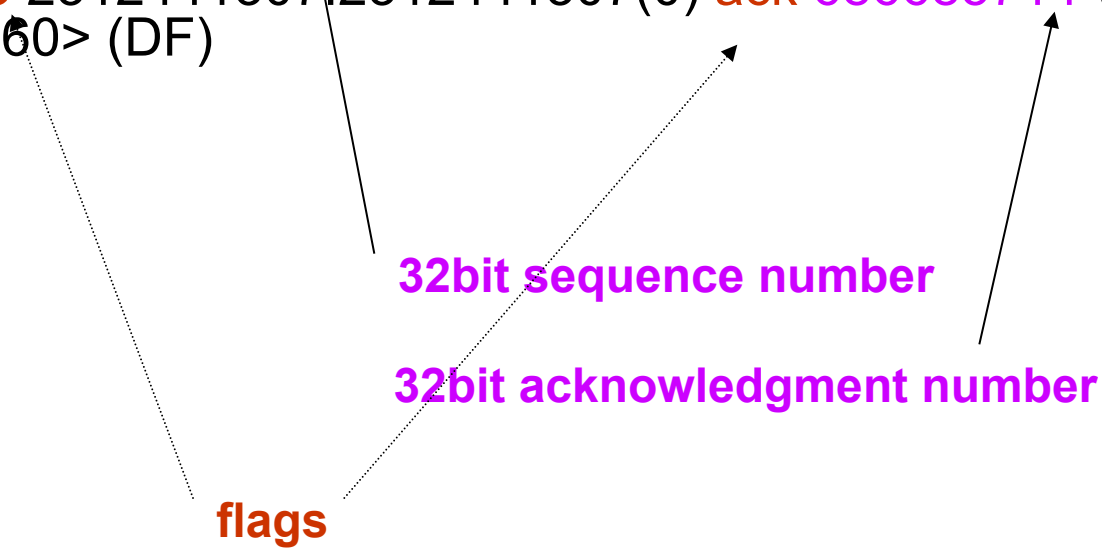
- dv# tcpdump tcp and host mint100.aist-nara.ac.jp
- tcpdump: listening on de0
- 12:16:00.146101 dv.aist-nara.ac.jp.49626 > mint100.aist-nara.ac.jp.telnet: S 686685713:686685713(0) win 65535 <mss 1460,nop,wscale 1,nop,nop,timestamp 110942140 0> (DF) [tos 0x10]
- 12:16:00.146350 mint100.aist-nara.ac.jp.telnet > dv.aist-nara.ac.jp.49626: S 2312441307:2312441307(0) ack 686685714 win 17520 <mss 1460> (DF)
- 12:16:00.146405 dv.aist-nara.ac.jp.49626 > mint100.aist-nara.ac.jp.telnet: . ack 1 win 65535 (DF) [tos 0x10]

Reply with Sequence number + 1 as an ACK implies acknowledgment

Meanings of tcpdump output

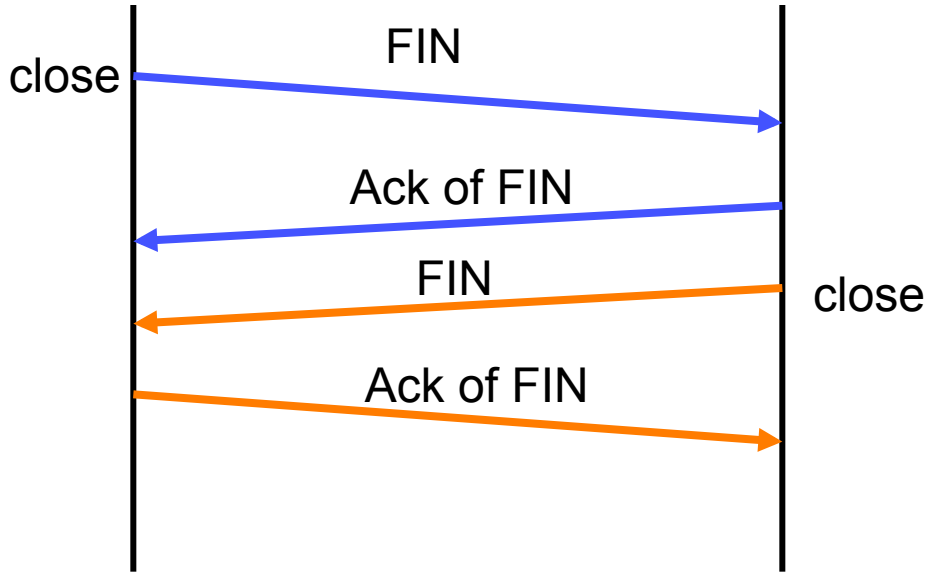
- time src.port > dst.port flag [from:to(nbytes) | ack #] win # opt

- 12:16:00.146101 dv.aist-nara.ac.jp.49626 > mint100.aist-nara.ac.jp.telnet: S 686685713:686685713(0) win 65535 <mss 1460,nop,wscale 1,nop,nop,timestamp 110942140 0> (DF) [tos 0x10]
- 12:16:00.146350 mint100.aist-nara.ac.jp.telnet > dv.aist-nara.ac.jp.49626: S 2312441307:2312441307(0) ack 686685714 win 17520 <mss 1460> (DF)



Questions?

Virtual Circuit(2):TCP connection release



12:16:07.086568 mint100.aist-nara.ac.jp.telnet > dv.aist-nara.ac.jp.49626: FP 713:721(8) ack 245 win 17520 (DF) [tos 0x10]
12:16:07.086738 dv.aist-nara.ac.jp.49626 > mint100.aist-nara.ac.jp.telnet: . ack 722 win 65535 (DF) [tos 0x10]
12:16:07.086998 dv.aist-nara.ac.jp.49626 > mint100.aist-nara.ac.jp.telnet: F 245:245(0) ack 722 win 65535 (DF) [tos 0x10]
12:16:07.087180 mint100.aist-nara.ac.jp.telnet > dv.aist-nara.ac.jp.49626: . ack 246 win 17519 (DF) [tos 0x10]

TCP connection reset

- RST
  - Abortive release
  - Nonexistent port

- 13:48:35.948096 dv.aist-nara.ac.jp.49635 > mint100.aist-nara.ac.jp.8080: S 1342403683:1342403683(0) win 65535 <mss 1460,nop,wscale 1,nop,nop,timestamp 111497668 0> (DF) [tos 0x10]
- 13:48:35.948265 mint100.aist-nara.ac.jp.8080 > dv.aist-nara.ac.jp.49635: R 0:0(0) ack 1342403684 win 0

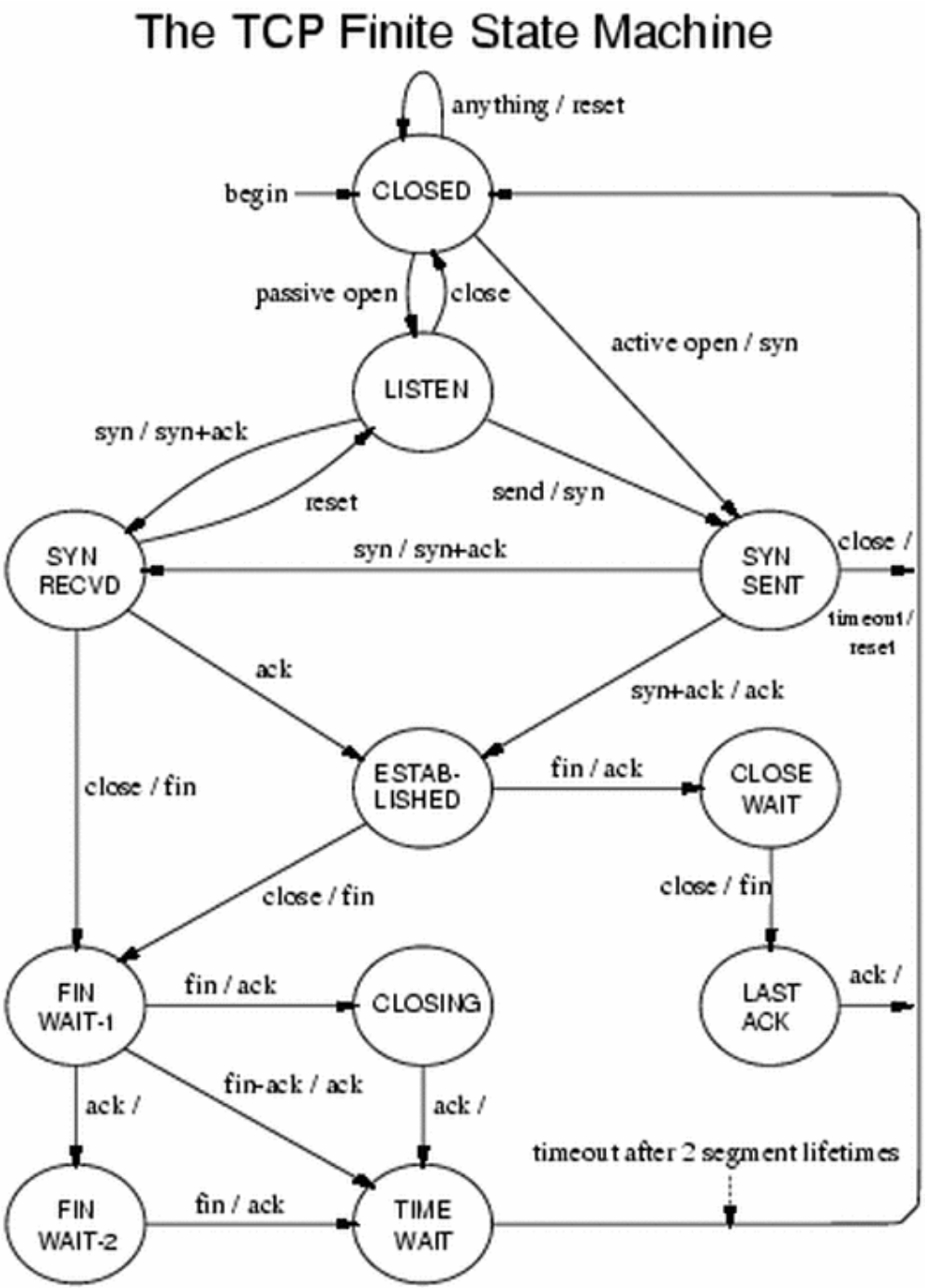
Virtual Circuit(3): Using extended features through TCP options

- TCP options in 3-way handshake
  - agrees with communication peer on their use

- 12:16:00.146101 dv.aist-nara.ac.jp.49626 > mint100.aist-nara.ac.jp.telnet: S 686685713:686685713(0) win 65535 <mss 1460,nop,wscale 1,nop,nop,timestamp 110942140 0> (DF) [tos 0x10]
  - MSS option (RFC793, Sep 1981)
  - Window scale option (RFC1323, May 1992)
  - Timestamp option (RFC1323)
  - Selective ACK option (RFC2018, Oct 1996)
  - etc.

Virtual Circuit Summary: TCP state transition diagram

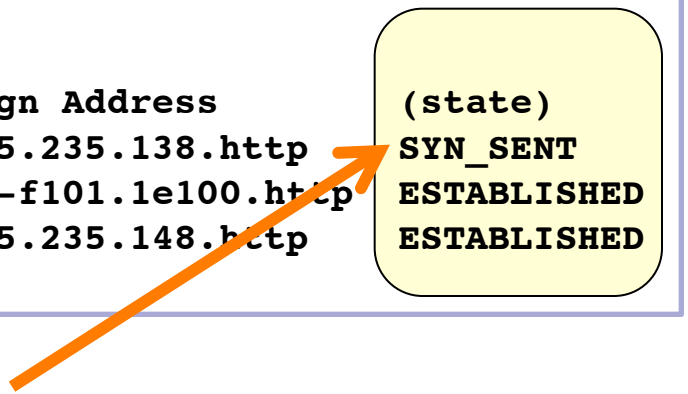
State transition:
trigger / response



Troubleshooting TCP with state machine

- netstat

```
$ netstat
Active Internet connections
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  45.1.20.101.57456     74.125.235.138.http    SYN_SENT
tcp4       0      0  45.1.20.101.57455     ey-in-f101.1e100.http  ESTABLISHED
tcp4       0      0  45.1.20.101.57454     74.125.235.148.http    ESTABLISHED
```



SYN sent, awaiting SYN+ACK response

Many other open-source tools are available:
lsof, trpt, tcptraceroute, tcptrace, tcpflow, etc.

Questions?

Conclusion

- Transport layer

- The transport protocol on the internet – TCP
- TCP: service model, and features

- Efficiency: ACK, piggybacking, Nagle algorithm
- TCP connection establishment and release
- Diagnosis: tools + state machine knowledge