IP Transport Layer
Ch. 25–6

Transport Protocols Over IP

The IP layer provides an unreliable service over a virtual network.

It delivers virtual packets to virtual addresses.

The transport layer provides usable services over that virtual network.

Transport layer services see only IP below them. They don't know or care what hardware is used.

TCP and UDP are transport protocols

User Datagram Protocol
UDP

Allows applications to send messages.

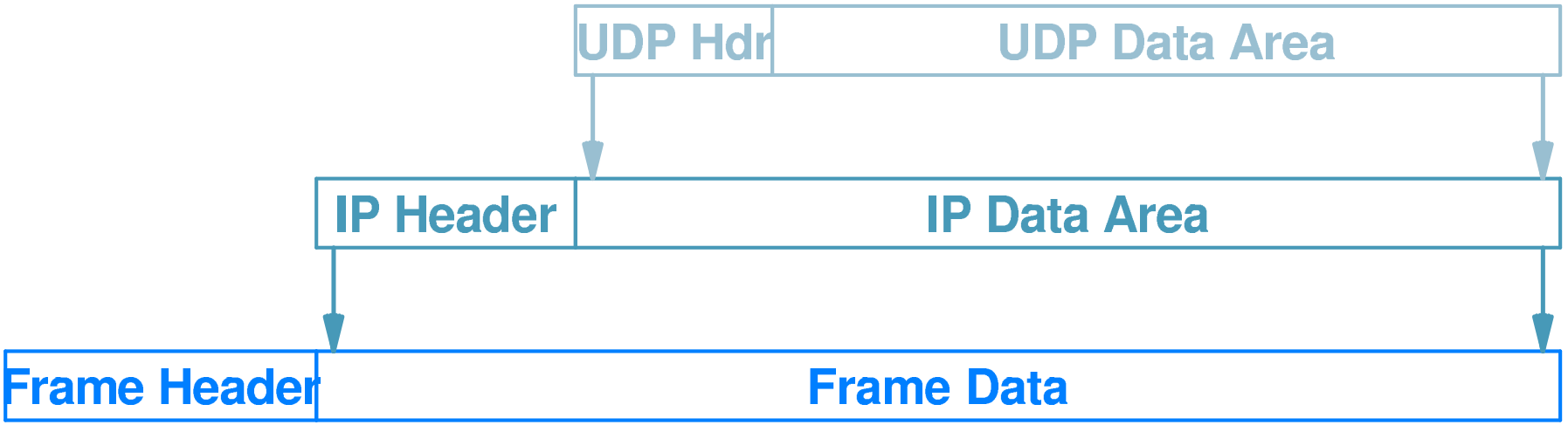
Messages are sent to a *port number* on a particular host.

The port number specifies which program should receive the message.
Independent of any OS identifiers.

Delivery is unreliable as IP packets.

Best effort: packets may be
Lost Duplicated Reordered

UDP Message Encapsulation

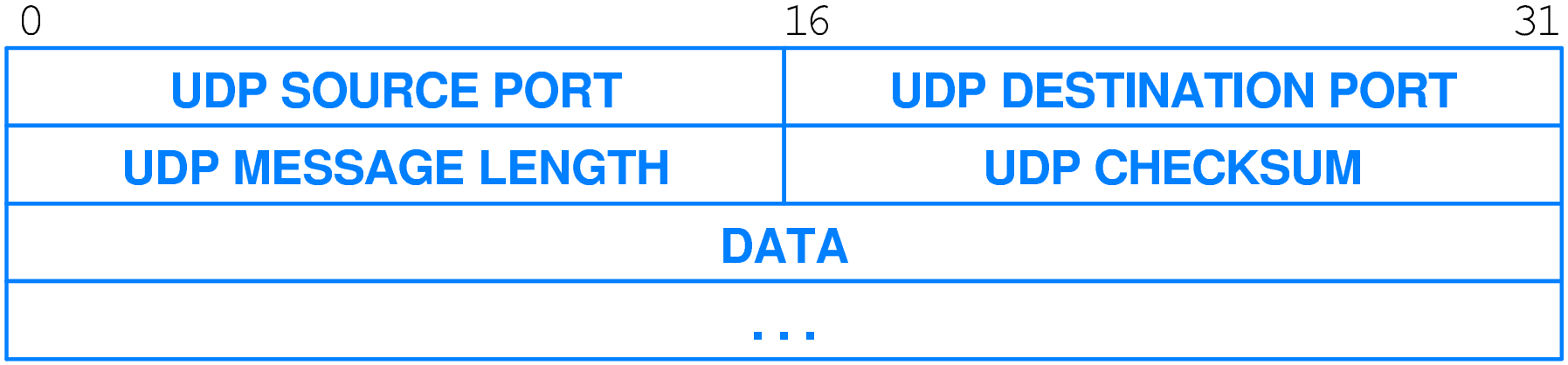


UDP messages ride in IP datagrams.

TYPE field in IP header set to 17 for UDP.

Type of 0800 for IP set in the Frame Header.

UDP Header



Checksum may be sent as zero, indicating no checksum.

Limits

UDP messages are limited to the size of an IP datagram.

Large UDP messages may fragment.
Inefficient

Flexibility

UDP messages can use unicast, multicast, or broadcast addresses.
Not possible with connection-oriented services like TCP.

Hosts can relate one-to-many or many-to-many.

Efficiency of multicast delivery depends greatly on the IP implementation, which may also depend on the underlying hardware.

UDP Checksum

The use of the checksum is optional.

If used, it essentially turns corrupted messages into lost messages.

UDP computes its checksum over the UDP packet plus:
the source, destination and prototype fields from the IP header,
the size of the UDP datagram

Who Uses UDP?

Not most traditional applications: SMTP, POP, FTP, HTTP.

DNS can use UDP or TCP
UDP is used for most routine things.

Multimedia streams and VOIP use UDP.
UDP was originally created to support voice teleconferencing.

Some VPNs use it.

Transmission Control Protocol

TCP

TCP builds a reliable transport mechanism using IP datagrams.

Must create a reliable service founded on an unreliable one.

Service Provided

Connection-oriented

Point-to-point
Exactly two programs communicate.

Complete reliability
No loss, duplication, or reordering.

Full duplex
Each connection allows data to travel in each direction.

Service Provided

Stream interface

All data is merged to a single stream of bytes.

Message boundaries are not preserved.

Reliable connection startup

Both ends must agree.

Old packets cannot interfere.

Graceful connection shutdown

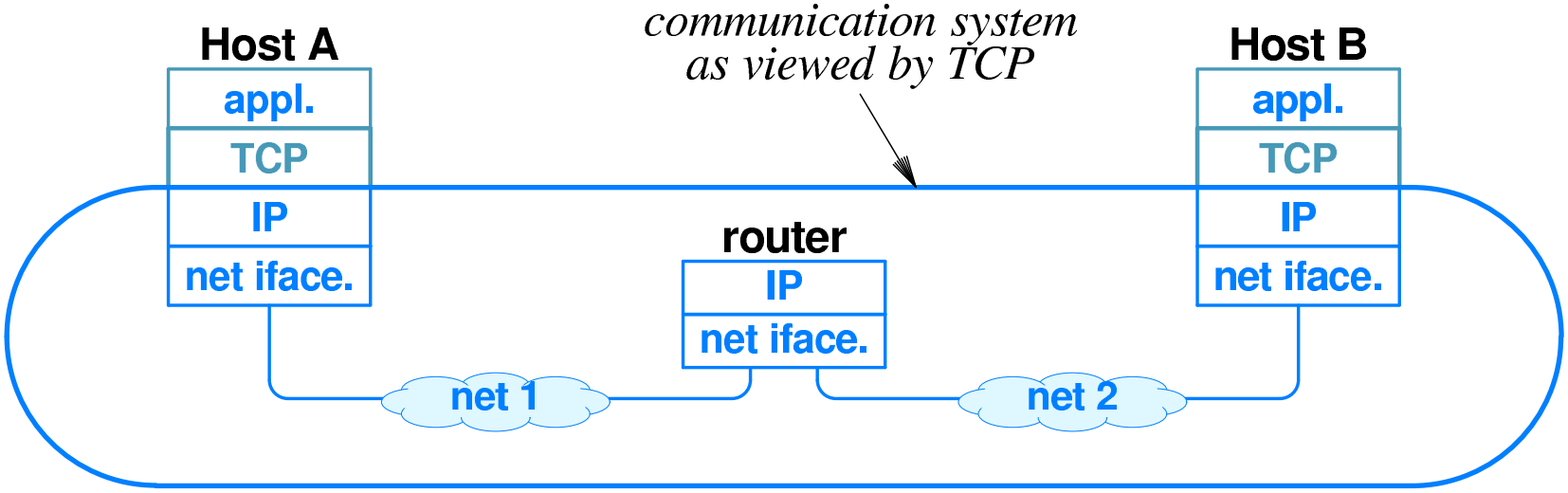
Both ends must agree to terminate.

TCP Connections

A direct, point-to-point link between applications on different computers.

Virtual connection: Created using software at the endpoints. Routers between the endpoints don't know about it.

Routers and TCP



Routers need only handle IP datagrams.

They do not implement TCP or UDP.
Or any other transport.

Reliable Transport

Goal: Communications should be reliable.
You send it, it gets there.

Problems

Messages may be lost, duplicated, mis-ordered, or corrupted.
Corrupted = lost

One of the endpoints may be rebooted.

Congestion: The senders together may send too much data, congesting the routers.
Routers drop packets where there are too many to handle.

Overrun: Sender transmits too fast for the receiver to consume.

Sequence Numbers

Each message is assigned a sequence number on transmission.
Receiver checks the sequence.

Detects out-of-order and duplicate reception.

Sequence Numbers, Cont.

The receiver acknowledges data it receives.
It sends return messages informing the sender what has gotten through.

Sender keeps track of what has and has not been acknowledged.

If the acknowledgment is not received in a reasonable amount of time, re-transmit.

Detects lost packets.

Sessions

Transmission may be greatly delayed.

A packet from an old conversation may arrive with a current sequence number.

Assign a new session number when starting a connection.
Discard packets with old session numbers.

or

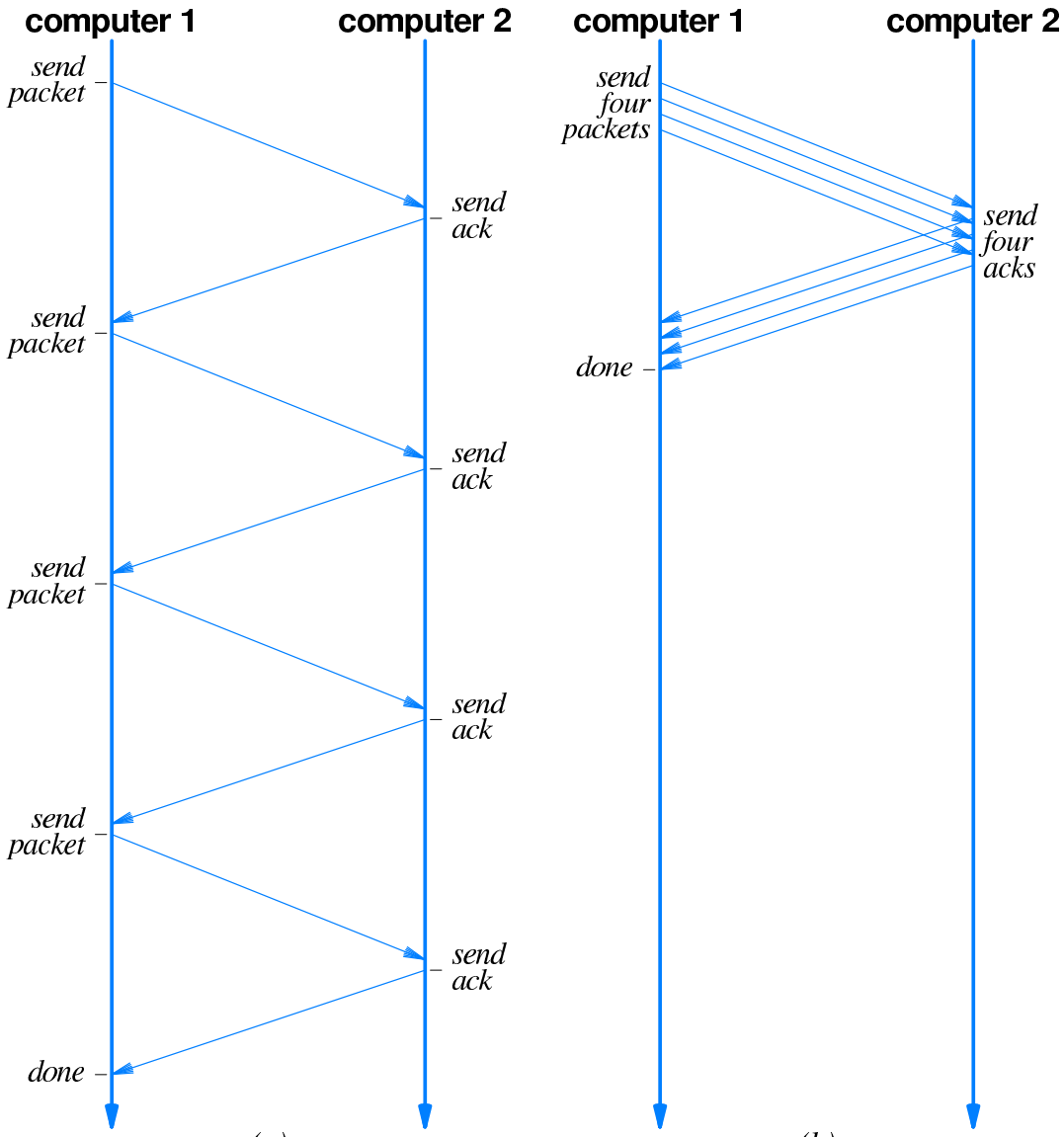
Use large sequence numbers and don't restart new connections at zero.

When To Send The Next Message

Wait to send the next message until the last one is acknowledged.
"*Stop-and-go*"

Go ahead and send another message while you're waiting for the acknowledgement.
This is faster.

Stop And Go Is Slow



Faster Is Not Always Better

Could simply send data as fast as the NIC will take it.

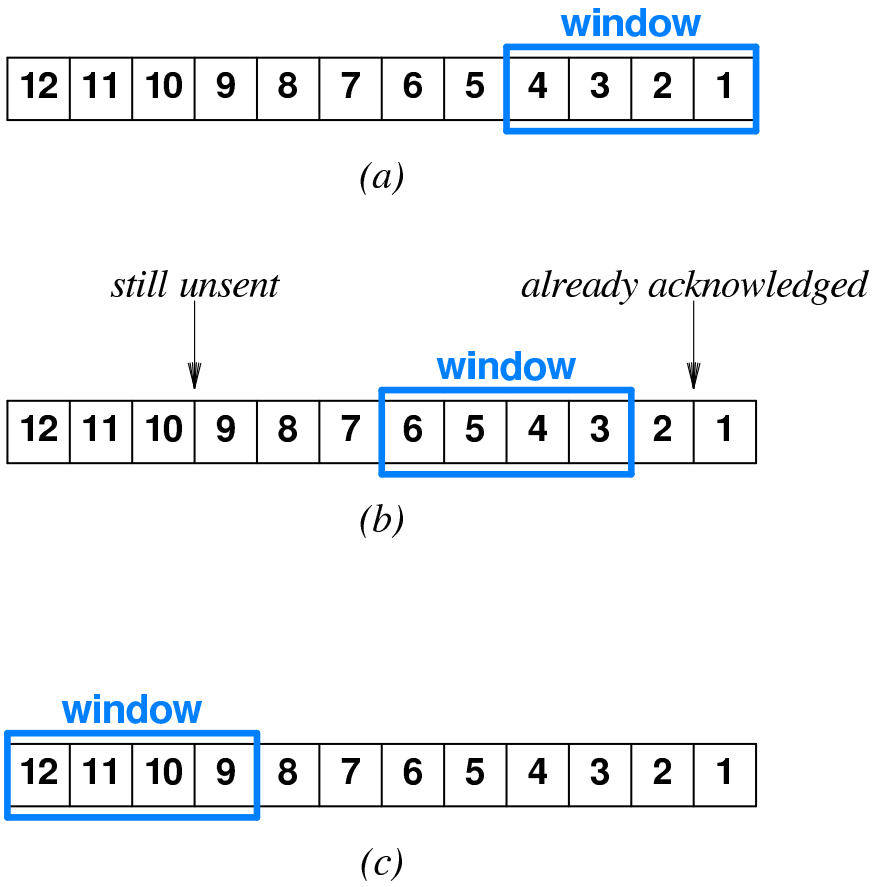
May congest the network
Especially if everyone does it.

May overwhelm the receiver if it is a smaller machine.

Place a limit on the number of unacknowledged packets.
This limits the sending rate.

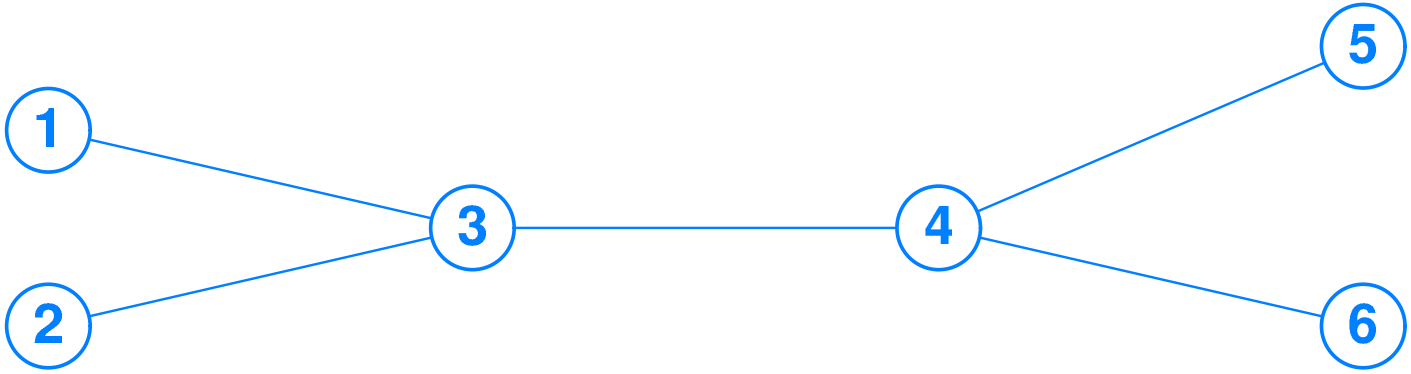
Sliding-window protocol.

Sliding-Window Protocols



The window should be as large as can be accommodated by the network and the receiver.

Congestion



All links of the same capacity.

Link from 3 to 4 can easily become congested.

When a switch runs out of storage, it simply drops new packets.

Congestion may occur far away from the sender, regardless of the capacity of its own link.

Detecting Congestion

- Have switches inform senders.

- Use packet loss as a measure of congestion.

When congestion is detected, the sender must slow down.
TCP reduces the window size.

Numbering and Windows In TCP

TCP numbers each byte in the transmission, not each segment.
Segments actually carry the sequence number of their first byte.

Acknowledgement numbers are for all the bytes before this one.

Send window is measured in bytes.
Can represent any number of segments limited by total bytes.

Congestion is detected by packet loss.

Adaptive Retransmission

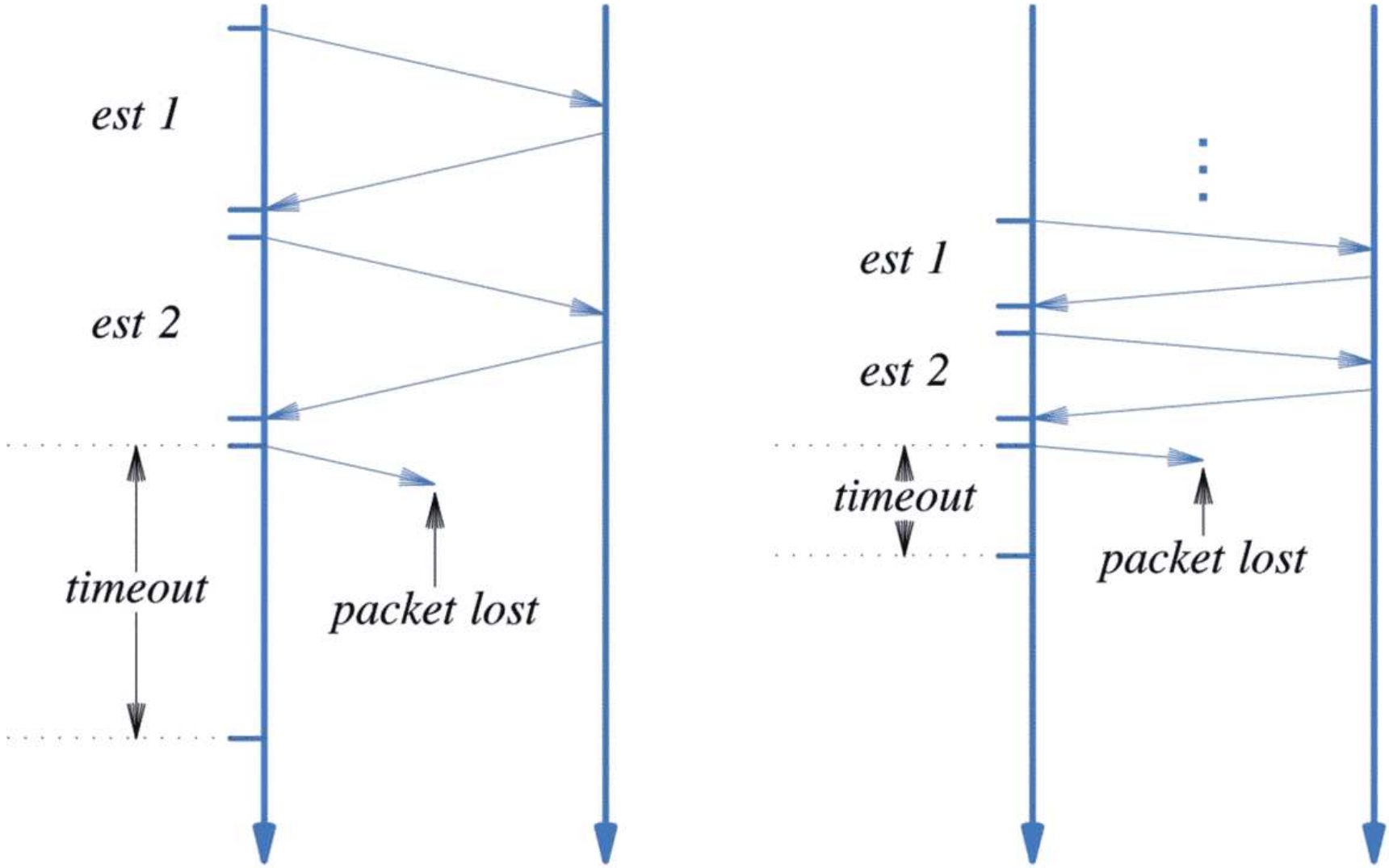
Transmission times vary
An Ethernet across the room or a satellite link across the ocean.

If the timeout is too short, unnecessary retransmission.

If the timeout is too long, will waste time waiting before needed retransmission.

Sender keeps a running average round-trip-time.
Timeout based on RTT measurement.

Adaptive Retransmission



Round-Trip Times

The sender can measure time from sending to ack receipt.
Original approach.

RFC 1323 defines a time-stamp option.
The sender adds its time to the data segment.
The stamp is returned on the ack.

Simplifies sender and provides better measurement.

Congestion Control

The sending window increases on successful acknowledgement, and decreases on packet loss.

The idea is settle at the highest transmission rate which doesn't cause congestion.

Increases are fast at first, then slower.

Decreases are fast.

Congestion is managed in the aggregate when all senders behave.

When an acknowledgment is received

If the window is "small," the window increases by the maximum packet size for each acknowledgment.
Effectively doubles the window each time a window's-worth is acknowledged.

If the window is "large," the window increases by the maximum packet size for each round trip time.

When a packet is lost

The window is reduced to one max packet size

The threshold between "small" and "large" becomes half the amount of unacknowledged data.

Flow Control

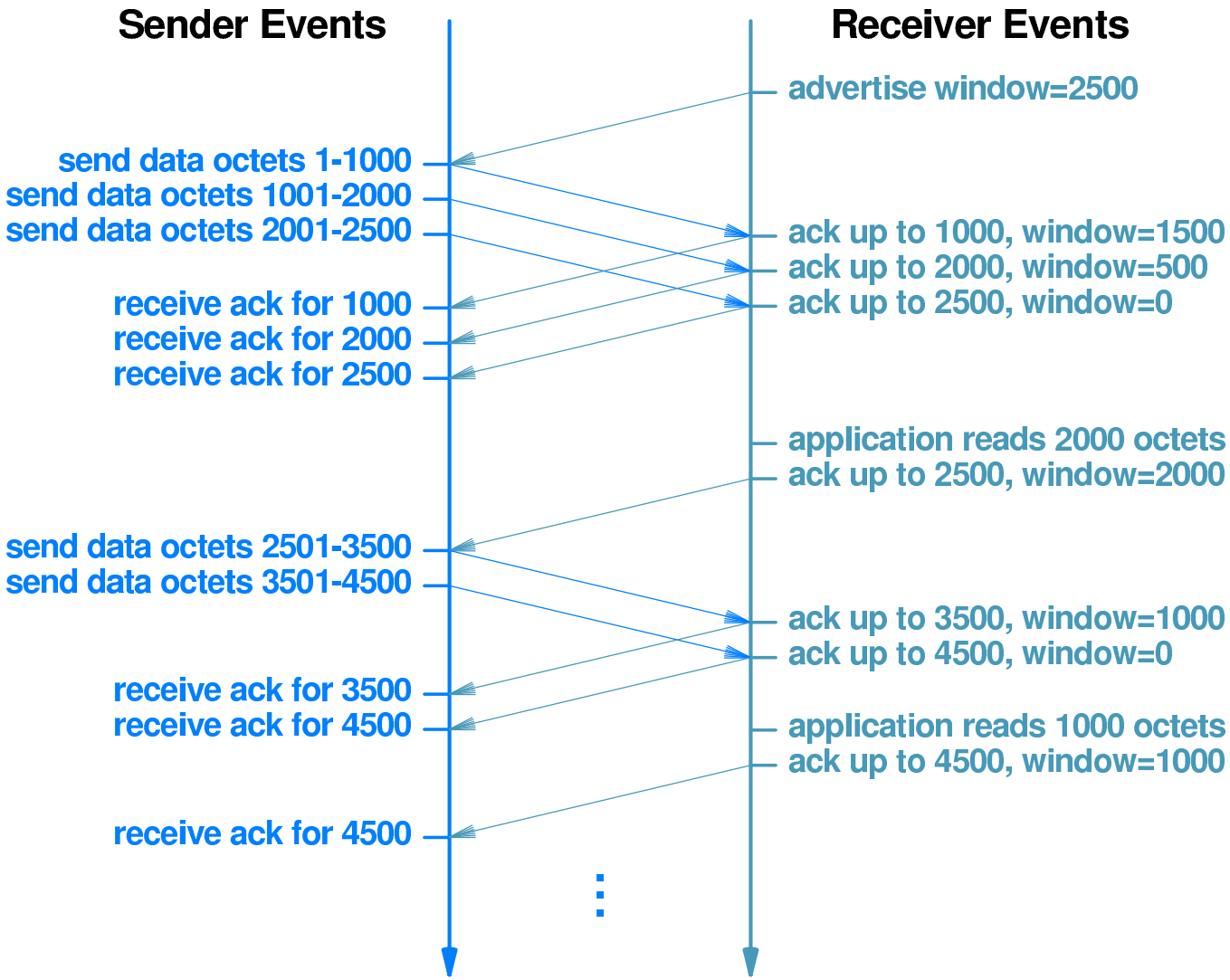
Preventing the sender from overrunning the receiver.

Each end of the communication allocates a certain amount of space to store received data.

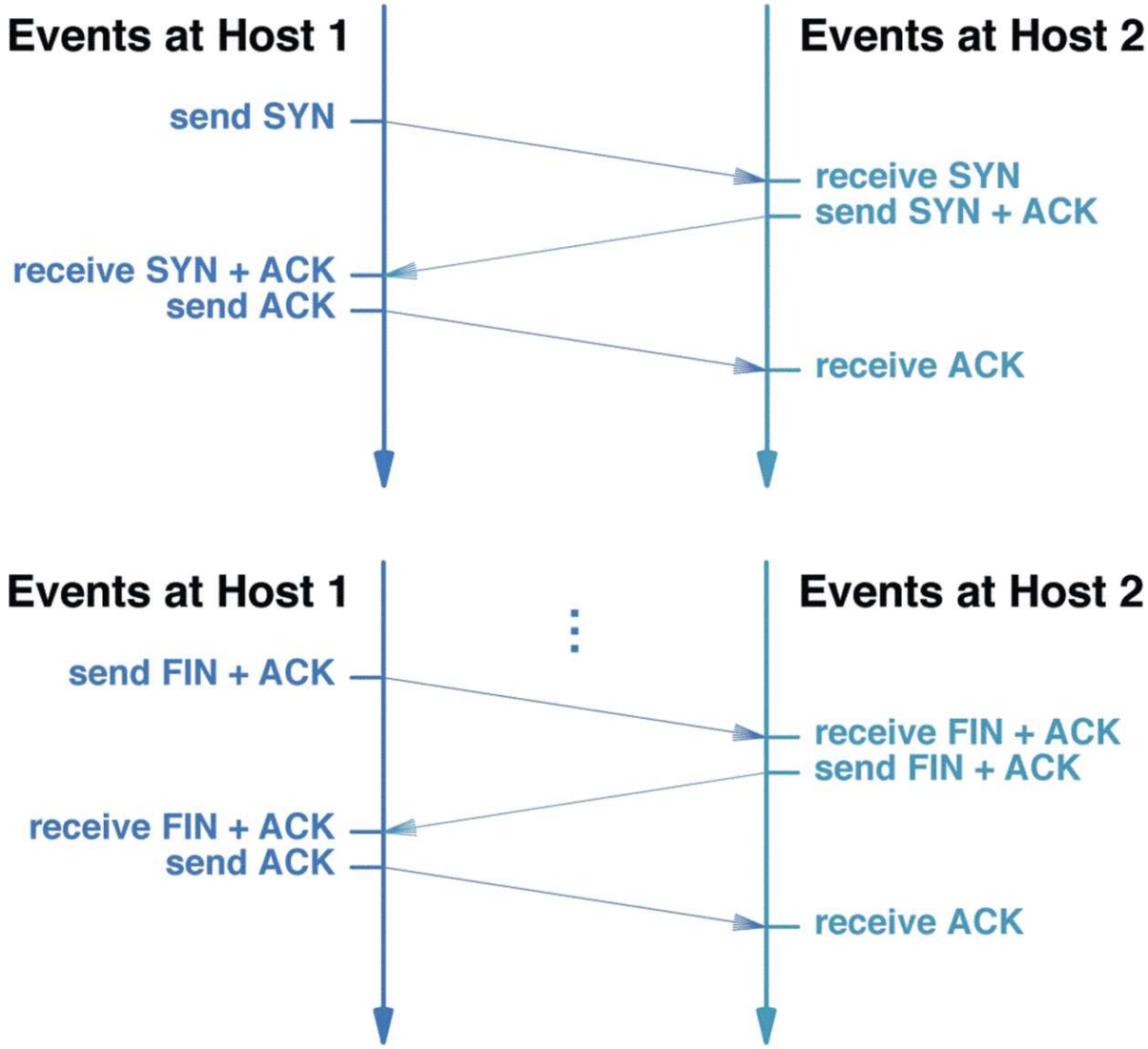
Each packet contains an *advertisement* which tells how much free reception space is available.

The sending window is limited to the most recently-received advertisement.

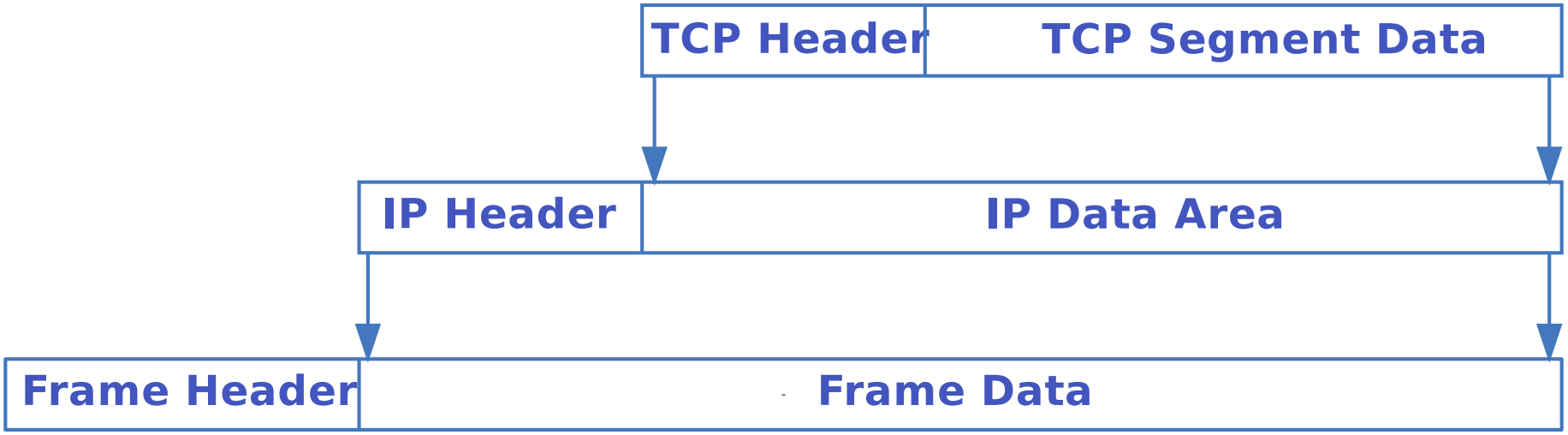
Windows and Acks



Opening and Closing A Connection



TCP Message Encapsulation

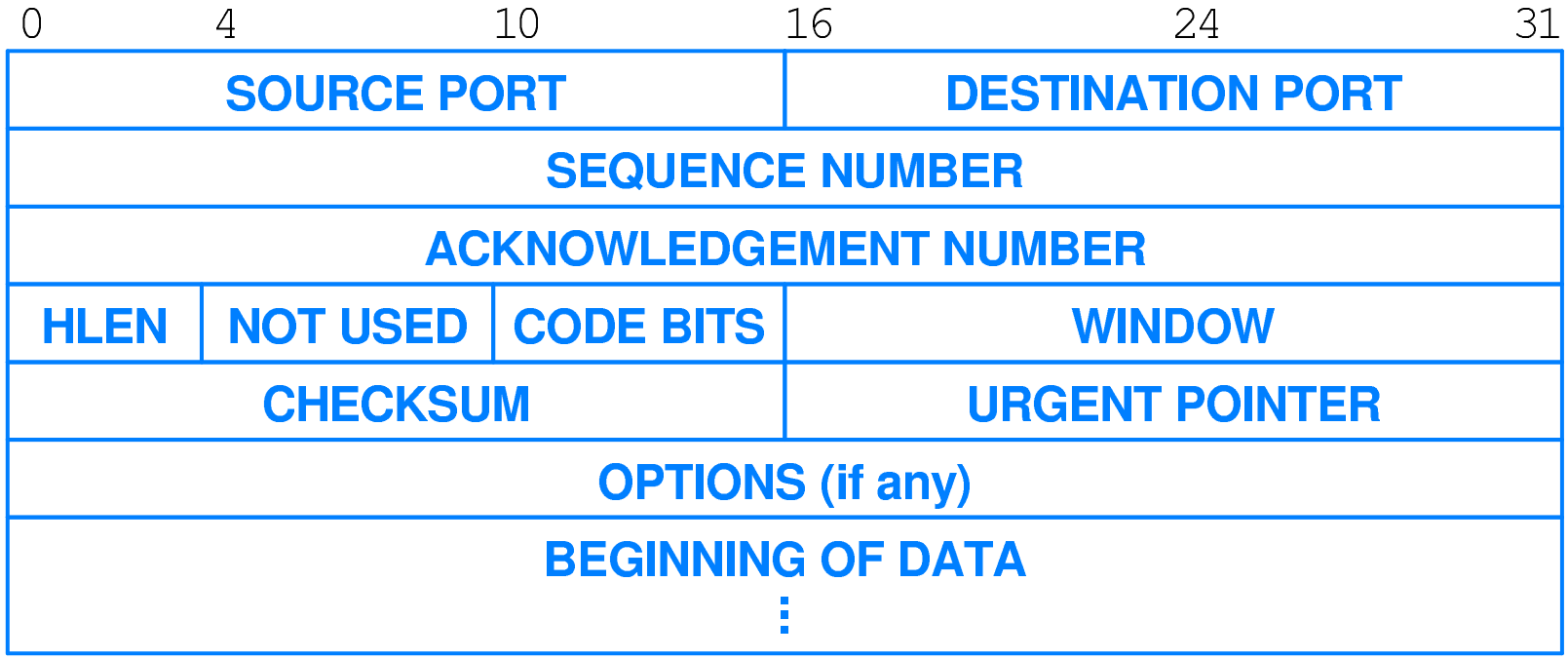


TCP segments ride in IP datagrams.

TYPE field in IP header set to 6 for TCP.

Type of 0800 for IP set in the Frame Header.

TCP Headers


0
4
10
16
24
31
SOURCE PORT
DESTINATION PORT
SEQUENCE NUMBER
ACKNOWLEDGEMENT NUMBER
HLEN
NOT USED
CODE BITS
WINDOW
CHECKSUM
URGENT POINTER
OPTIONS (if any)
BEGINNING OF DATA

TCP Flags

Called CODE BITS in the text figure; everyone else says flags.

SYN and FIN we've discussed.

ACK	The acknowledgment number is valid.
RST	Reset the connection. Indicates an error or crash.
PSH	Push. Deliver data ASAP. Usually ignored.
URG	Urgent. Deliver data immediately, out of order. The URGENT POINTER indicates the end of the urgent data.

Header Sequence Numbers

The sequence number is the number of first byte in this segment.

If this segment has no data, the acknowledgement number is for the next byte to be sent.

If the ACK bit is true, the acknowledgement number means "I have received all bytes before this one."
The number of the first missing byte, or the next one expected.

Selective Acknowledgement

The ack field provides *cumulative acknowledgement.*

RFC 2018 allows for *selective acknowledgement* (SACK).

An option can be included which acknowledges a list of ranges.

Gives the sender a clearer idea of what has gotten through.

Loss = Congestion

TCP assumes a loss is due to congestion.

Reasonable assumption on a wired network.

Not so much on wireless.

Explicit Congestion Notification: ECN

Recent proposed standard (RFC 3168) allows TCP to request routers set a congested bit on passing traffic.

This bit is sent back to the sender on the acknowledgement.

When an ack with "I'm congested" comes back, TCP adjusts the window as for a loss.

Slows the sender down before packets must be discarded.

Why TCP Is Sometimes The Wrong Answer

Having the latest data may matter more that avoiding all loss.

If an on-line game loses a packet giving the mouse location, it wants to send the current location, not resend the old one.

When you lose a packet of real-time audio, you want to send the next one, not resend the old one.

When packets must be delivered in order, a loss delays everything behind it.

Many audio, game and other real-time aps use UDP instead of TCP to avoid these problems.

Why UDP May Become A Problem

UDP has no congestion control.
No sending window.
No slow-down response to congestion.

If Internet traffic contains many long-lived UDP interactions, much sending will be unresponsive to congestion.

The Internet could experience congestion collapse.

Datagram Congestion Control Protocol (DCCP)

"Standards track" protocol, RFC 4340

Designed to meet the needs of real-time applications with flexible congestion control.

TCP-like setup and tear-down handshake.

Bi-directional stream of messages (not stream of bytes).

Messages may be lost or delivered out of order.

DCCP, Cont

Optionally, applications receive notice of message loss.
Doing something (or nothing) about it is up to the application.

Congestion control friendly to TCP.

Sequence numbers by packet, not byte.

Every packet has a 48-bit sequence number.
No retransmission, so numbers not reused.

DCCP Acknowledgement System

Most packets contain a "greatest sequence number received" acknowledgement number.

Some contain an ack vector, which describes the status of each sequence number.
These reports are repeated until the sender acknowledges the acknowledgement status notice.

Explicit Congestion Notification.

Stream Control Protocol (SCTP)

An alternative reliable protocol. RFC 4960

Connection-oriented
TCP-like setup and teardown

Message-oriented.
Messages delivered reliably.

Multiple stream: Order is enforced per stream.
Messages may also be "unordered:" not part of any stream.

Application chooses which ordering matters

Endpoints may include a set of IP addresses
Enhances reliability

SCTP

Each SCTP packet contains a common header and one or more chunks.

Chunks may be user data or control messages.
A packet may contain any number of each.

A user data chunk may be an entire message,
or a portion of a long message.
SCTP tries to avoid IP needing to fragment its packets.

SCTP

Each data chunk has a Transmission Sequence Number (TSN) for acknowledgement.

Each data chunk also has a separate stream sequence number for delivery order.

Some chunks are lists containing TSN acknowledgements.

Congestion control based on TCP.
Similar flow control.

Both DCCP and SCTP provide selective acknowledgement:
Return a list of what we got,
rather than everything before this.

Sources

Comer, *Computer Networks and Internets*
(Our beloved textbook.)

RFC 1323 RFC 2018 RFC 2581

RFC 3168 RFC 3286

RFC 4340 RFC 4336