Chapter3- Transport Layer

Transport Layer: build on the network layer to provide data delivery service for applications with the desired reliability and quality.

Topics:

- Transport-layer service
- multiplexing and demultiplexing
- connectionless transport UDP
- principles of reliable data transfer
- connection-oriented transport: TCP
- principles of congestion control
- TCP congestion control

Transport-layer service

- provide end-to-end connectivity across network, run in end systems
- more than one transport protocols:
    - TCP: reliable
    - UDP: best-effort

TCP	UDP
connections(streams)	datagrams
delivered once, reliably and in order	may be lost, reordered and duplicated
arbitrary length	limited message size
flow control	regardless of receiver state
congestion control	regardless of network state

multiplexing and demultiplexing

port: 16-bit integers representing local address of a process.

1-multiplexing(sender host)

gathering data from multiple sockets, enveloping with header.

- upward multiplexing: one network connection
- downward multiplexing: multiple network connections

2-demultiplexing(receiver host)

delivering received segments into correct sockets.

3-connection-oriented

TCP socket tuple (src port, src ip, dst port, dst ip) <-> one socket

δ : different <socket,process> share one port number possible

4-connectionless

UDP tuple(dst ip, dst port)

connectionless transport :UDP

1- characteristic

- connectionless: do not have handshaking
- best-effort: data may be lost, disorder, error.

δ : **there could be reliable application over UDP, eg. QUIC, google.**

2- UDP segment

(32-bit long)

source port #	destination port #
length	checksum

length: the byte length of UDP datagram.

3-checksum

the inverse of the sum of all 16-bit byte in the UDP segment.

δ: wraparound here

4- reason for UDP

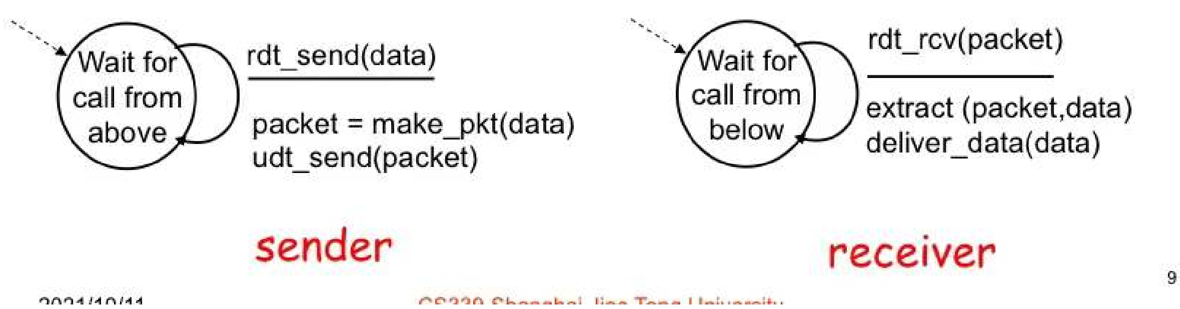
- no connection establishment
- no need to store the connection state
- more control for application layer
- shorter header

reliable data transfer protocol

Using FSM(finite state machine) to present for the senders and receivers.

1- reliable data transfer service (rdt)

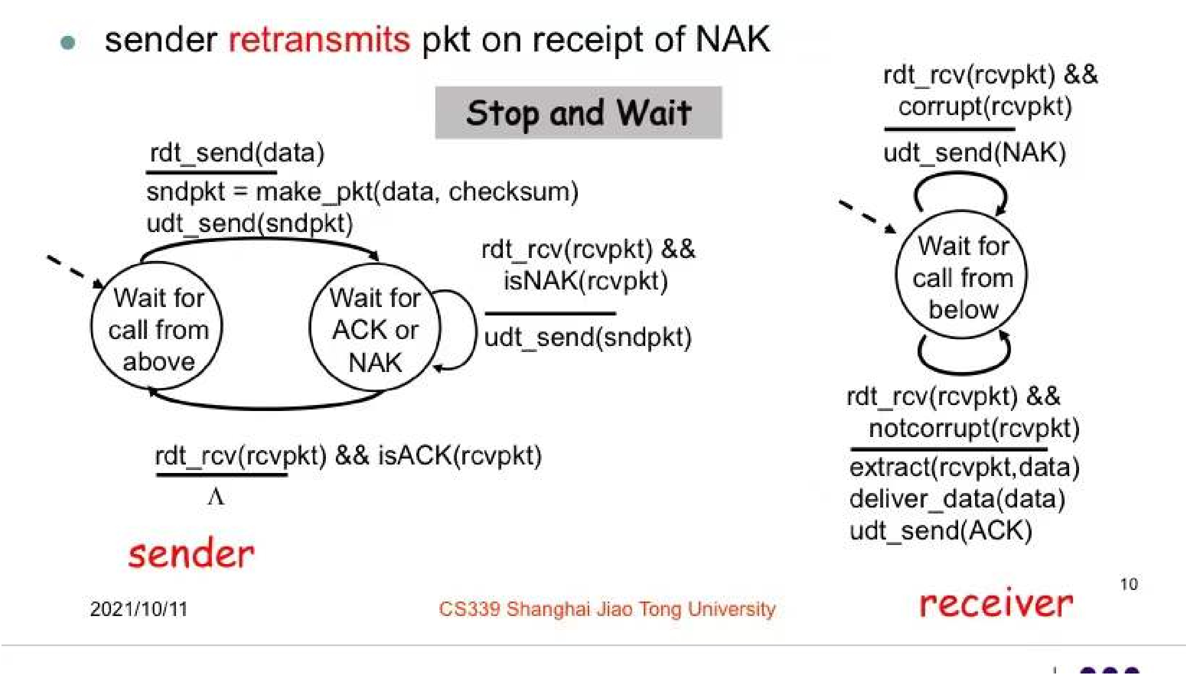
rdt1.0



rdt 1.0

hypothesis: the channel below is reliable. No data loss, no error, no disorder. And the receiver has enough buffer and CPU power.

rdt2.0



rdt2.0

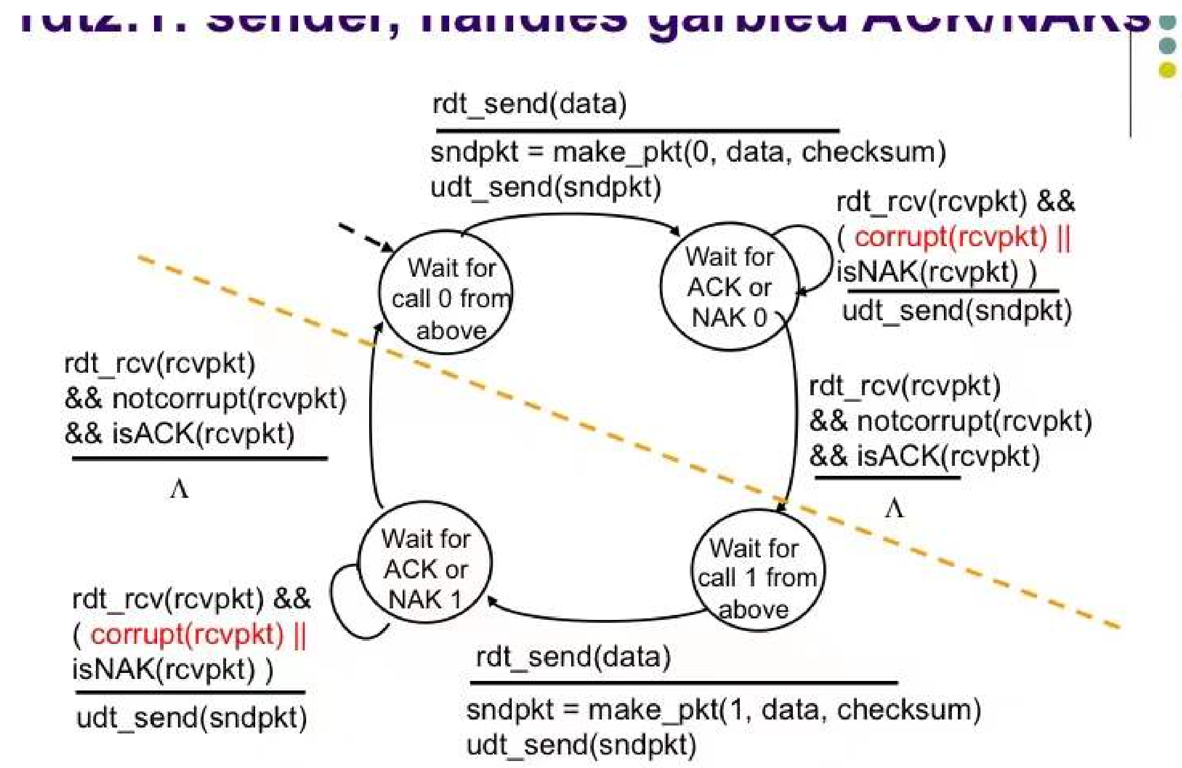
hypothesis: there could be data loss or data error.

ARQ(automatic repeat request): error detection, receiver feedback, retransmit

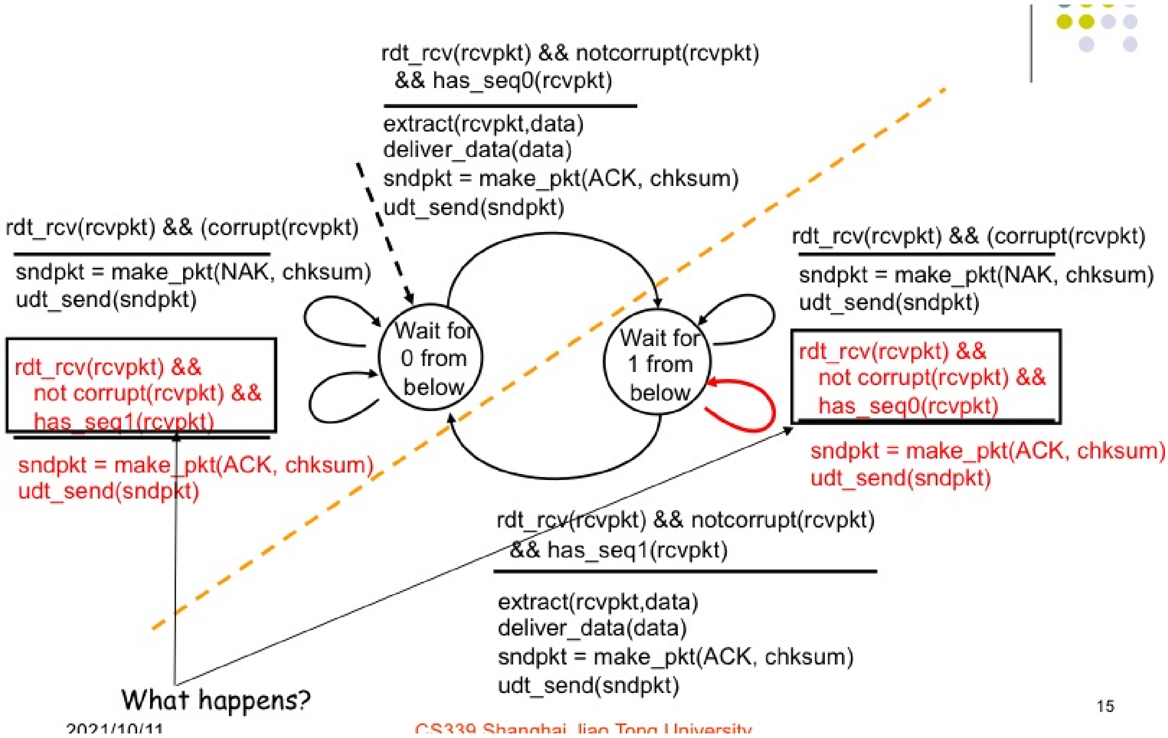
stop-and-wait protocol.

rdt2.1

To solve the problem that ACK/NCK could be corrupted, we just resend the packet, and introduce **sequence number** in.



rdt 2.1 sender

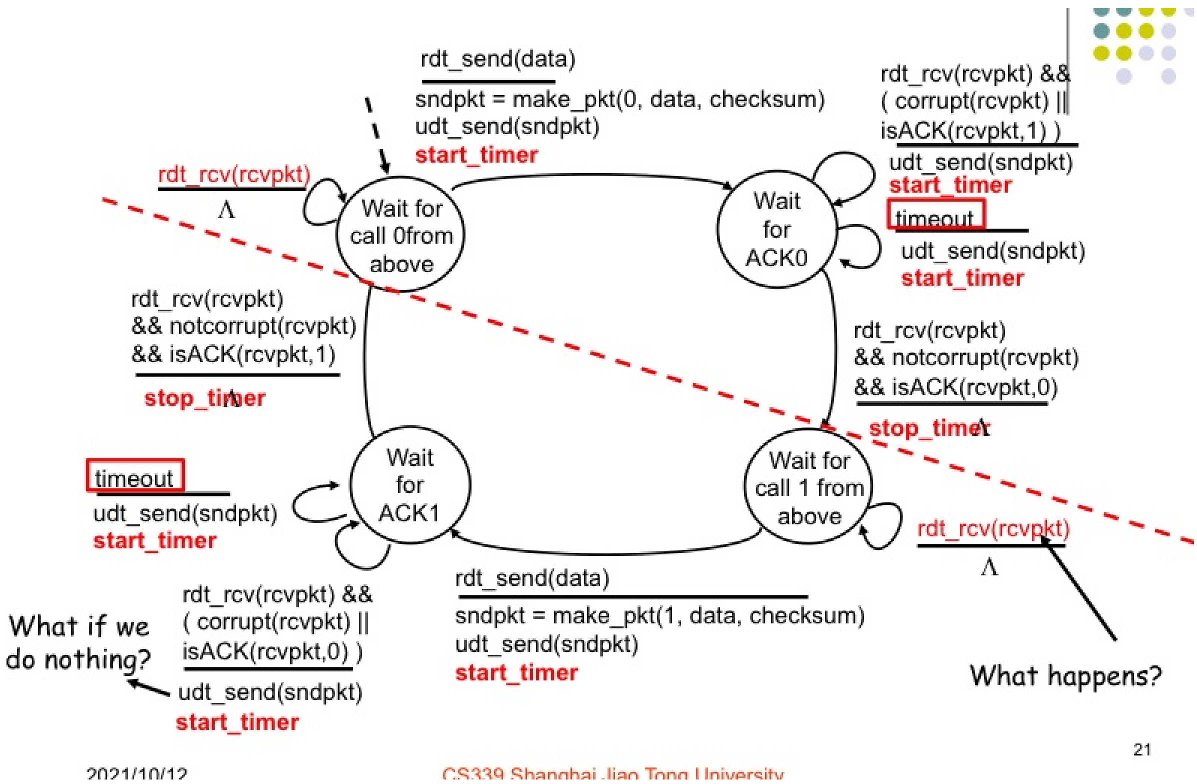


rdt 2.1 receiver

rdt2.2

Using duplicated ACK

Considering that the data may be lost, we introduce **countdown timeout**.



rdt3.0

2-pipelining protocols

Go-back-N	Selective Repeat
sender up to N unacked packets	sender up to N unacked packets
receiver one packet buffer	receiver up to N packets buffer
one timeout	m timer

3- error control

error types:

- bit error: using error detection,and retransmission
  - error detection

    Error Detection and Corrections bits(EDC)

    1. parity checking: single bit parity / two dimensional bit parity
    2. checksum
    3. CRC code(35条消息) CRC校验详解（附代码示例）_u013073067的博客-CSDN博客crc校验
    4. hamming code(35条消息) ECC校验——汉明码（Hamming Code）_agility9527的博客-CSDN博客ecc纠错码 matlab
  - error correction
- packet loss: using timer and retransmission
- duplicated packet/out-of-order: using sequence number

4-flow control

- stop and wait

$$Utility = \frac{L/R}{RTT + L/R}$$

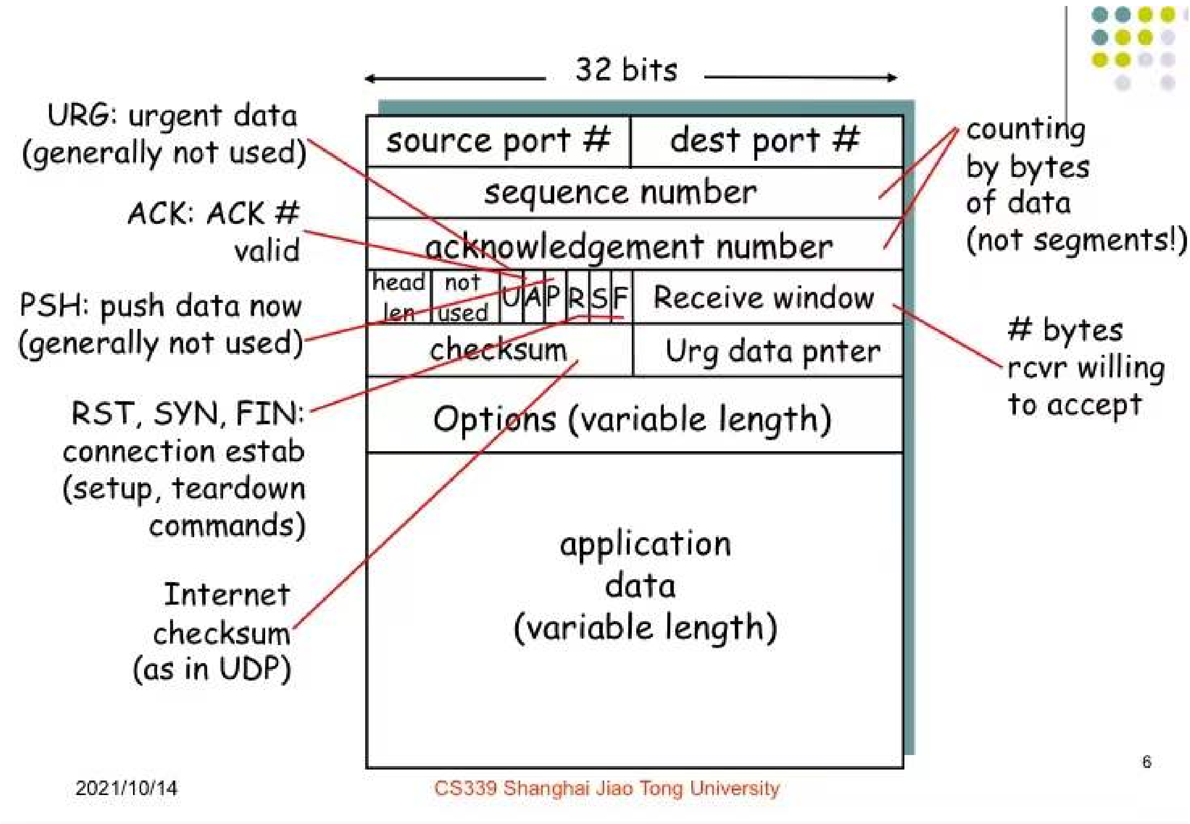
- sliding window

$$Utility = \frac{W * L/R}{RTT + L/R}$$

connection-oriented transport protocol: TCP

- end-end(not support broadcast)
- connection-oriented
- reliable byte stream
- full duplex
- flow & congestion control

1- TCP segment structure



TCP header is usually 20 bytes.

sequence number: byte stream "number" of the first byte in segment's data

ACK: the sequence number of next expected byte

The initial sequence number could be chosen randomly to avoid segment mistaken.

checksum: check the header, data and a pseudo header.

2- connection management

- connection establishment

  three-way handshaking

- connection release(with alive-timer)

  fin + ack

3- reliable data transfer

TCP reliable data transfer is hybrid of GBN and SR.

- retransmission
- interval double
- fast retransmit

4- timer management

$$RTT = \alpha RTT + (1-\alpha)M$$
$$D = \alpha D + (1-\alpha)|RTT - M|$$
$$timeout = RTT + 4D$$

in which, RTT is the best current estimate of round-trip delay, D is the estimate of deviation of round-trip delays, M is measured round-trip delay.

5- flow control

TCP flow control is achieve by Window Size Announcement.

if window size==0:

stop sending

except: urgent data/ reannounce requirement.

principles of congestion control

congestion: too much data, too fast for network to control. showing that lost packets and long delay.

reasons: data burst, lack of capacity/bandwidth, insufficient memory of routers, slow processors of routers.

TCP congestion control

- end to end: without explicit feedback from network **(TCP)**
- network-assisted: routers provide feedback, routers intelligent drop packets

Additive Increase, Multiplicative Decrease(AIMD)

slow start

congestion control

rapid back-off

fast recovery

If there are K **connections**, the average rate will be $\frac{R}{k}$.