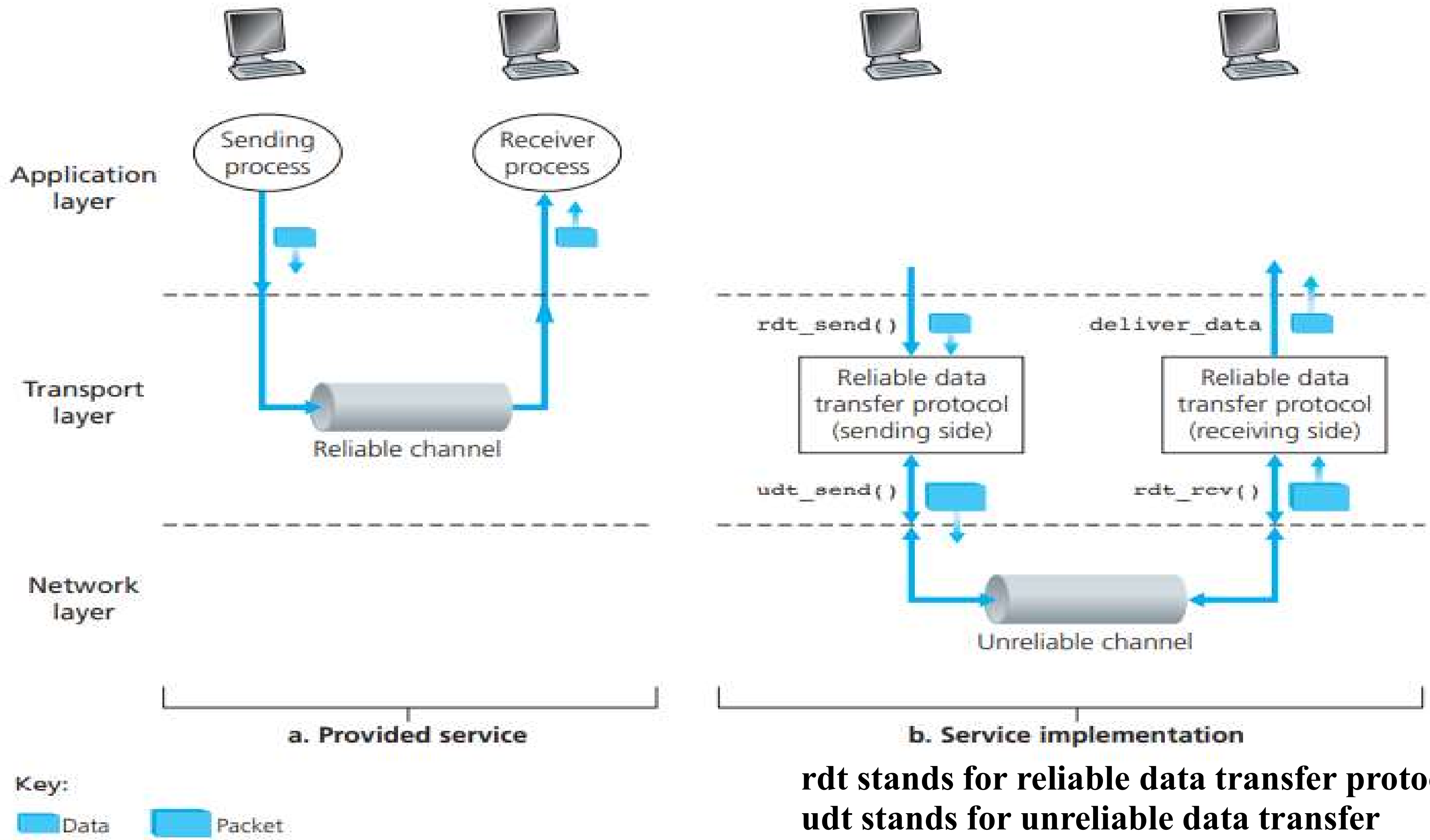
Principles of Reliable Data Transfer

- With a reliable channel, no transferred data bits are corrupted (flipped from 0 to 1, or vice versa) or lost, and all are delivered in the order in which they were sent.
- This is precisely the service model offered by TCP to the Internet applications that invoke it.
- It is the responsibility of a reliable data transfer protocol to implement this service abstraction.
- This task is made difficult by the fact that the layer below the reliable data transfer protocol may be unreliable.

Principles of Reliable Data Transfer



rdt stands for reliable data transfer protocol
udt stands for unreliable data transfer

Figure 2.17 ♦ Reliable data transfer: Service model and service implementation

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

Building a Reliable Data Transfer Protocol

- We can go through a series of protocols, each one becoming more complex, arriving at a flawless, reliable data transfer protocol.

A. Reliable Data Transfer over a Perfectly Reliable Channel: rdt1.0

- We first consider the simplest case, in which the underlying channel is completely reliable.
- The finite-state machine (FSM) definitions for the rdt1.0 sender and receiver are shown in Figure 2.18.

Principles of Reliable Data Transfer

A. Reliable Data Transfer over a Perfectly Reliable Channel: rdt1.0

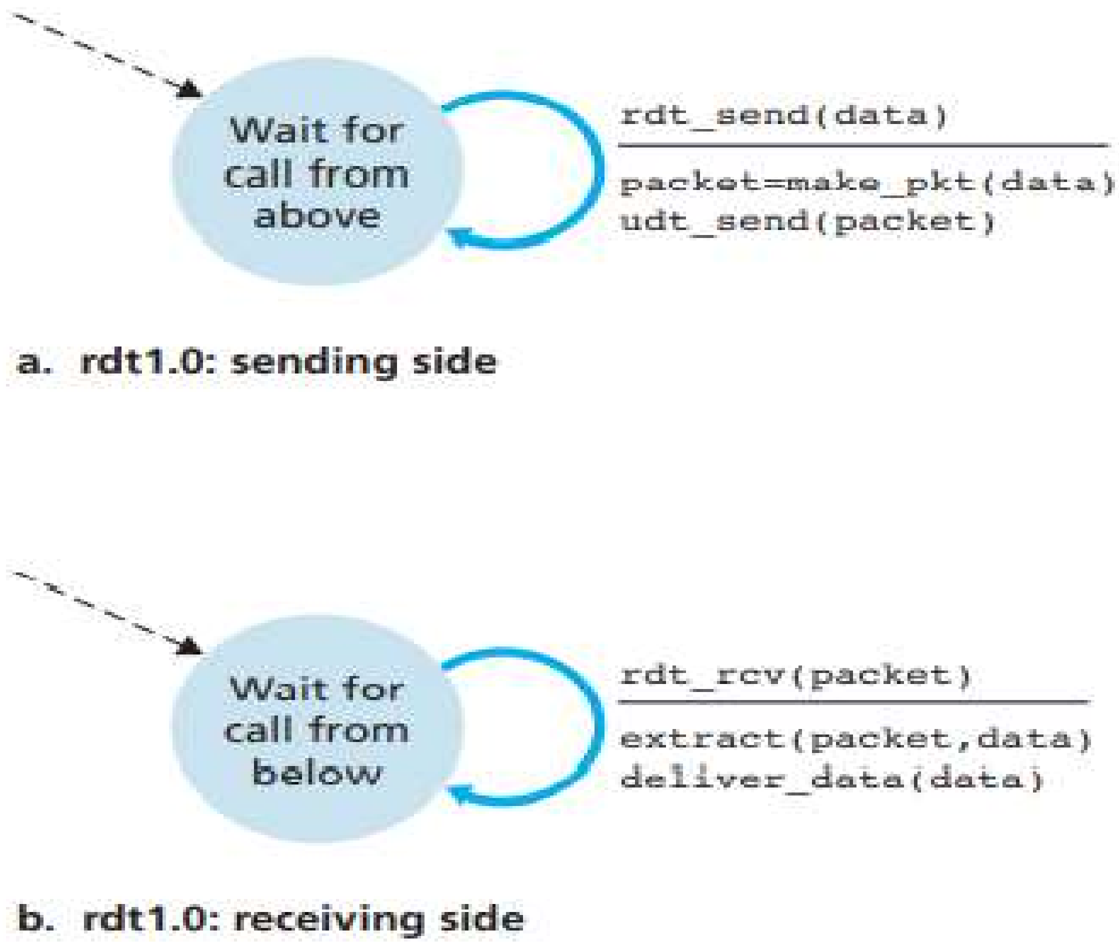


Figure 2.18 • rdt1.0 – A protocol for a completely reliable channel

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

B. Reliable Data Transfer over a Channel with Bit Errors: rdt2.0

- A more realistic model of the underlying channel is one in which bits in a packet may be corrupted.
- Such bit errors typically occur in the physical components of a network as a packet is transmitted, propagates, or is buffered.
- The message-dictation protocol uses both positive acknowledgments ("OK") and negative acknowledgments ("Please repeat that.").
- These control messages allow the receiver to let the sender know what has been received correctly, and what has been received in error and thus requires repeating.

Principles of Reliable Data Transfer

B. Reliable Data Transfer over a Channel with Bit Errors: rdt2.0

- In a computer network setting, reliable data transfer protocols based on retransmission are known as ARQ (Automatic Repeat reQuest) protocols.
- Fundamentally, three additional protocol capabilities are required in ARQ protocols to handle the presence of bit errors:
  - Error detection
  - Receiver feedback
  - Retransmission

- Figure 2.19 shows the FSM representation of rdt2.0, a data transfer protocol employing error detection, positive acknowledgments, and negative acknowledgments.

Principles of Reliable Data Transfer

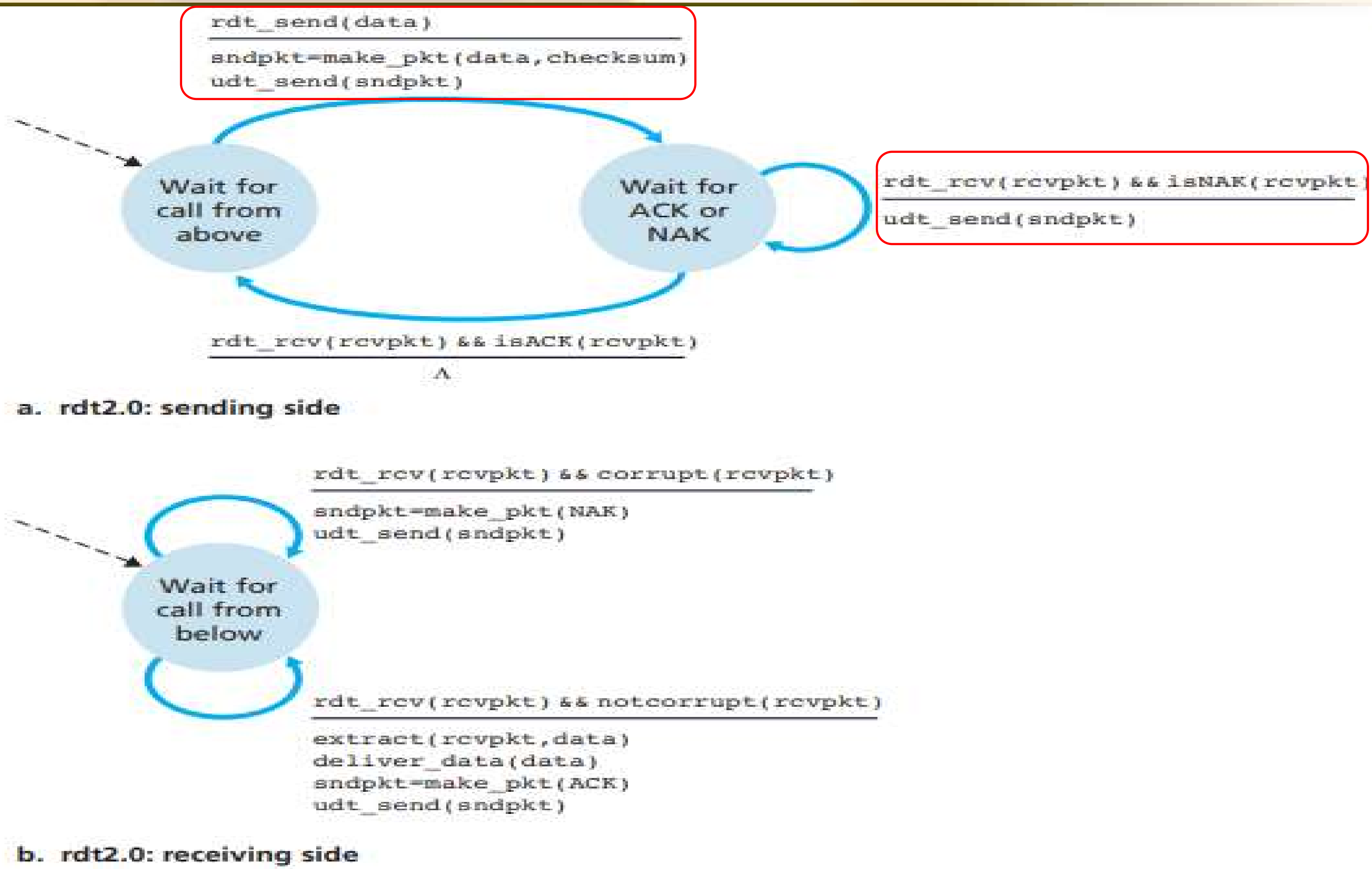


Figure 2.19 ♦ rdt2.0—A protocol for a channel with bit errors

James F Kurose and Keith W Ross, “Computer Networking: A Top - Down Approach”, Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

B. Reliable Data Transfer over a Channel with Bit Errors: rdt2.0

- The send side of rdt2.0 has two states.
- In the leftmost state, the send-side protocol is waiting for data to be passed down from the upper layer. When the rdt_send(data) event occurs, the sender will create a packet (sndpkt) containing the data to be sent, along with a packet checksum and then send the packet via the udt_send(sndpkt) operation.
- In the rightmost state, the sender protocol is waiting for an ACK or a NAK packet from the receiver.

Principles of Reliable Data Transfer

B. Reliable Data Transfer over a Channel with Bit Errors: rdt2.0

- It is important to note that when the sender is in the wait-for-ACK-or-NAK state, it cannot get more data from the upper layer; that is, the rdt_send() event can not occur; that will happen only after the sender receives an ACK and leaves this state.
- Thus, the sender will not send a new piece of data until it is sure that the receiver has correctly received the current packet.
- Because of this behavior, protocols such as rdt2.0 are known as stop-and-wait protocols.

stop-and-wait protocols

Principles of Reliable Data Transfer

B. Reliable Data Transfer over a Channel with Bit Errors: rdt2.0

- The receiver-side FSM for rdt2.0 still has a single state.
- On packet arrival, the receiver replies with either an ACK or a NAK, depending on whether or not the received packet is corrupted.
- In Figure 2.19, the notation rdt_rcv(rcvpkt) && corrupt(rcvpkt) corresponds to the event in which a packet is received and is found to be in error.

stop-and-wait protocols

Principles of Reliable Data Transfer

Reliable Data Transfer over a Channel with Bit Errors: rdt2.0

Figure 2.19 ♦ rdt2.0—A protocol for a channel with bit errors

James F Kurose and Keith W Ross, “Computer Networking: A Top - Down Approach”, Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

B. Reliable Data Transfer over a Channel with Bit Errors: rdt2.0

- Protocol rdt2.0 has a fatal flaw. We haven't accounted for the possibility that the ACK or NAK packet could be corrupted! AND whether an arriving packet contains new data or is a retransmission!
- A simple solution to this new problem (and one adopted in almost all existing data transfer protocols, including TCP) is to add a new field to the data packet and have the sender number its data packets by putting a sequence number into this field.
- The receiver then need only check this sequence number to determine whether or not the received packet is a retransmission.

Principles of Reliable Data Transfer

B.a. Reliable Data Transfer over a Channel with Bit Errors: rdt2.1

- For this simple case of a stop-and wait protocol, a 1-bit sequence number will suffice, since it will allow the receiver to know whether the sender is resending the previously transmitted packet (the sequence number of the received packet has the same sequence number as the most recently received packet) or a new packet (the sequence number changes, moving "forward" in modulo-2 arithmetic).
- Protocol rdt2.1 uses both positive and negative acknowledgments from the receiver to the sender.

Principles of Reliable Data Transfer

B. a. Reliable Data Transfer over a Channel with Bit Errors: rdt2.1

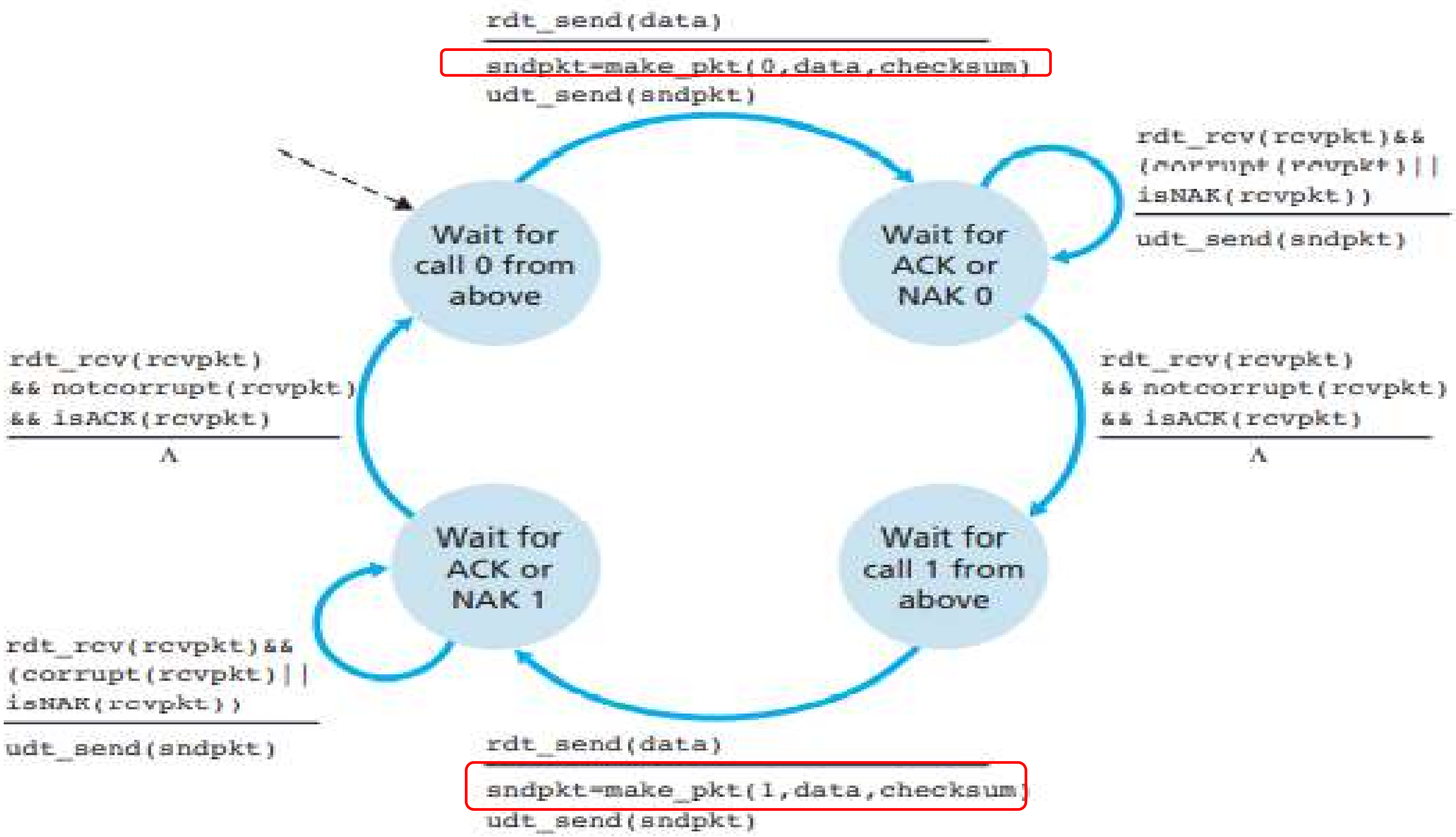


Figure 2.20 ♦ rdt2.1 sender

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

B. a. Reliable Data Transfer over a Channel with Bit Errors: rdt2.1

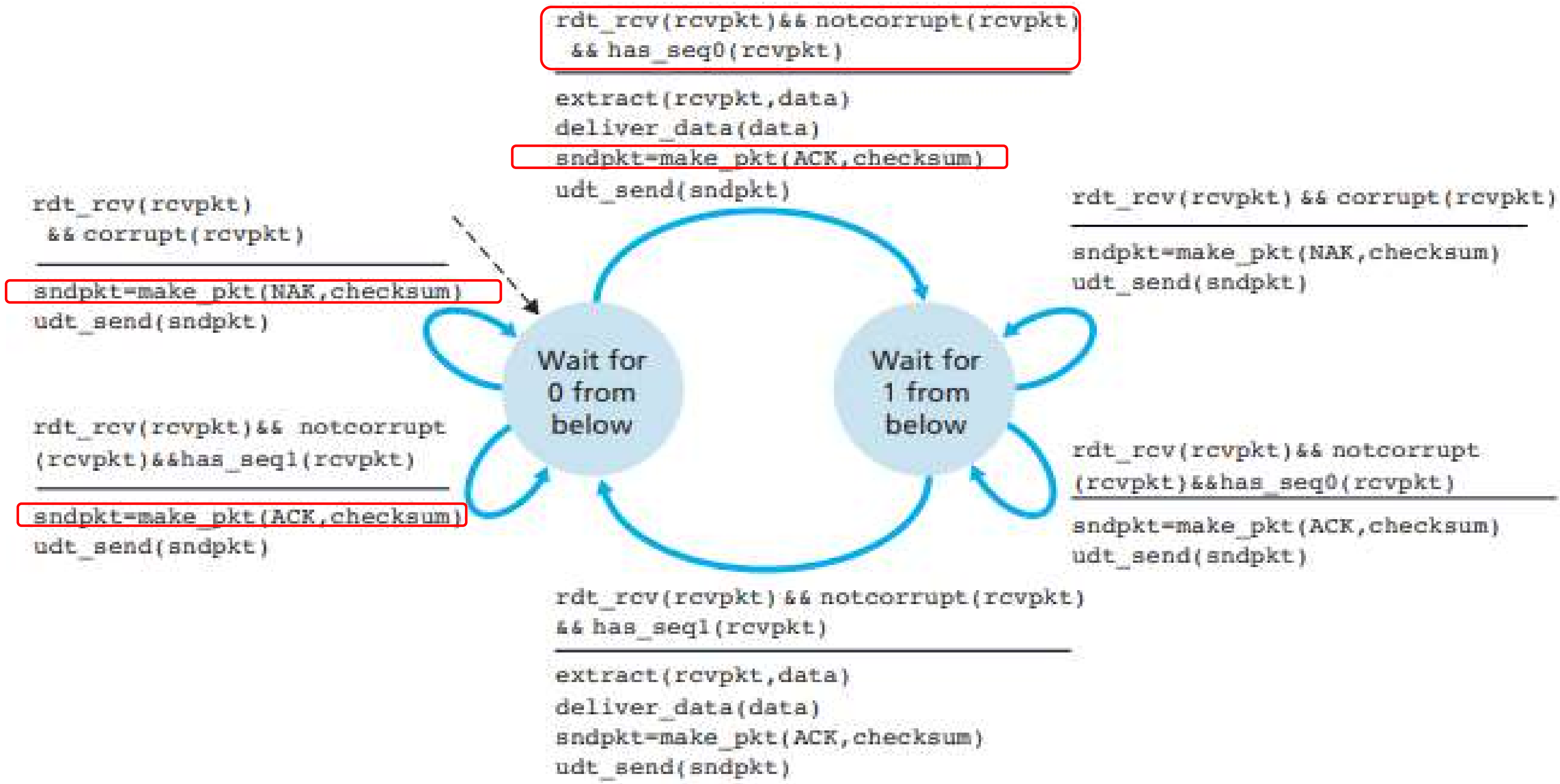


Figure 2.21 ♦ rdt2.1 receiver

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

B. b. Reliable Data Transfer over a Channel with Bit Errors: rdt2.2

- One subtle change between rtdt2.1 and rdt2.2 is that the receiver must now include the sequence number of the packet being acknowledged by an ACK message (this is done by including the ACK0 or ACK1 argument in make_pkt() in the receiver FSM), and the sender must now check the sequence number of the packet being acknowledged by a received ACK message (this is done by including the 0 or 1 argument in isACK() in the sender FSM).

Principles of Reliable Data Transfer

B. b. Reliable Data Transfer over a Channel with Bit Errors: rdt2.2

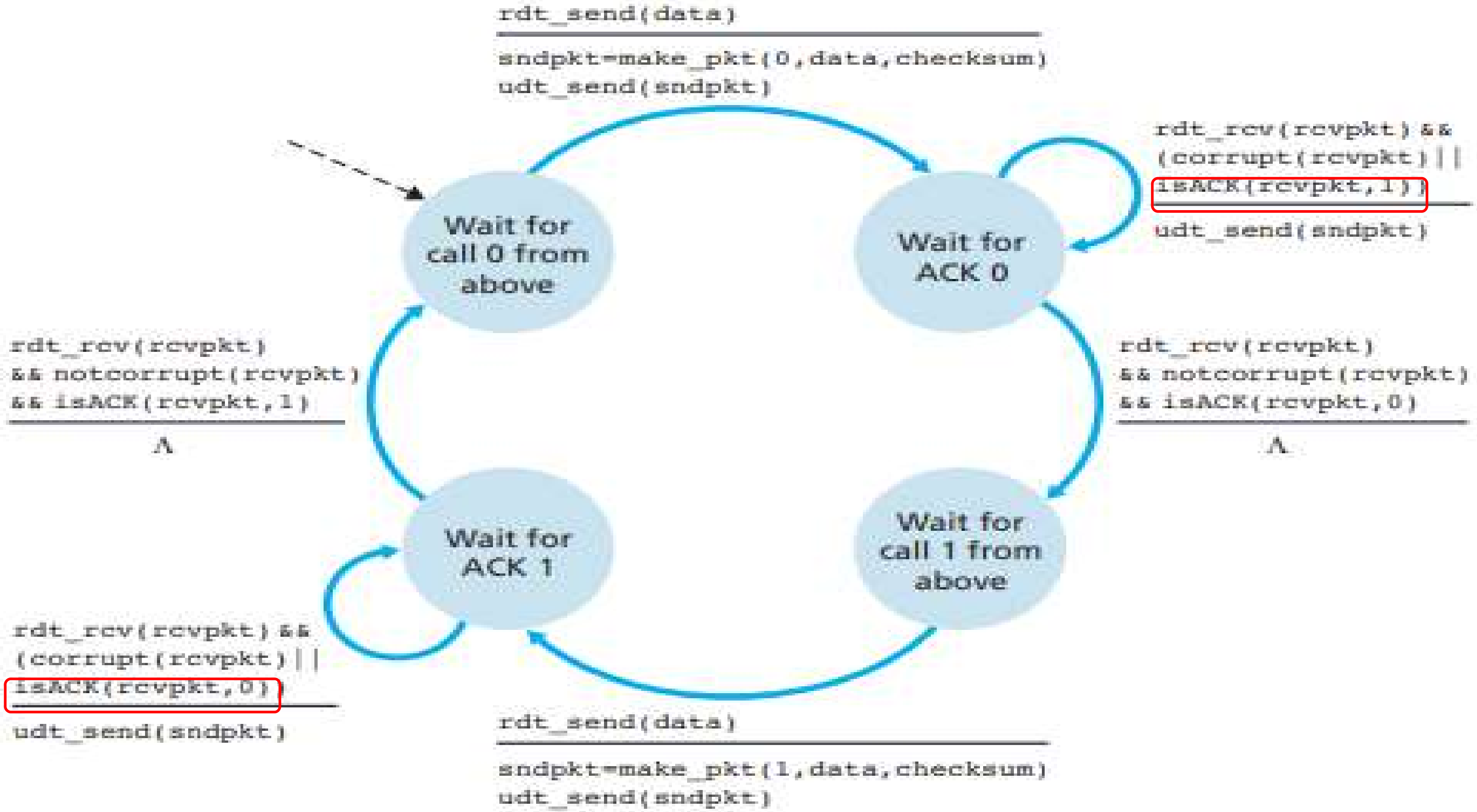


Figure 2.22 ♦ rdt2.2 sender

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

B. b. Reliable Data Transfer over a Channel with Bit Errors: rdt2.2

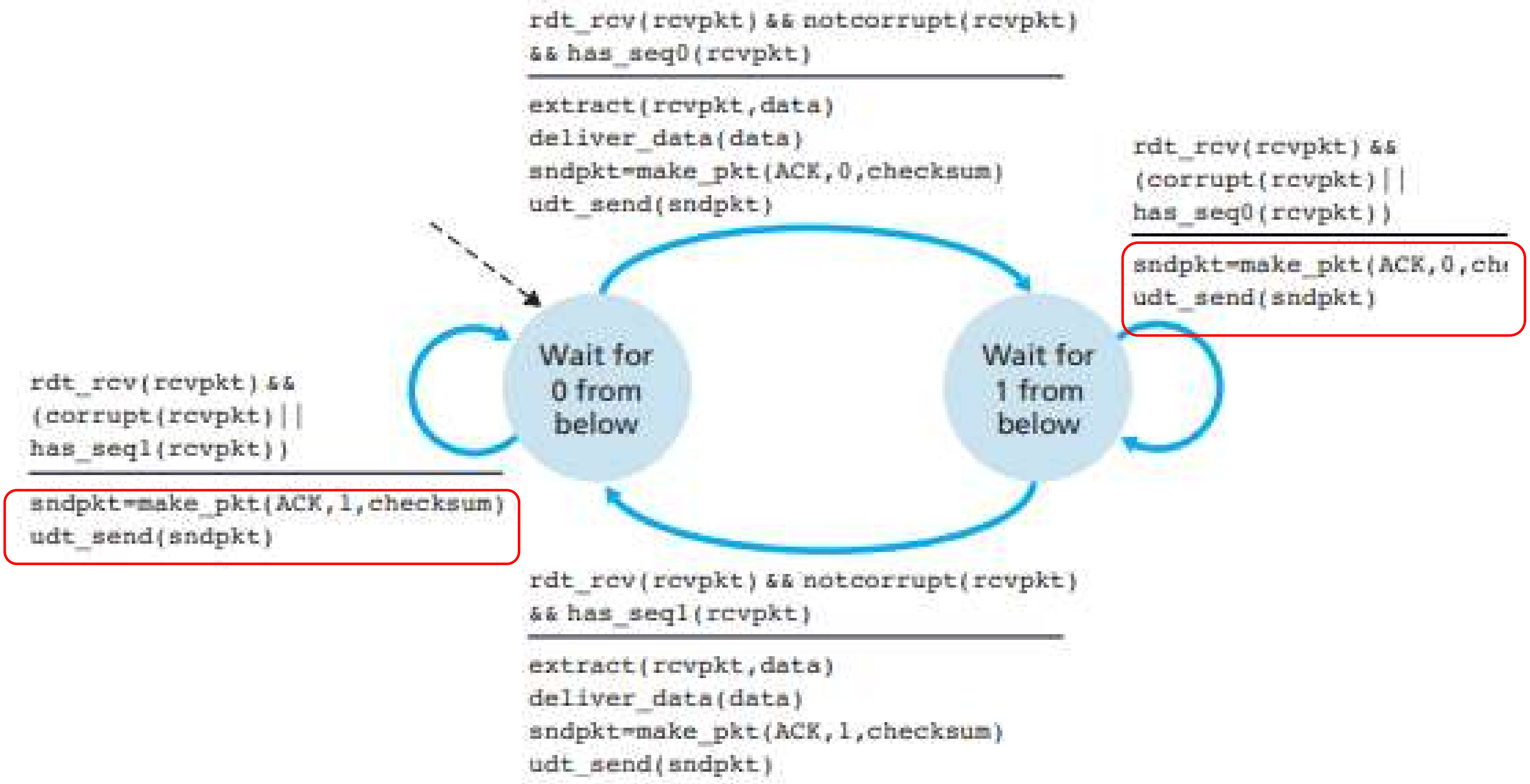


Figure 2.23 ♦ rdt2.2 receiver

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

C. Reliable Data Transfer over a Lossy Channel with Bit Errors: rdt3.0

- Suppose now that in addition to corrupting bits, the underlying channel can lose packets as well, a not-uncommon event in today's computer networks (including the Internet).
- Two additional concerns must now be addressed by the protocol: how to detect packet loss and what to do when packet loss occurs.
- The use of checksumming, sequence numbers, ACK packets, and retransmissions—the techniques already developed in rdt2.2—will allow us to answer the latter concern.
- Handling the first concern will require adding a new protocol mechanism.

Principles of Reliable Data Transfer

C. Reliable Data Transfer over a Lossy Channel with Bit Errors: rdt3.0

- Implementing a time-based retransmission mechanism requires a countdown timer that can interrupt the sender after a given amount of time has expired. The sender will thus need to be able to (1) start the timer each time a packet (either a first-time packet or a retransmission) is sent, (2) respond to a timer interrupt (taking appropriate actions), and (3) stop the timer.
- Because packet sequence numbers alternate between 0 and 1, protocol rdt3.0 is sometimes known as the alternating-bit protocol.

alternating-bit protocol

Principles of Reliable Data Transfer

C. Reliable Data Transfer over a Lossy Channel with Bit Errors: rdt3.0

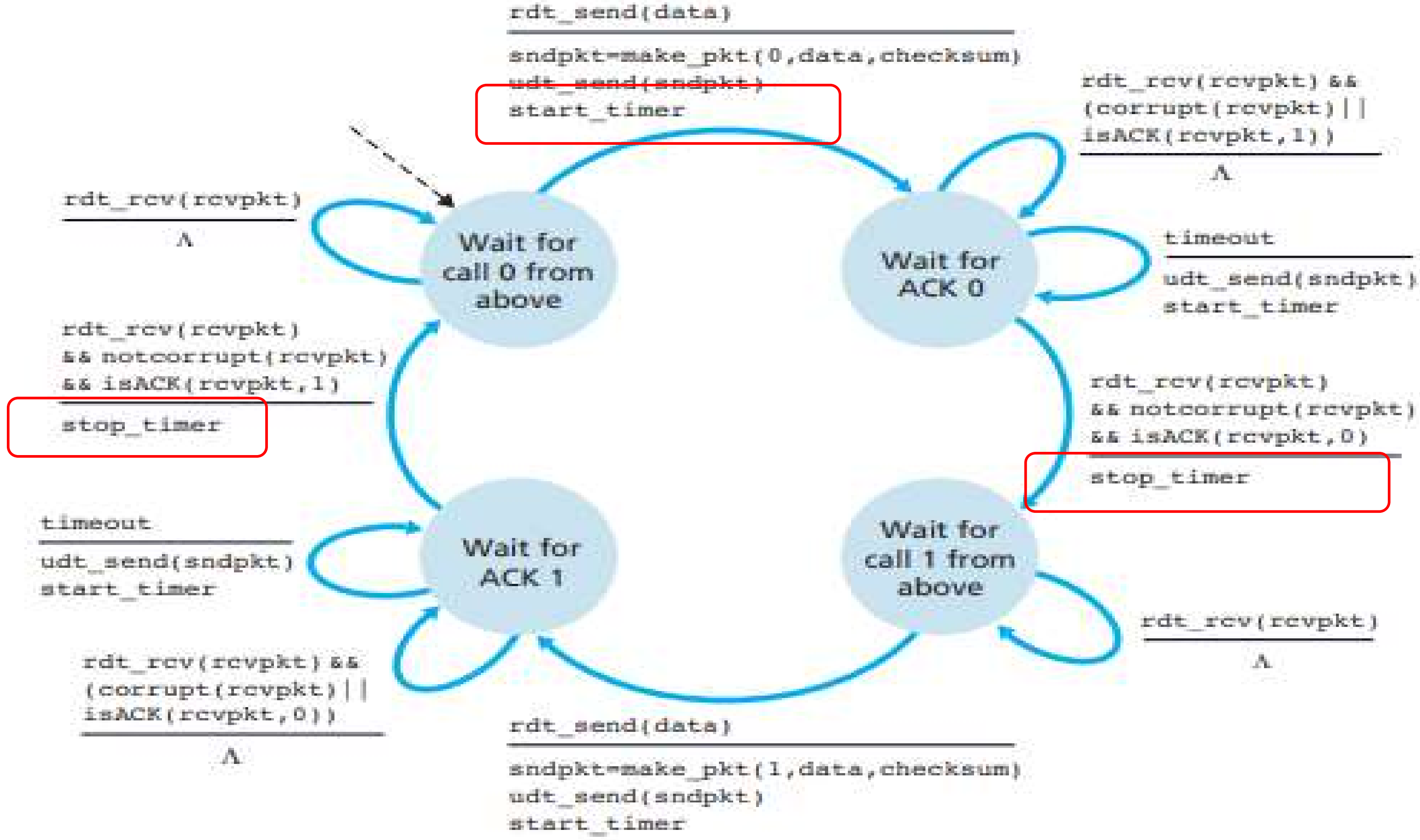


Figure 2.24 ♦ rdt3.0 sender

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

C. Reliable Data Transfer over a Lossy Channel with Bit Errors: rdt3.0

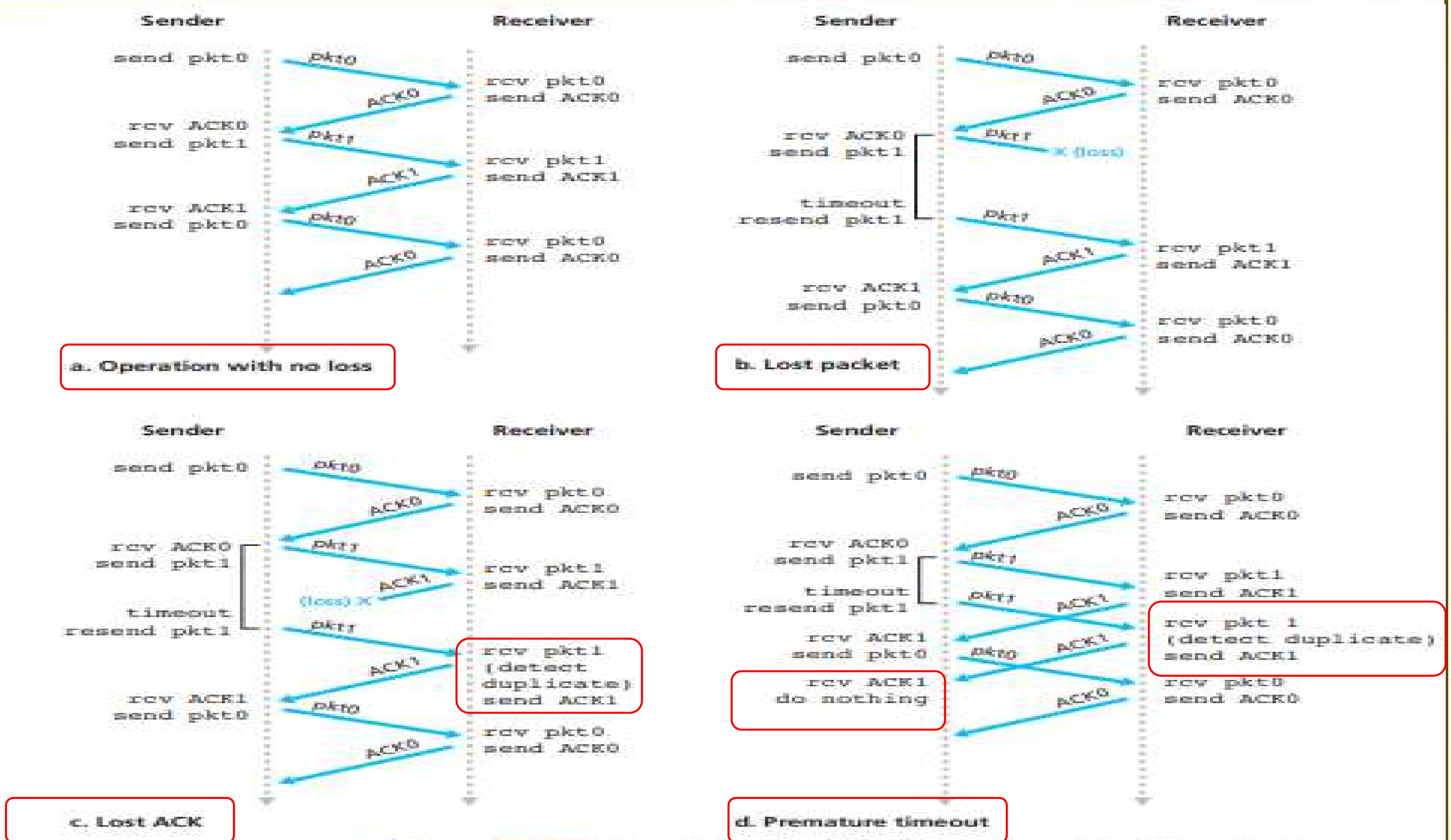


Figure 2.25 • Operation of rdt3.0, the alternating-bit protocol

James F Kurose and Keith W Ross, “Computer Networking: A Top - Down Approach”, Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

D. Pipelined Reliable Data Transfer Protocols

- Protocol rdt3.0 is a functionally correct protocol, but it is unlikely that anyone would be happy with its performance, particularly in today's high-speed networks.
- At the heart of rdt3.0's performance problem is the fact that it is a stop-and-wait protocol.

Principles of Reliable Data Transfer

D. Pipelined Reliable Data Transfer Protocols

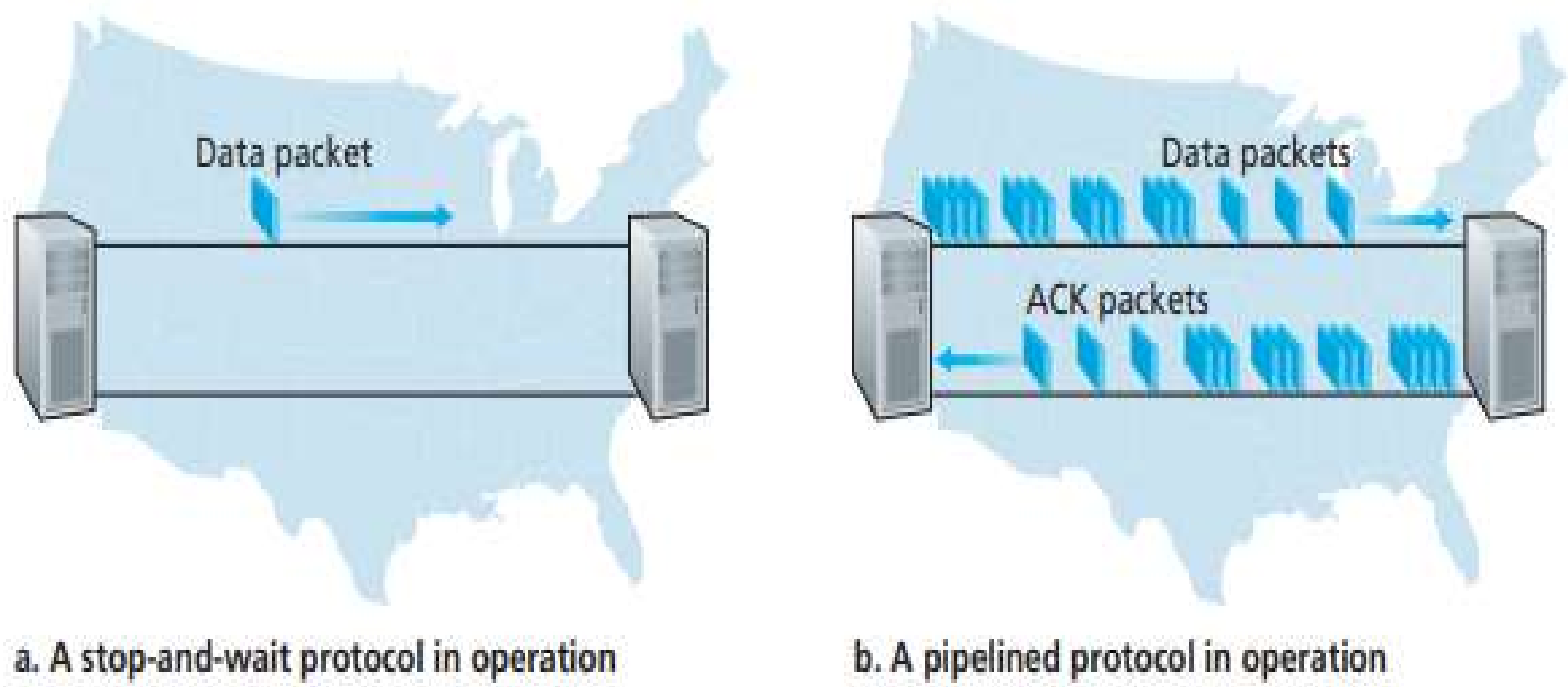


Figure 2.26 ♦ Stop-and-wait versus pipelined protocol

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

D. Pipelined Reliable Data Transfer Protocols

Stop-and-wait protocol

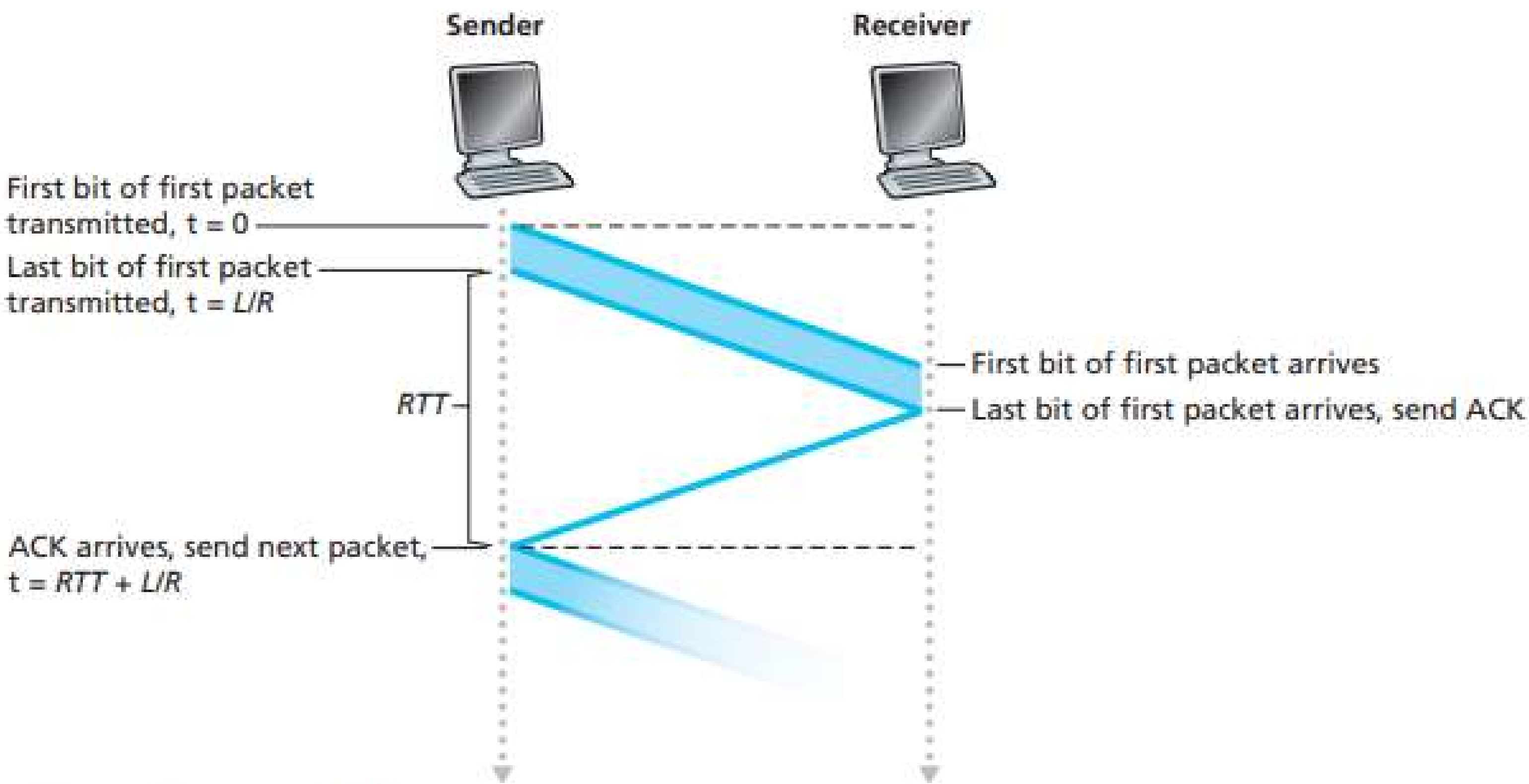
- The speed-of-light round-trip propagation delay between two end systems, RTT, is approximately 30 milliseconds.
- Suppose that they are connected by a channel with a transmission rate, R, of 1 Gbps (10^9 bits per second). With a packet size, L, of 1,000 bytes (8,000 bits) per packet, including both header fields and data, the time needed to actually transmit the packet into the 1 Gbps link is

$$d_{trans} = \frac{L}{R} = \frac{8000 \text{ bits/packet}}{10^9 \text{ bits/sec}} = 8 \text{ microseconds}$$

Principles of Reliable Data Transfer

Pipelined Reliable Data Transfer Protocols

Stop-and-wait protocol



a. Stop-and-wait operation

James F Kurose and Keith W Ross, “Computer Networking: A Top - Down Approach”, Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

Pipelined Reliable Data Transfer Protocols

Stop-and-wait protocol

- The packet then makes its 15-msec cross-country journey, with the last bit of the packet emerging at the receiver at t = RTT/2 + L/R = 15.008 msec.
- The receiver can send an ACK as soon as the last bit of a data packet is received, the ACK emerges back at the sender at t = RTT + L/R = 30.008 msec. (Ignore ACK transmission time)
- At this point, the sender can now transmit the next message. Thus, in 30.008 msec, the sender was sending for only 0.008 msec.
- If we define the utilization of the sender (or the channel) as the fraction of time the sender is actually busy sending bits into the channel, the stop-and-wait protocol has a sender utilization, U_{sender}, of

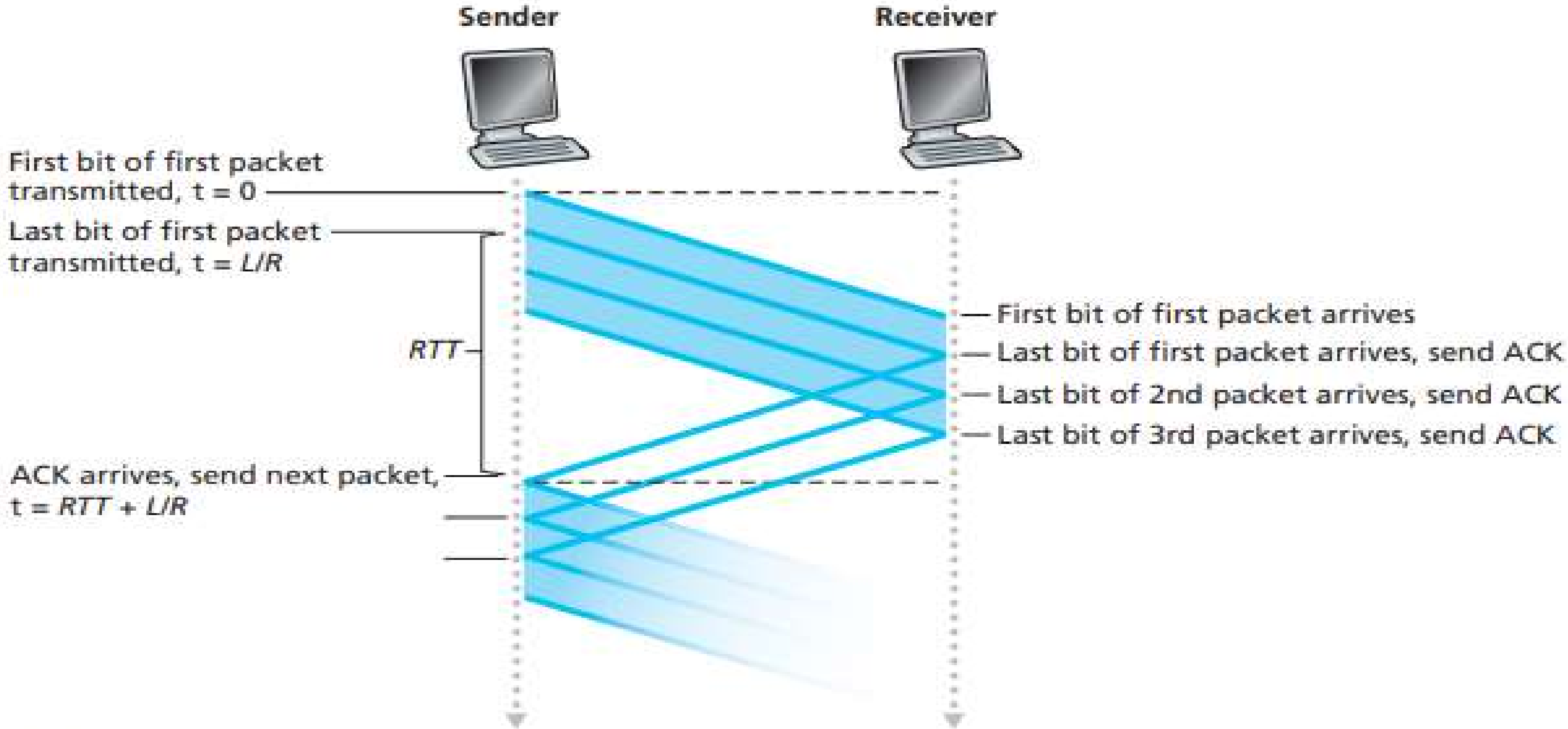
$$U_{sender} = \frac{L/R}{RTT + L/R} = \frac{.008}{30.008} = 0.00027$$

That is, the sender was busy only 2.7 hundredths of one percent of the time!

Principles of Reliable Data Transfer

Pipelined Reliable Data Transfer Protocols

- The solution to this particular performance problem is simple: Rather than operate in a stop-and-wait manner, the sender is allowed to send multiple packets without waiting for acknowledgment.



b. Pipelined operation

James F Kurose and Keith W Ross, “Computer Networking: A Top - Down Approach”, Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

Pipelined Reliable Data Transfer Protocols

- The range of sequence numbers must be increased, since each in-transit packet (not counting retransmissions) must have a unique sequence number and there may be multiple, in-transit, unacknowledged packets.
- The sender and receiver sides of the protocols may have to buffer more than one packet.
- Two basic approaches toward pipelined error recovery can be identified: Go-Back-N and selective repeat.

Principles of Reliable Data Transfer

Pipelined Reliable Data Transfer Protocols

Go-Back-N (GBN)

- In a Go-Back-N (GBN) protocol, the sender is allowed to transmit multiple packets (when available) without waiting for an acknowledgment, but is constrained to have no more than some maximum allowable number, N, of unacknowledged packets in the pipeline.

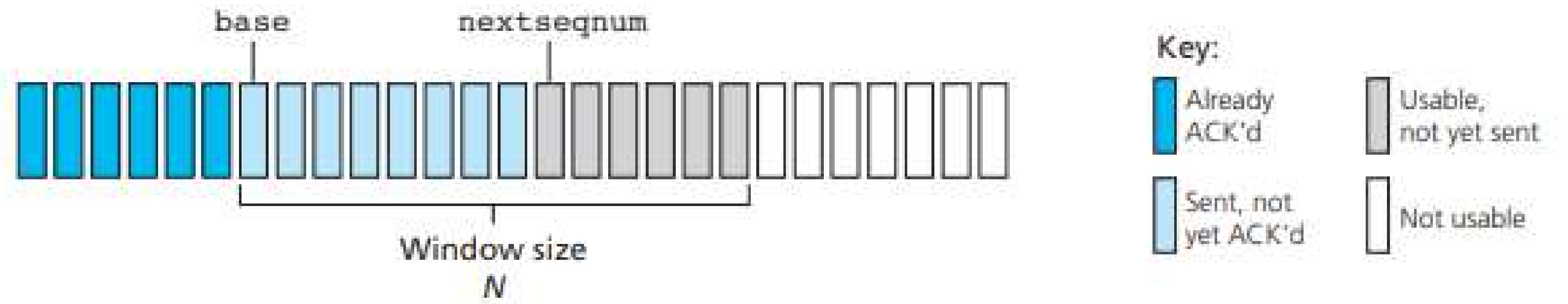


Figure 2.27 • Sender's view of sequence numbers in Go-Back-N

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

Pipelined Reliable Data Transfer Protocols

Go-Back-N (GBN)

- The range of permissible sequence numbers for transmitted but not yet acknowledged packets can be viewed as a window of size N over the range of sequence numbers.
- As the protocol operates, this window slides forward over the sequence number space.
- For this reason, N is often referred to as the window size and the GBN protocol itself as a sliding-window protocol.

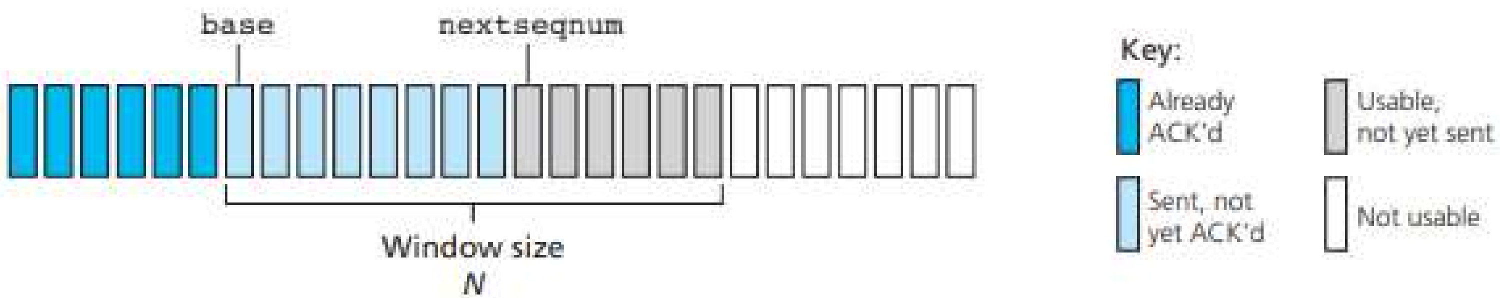


Figure 2.27 • Sender's view of sequence numbers in Go-Back-N

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

Pipelined Reliable Data Transfer Protocols

Go-Back-N (GBN)

ACK-based, NAK-free, GBN protocol

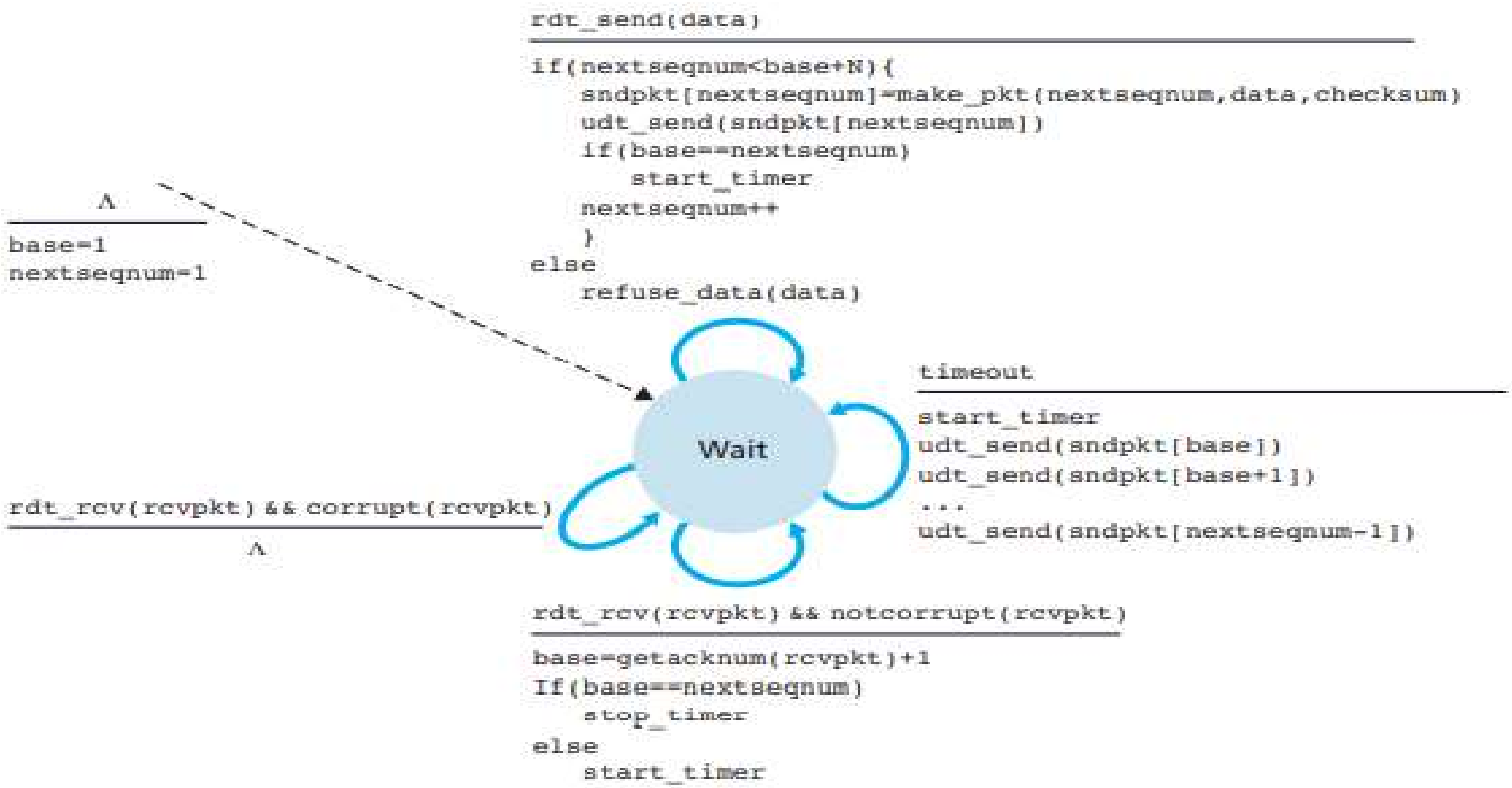


Figure 2.28 ♦ Extended FSM description of GBN sender

James F Kurose and Keith W Ross, “Computer Networking: A Top - Down Approach”, Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

Pipelined Reliable Data Transfer Protocols

Go-Back-N (GBN)

The GBN sender must respond to three types of events:

1. Invocation from above:

- When rdt_send() is called from above, the sender first checks to see if the window is full, that is, whether there are N outstanding, unacknowledged packets.
- If the window is not full, a packet is created and sent, and variables are appropriately updated.
- If the window is full, the sender simply returns the data back to the upper layer, an implicit indication that the window is full. The upper layer would presumably then have to try again later.
- In a real implementation, the sender would more likely have either buffered (but not immediately sent) this data, or would have a synchronization mechanism (for example, a semaphore or a flag) that would allow the upper layer to call rdt_send() only when the window is not full.

Principles of Reliable Data Transfer

Pipelined Reliable Data Transfer Protocols

Go-Back-N (GBN)

The GBN sender must respond to three types of events:

2. Receipt of an ACK:

- In GBN protocol, an acknowledgment for a packet with sequence number n will be taken to be a cumulative acknowledgment, indicating that all packets with a sequence number up to and including n have been correctly received at the receiver.

Principles of Reliable Data Transfer

Pipelined Reliable Data Transfer Protocols

Go-Back-N (GBN)

The GBN sender must respond to three types of events:

3. A timeout event:

- The protocol's name, "Go-Back-N," is derived from the sender's behavior in the presence of lost or overly delayed packets.
- As in the stop-and-wait protocol, a timer will again be used to recover from lost data or acknowledgment packets. If a timeout occurs, the sender resends all packets that have been previously sent but that have not yet been acknowledged.
- Our sender in Figure 2.28 uses only a single timer, which can be thought of as a timer for the oldest transmitted but not yet acknowledged packet.
- If an ACK is received but there are still additional transmitted but not yet acknowledged packets, the timer is restarted.
- If there are no outstanding, unacknowledged packets, the timer is stopped.

Principles of Reliable Data Transfer

Pipelined Reliable Data Transfer Protocols

Go-Back-N (GBN)

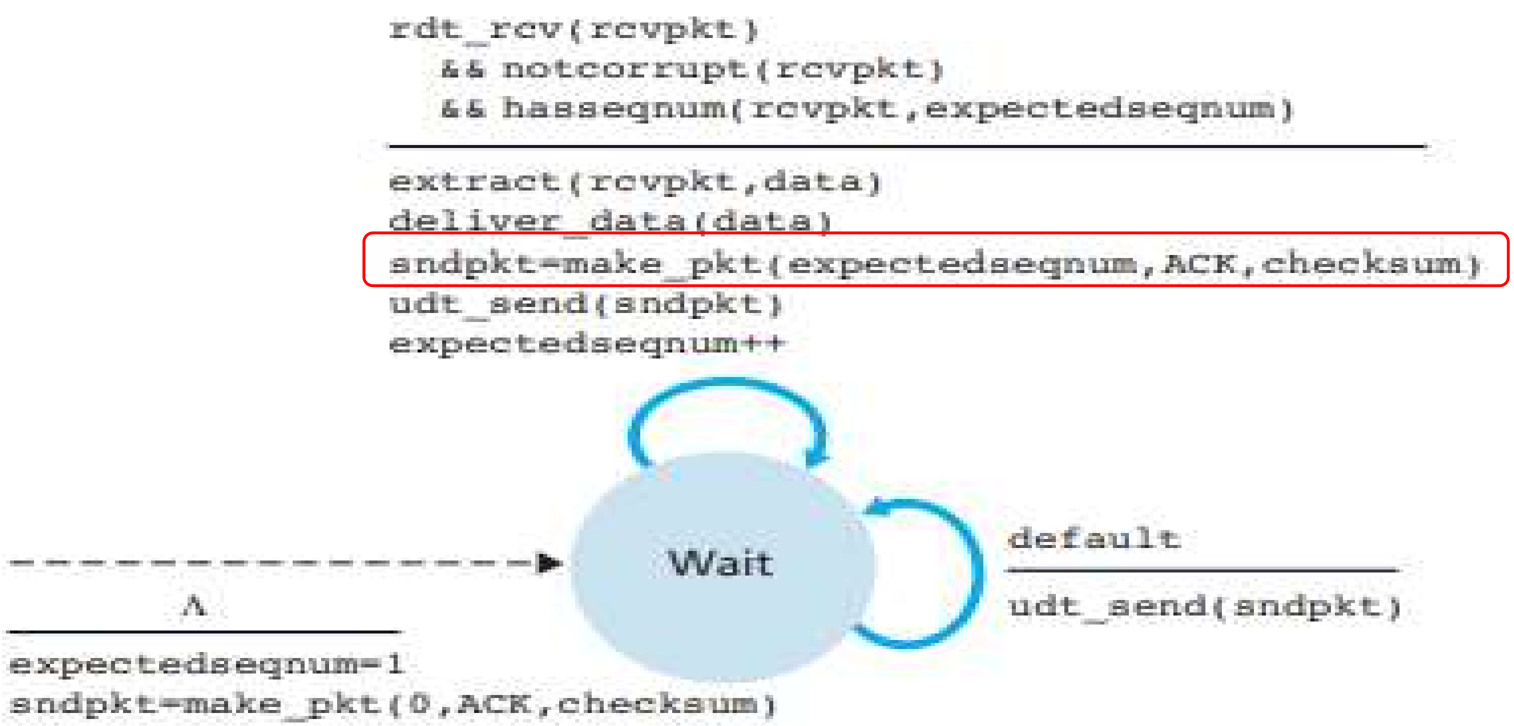


Figure 2.29 • Extended FSM description of GBN receiver

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

Pipelined Reliable Data Transfer Protocols

Go-Back-N (GBN)

The receiver's actions in GBN

- If a packet with sequence number n is received correctly and is in order (that is, the data last delivered to the upper layer came from a packet with sequence number n – 1), the receiver sends an ACK for packet n and delivers the data portion of the packet to the upper layer.
- In all other cases, the receiver discards the packet and resends an ACK for the most recently received in-order packet.
- Since packets are delivered one at a time to the upper layer, if packet k has been received and delivered, then all packets with a sequence number lower than k have also been delivered.
- Thus, the use of cumulative acknowledgments is a natural choice for GBN.

Principles of Reliable Data Transfer

Pipelined Reliable Data Transfer Protocols

Go-Back-N (GBN)

The receiver's actions in GBN

- In our GBN protocol, the receiver discards out-of-order packets.
- The advantage of this approach is the simplicity of receiver buffering—the receiver need not buffer any out-of-order packets.
- While the sender must maintain the upper and lower bounds of its window and the position of *nextseqnum* within this window, the only piece of information the receiver need maintain is the sequence number of the next in-order packet. This value is held in the variable *expectedseqnum*, shown in the receiver FSM in Figure 2.29.
- Of course, the disadvantage of throwing away a correctly received packet is that the subsequent retransmission of that packet might be lost or garbled and thus even more retransmissions would be required.

Principles of Reliable Data Transfer

Pipelined Reliable Data Transfer Protocols

Go-Back-N (GBN)

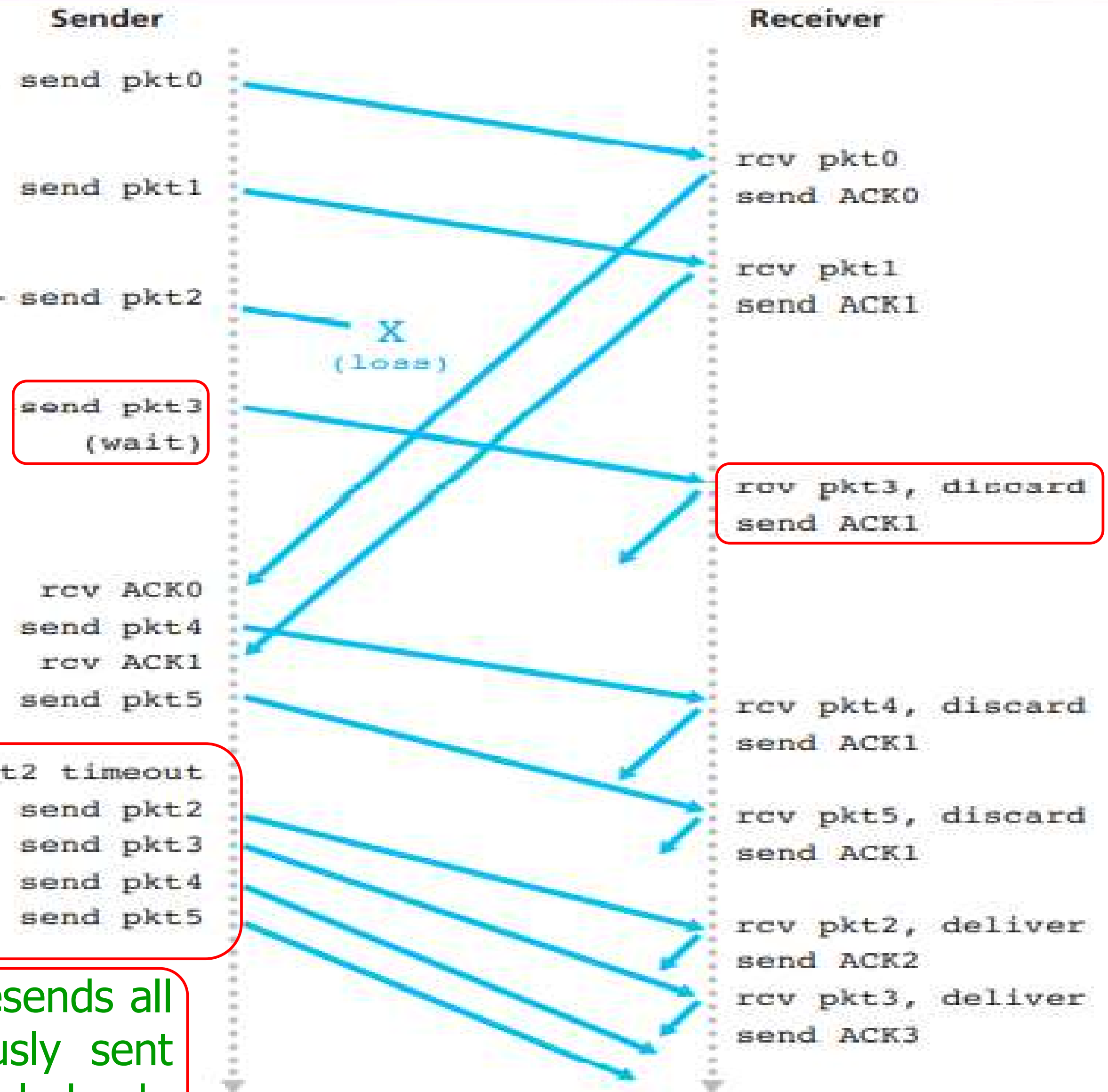
Window size N=4

- Figure 2.30 shows the operation of the GBN protocol for the case of a window size of four packets.

0
1
2
3

2
3
4
5

Go-Back-N
N=4



If a timeout occurs, the sender resends all packets that have been previously sent but that have not yet been acknowledged

Figure 2.30 • Go-Back-N in operation

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

Go-Back-N (GBN)

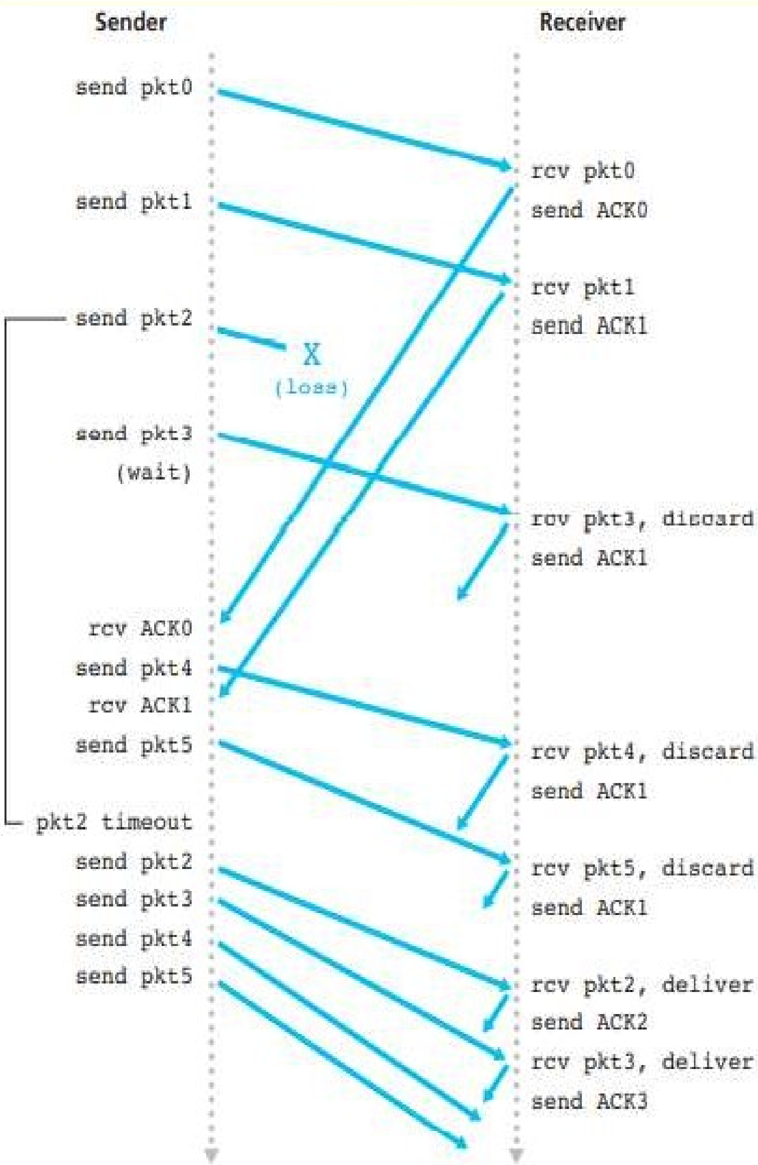


Figure 2.30 ♦ Go-Back-N in operation

- Because of the window size limitation, the sender sends packets 0 through 3 but then must wait for one or more of these packets to be acknowledged before proceeding.
- As each successive ACK (for example, ACK0 and ACK1) is received, the window slides forward and the sender can transmit one new packet (pkt4 and pkt5, respectively).
- On the receiver side, packet 2 is lost and thus packets 3, 4, and 5 are found to be out of order and are discarded.

James F Kurose and Keith W Ross, “Computer Networking: A Top - Down Approach”, Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

Go-Back-N (GBN)

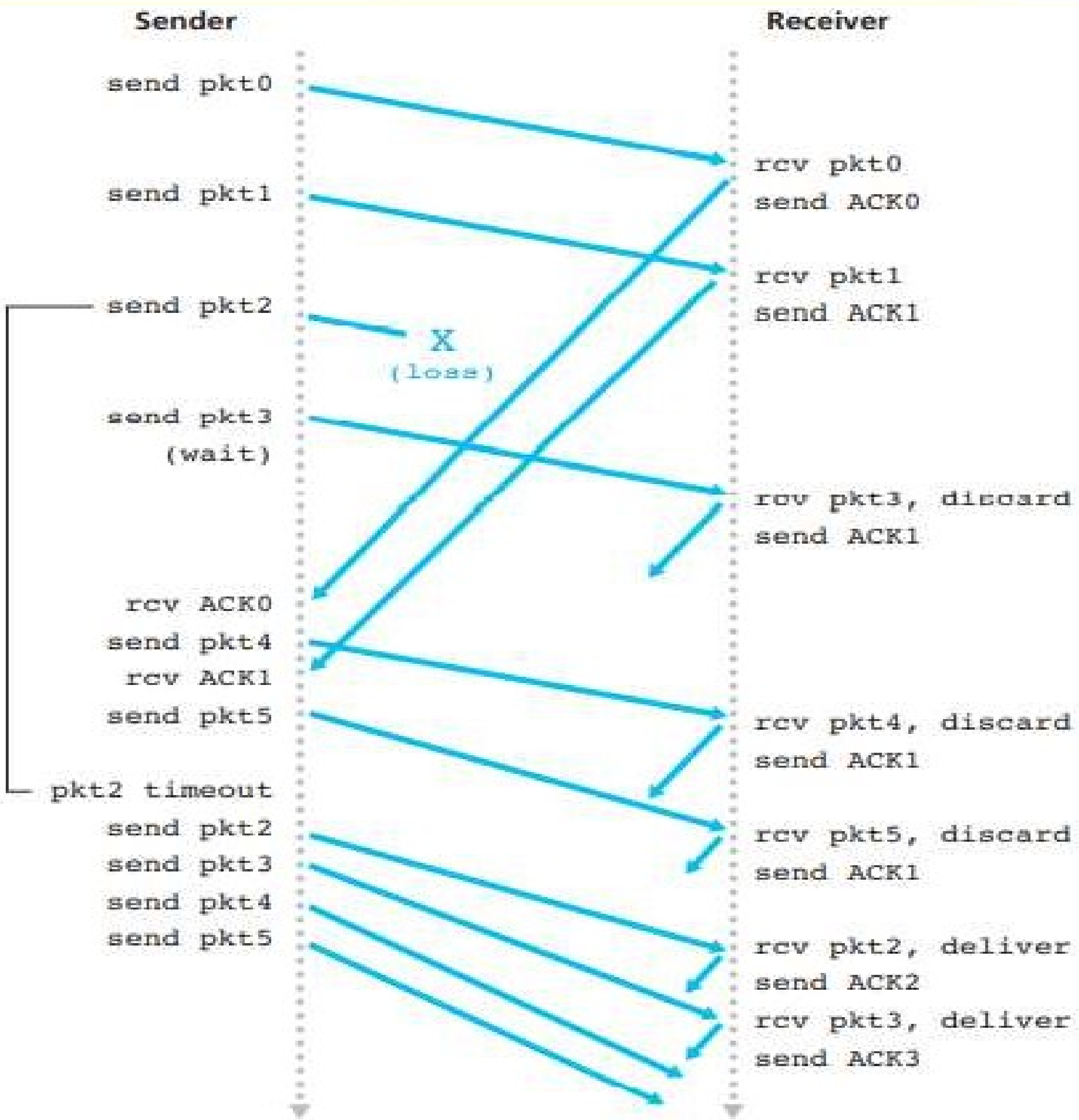


Figure 2.30 ♦ Go-Back-N in operation

- GBN protocol incorporates almost all of the techniques for the reliable data transfer components of TCP.
- These techniques include the use of sequence numbers, cumulative acknowledgments, checksums, and a timeout/retransmit operation

James F Kurose and Keith W Ross, “Computer Networking: A Top - Down Approach”, Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

Selective Repeat (SR)

- As the name suggests, selective-repeat protocols avoid unnecessary retransmissions by having the sender retransmit only those packets that it suspects were received in error (that is, were lost or corrupted) at the receiver.
- This individual retransmission will require that the receiver individually acknowledge correctly received packets.

Principles of Reliable Data Transfer

Selective Repeat (SR)

- The SR receiver will acknowledge a correctly received packet whether or not it is in order.
- Out-of-order packets are buffered until any missing packets (that is, packets with lower sequence numbers) are received, at which point a batch of packets can be delivered in order to the upper layer.

Principles of Reliable Data Transfer

Selective Repeat (SR)

Window size N=4

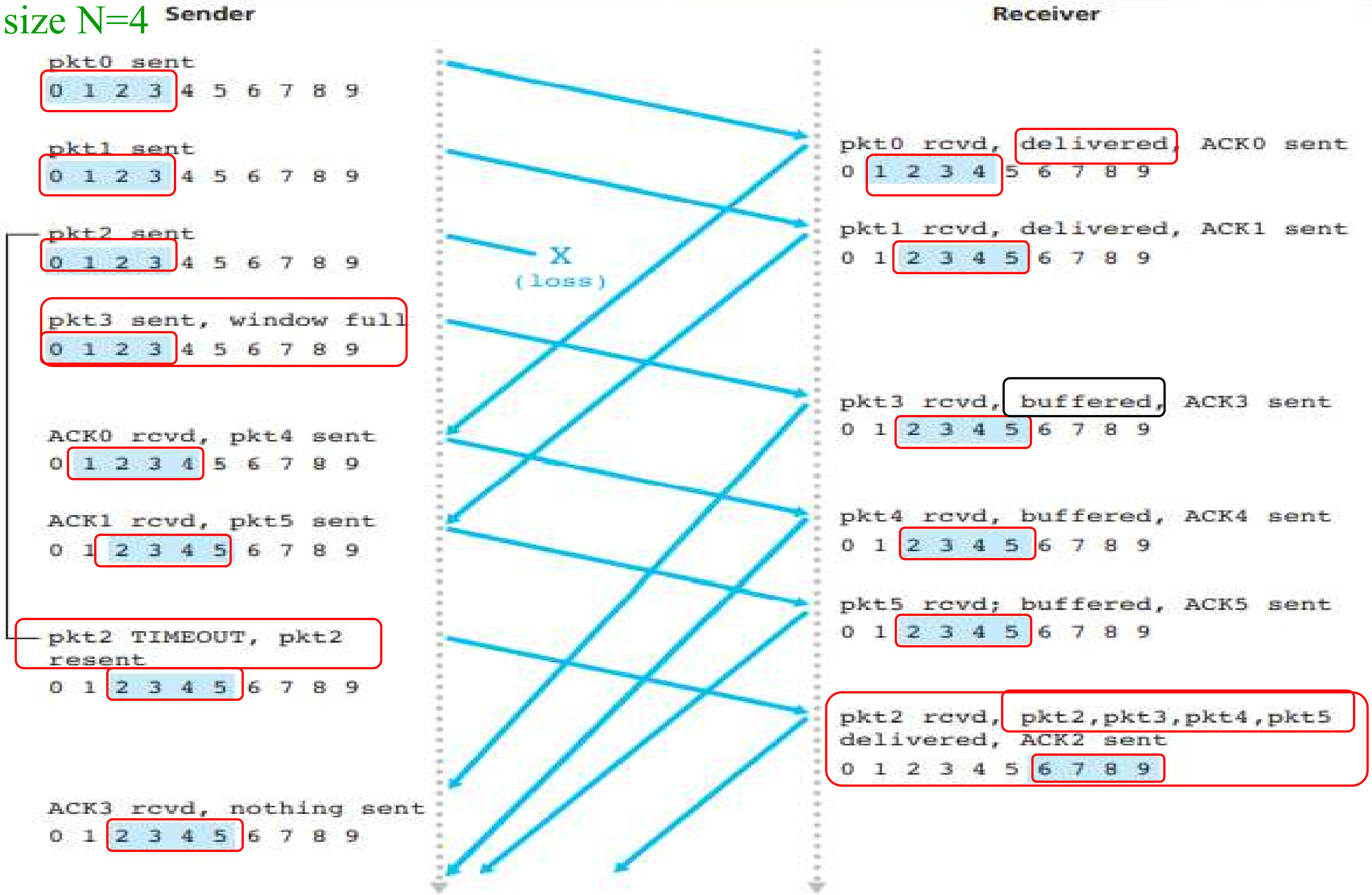


Figure 2.31 ♦ SR operation

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

Selective Repeat (SR)

- The SR receiver will acknowledge a correctly received packet whether or not it is in order.
- Out-of-order packets are buffered until any missing packets (that is, packets with lower sequence numbers) are received, at which point a batch of packets(pkt 2, pkt3, pkt4, pkt5) can be delivered in order to the upper layer.

Principles of Reliable Data Transfer

Selective Repeat (SR)

1. *Data received from above.* When data is received from above, the SR sender checks the next available sequence number for the packet. If the sequence number is within the sender's window, the data is packetized and sent; otherwise it is either buffered or returned to the upper layer for later transmission, as in GBN.
2. *Timeout.* Timers are again used to protect against lost packets. However, each packet must now have its own logical timer, since only a single packet will be transmitted on timeout. A single hardware timer can be used to mimic the operation of multiple logical timers [Varghese 1997].
3. *ACK received.* If an ACK is received, the SR sender marks that packet as having been received, provided it is in the window. If the packet's sequence number is equal to `send_base`, the window base is moved forward to the unacknowledged packet with the smallest sequence number. If the window moves and there are untransmitted packets with sequence numbers that now fall within the window, these packets are transmitted.

Figure 3.32a ♦ SR sender events and actions

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

Selective Repeat (SR)

1. *Packet with sequence number in* `[rcv_base, rcv_base+N-1]` *is correctly received.* In this case, the received packet falls within the receiver's window and a selective ACK packet is returned to the sender. If the packet was not previously received, it is buffered. If this packet has a sequence number equal to the base of the receive window (`rcv_base` in Figure 3.22), then this packet, and any previously buffered and consecutively numbered (beginning with `rcv_base`) packets are delivered to the upper layer. The receive window is then moved forward by the number of packets delivered to the upper layer. As an example, consider Figure 3.26. When a packet with a sequence number of `rcv_base=2` is received, it and packets 3, 4, and 5 can be delivered to the upper layer.
2. *Packet with sequence number in* `[rcv_base-N, rcv_base-1]` *is correctly received.* In this case, an ACK must be generated, even though this is a packet that the receiver has previously acknowledged.
3. *Otherwise.* Ignore the packet.

Figure 3.32b• SR receiver events and actions

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

Selective Repeat (SR)

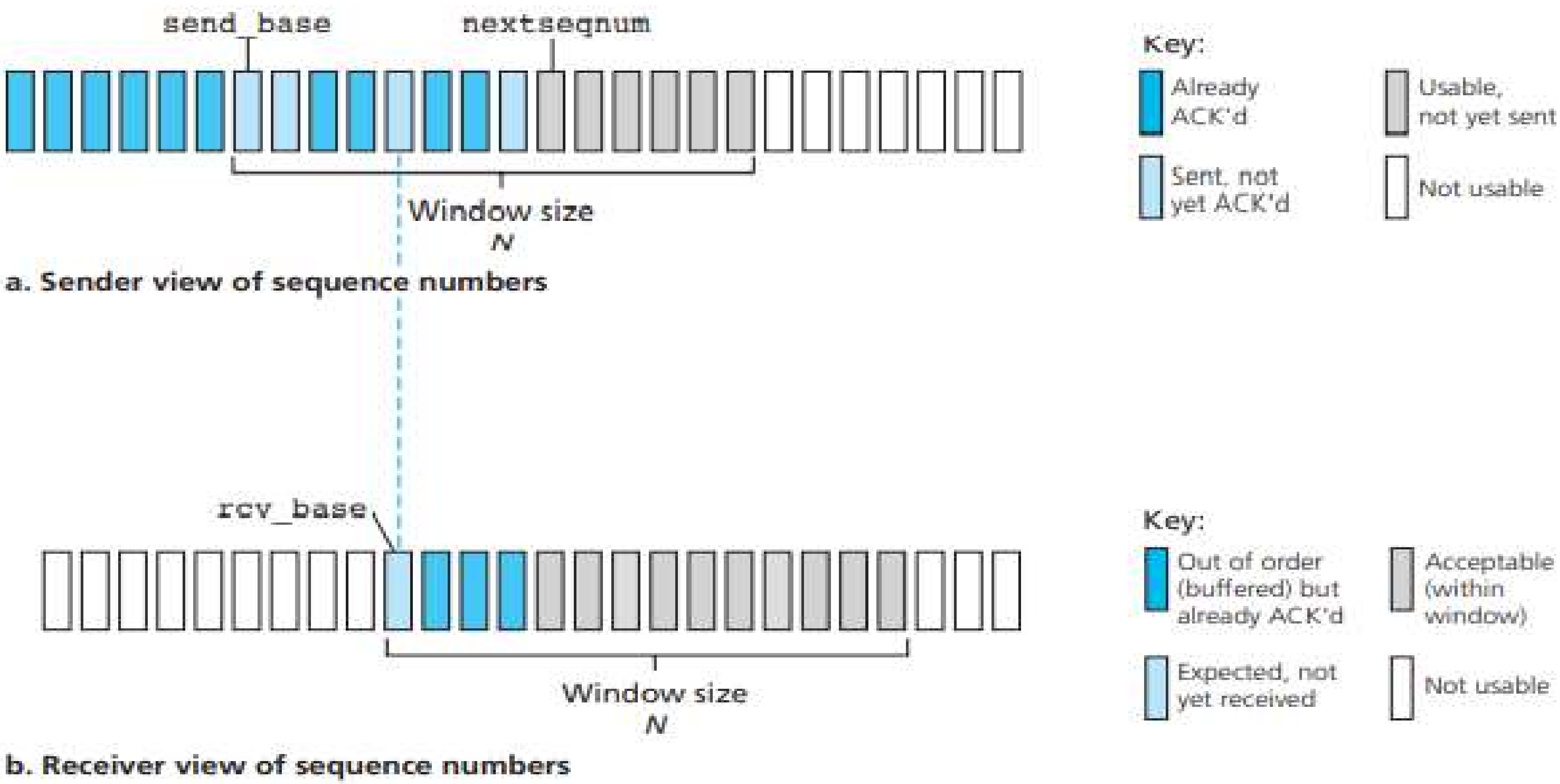


Figure 2.32 • Selective-repeat (SR) sender and receiver views of sequence-number space

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)

Principles of Reliable Data Transfer

Mechanism	Use, Comments
Checksum	Used to detect bit errors in a transmitted packet.
Timer	Used to timeout/retransmit a packet, possibly because the packet (or its ACK) was lost within the channel. Because timeouts can occur when a packet is delayed but not lost (premature timeout), or when a packet has been received by the receiver but the receiver-to-sender ACK has been lost, duplicate copies of a packet may be received by a receiver.
Sequence number	Used for sequential numbering of packets of data flowing from sender to receiver. Gaps in the sequence numbers of received packets allow the receiver to detect a lost packet. Packets with duplicate sequence numbers allow the receiver to detect duplicate copies of a packet.
Acknowledgment	Used by the receiver to tell the sender that a packet or set of packets has been received correctly. Acknowledgments will typically carry the sequence number of the packet or packets being acknowledged. Acknowledgments may be individual or cumulative, depending on the protocol.
Negative acknowledgment	Used by the receiver to tell the sender that a packet has not been received correctly. Negative acknowledgments will typically carry the sequence number of the packet that was not received correctly.
Window, pipelining	The sender may be restricted to sending only packets with sequence numbers that fall within a given range. By allowing multiple packets to be transmitted but not yet acknowledged, sender utilization can be increased over a stop-and-wait mode of operation. We'll see shortly that the window size may be set on the basis of the receiver's ability to receive and buffer messages, or the level of congestion in the network, or both.

Table 3.1 ◆ Summary of reliable data transfer mechanisms and their use

James F Kurose and Keith W Ross, "Computer Networking: A Top - Down Approach", Pearson Education; 6 th Edition (2017)