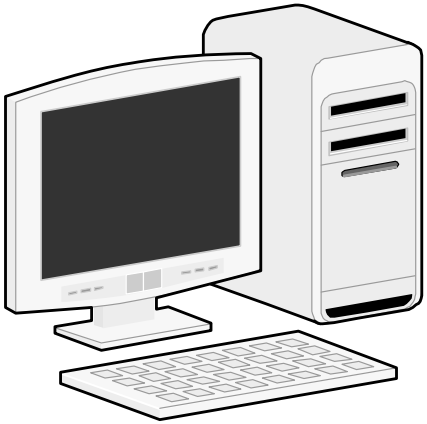
Engineering Overview of Computer Networking

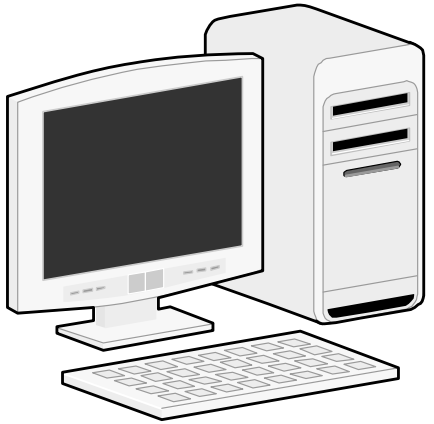
What is Computer Networking?

Logical separation of tasks in digital systems

Computation: Local operations (ALU, load, store, branch, OS, …)

Communication: Data exchange between computation units





Local computation

Request information —communication→ **Accept request**

Process request

Local computation

Receive information ←communication— **Send response**

Local computation

Physical Transmission

Serial data rate at physical layer

Bits per second = bps = b/s

Bytes per second = B/s

1 B/s = 8 b/s

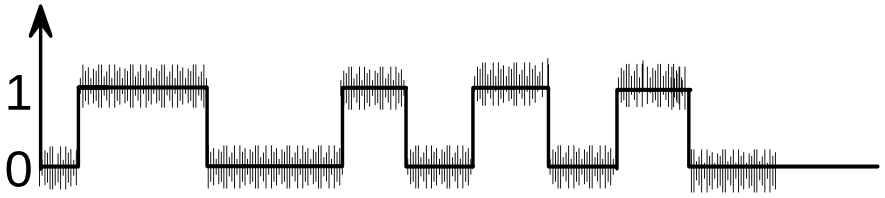
Capacity (bandwidth)

Maximum data rate on medium

Fixed by transmitter / medium / receiver

Limits

- Speed of circuits
- Signal to noise ratio (SNR)



Throughput

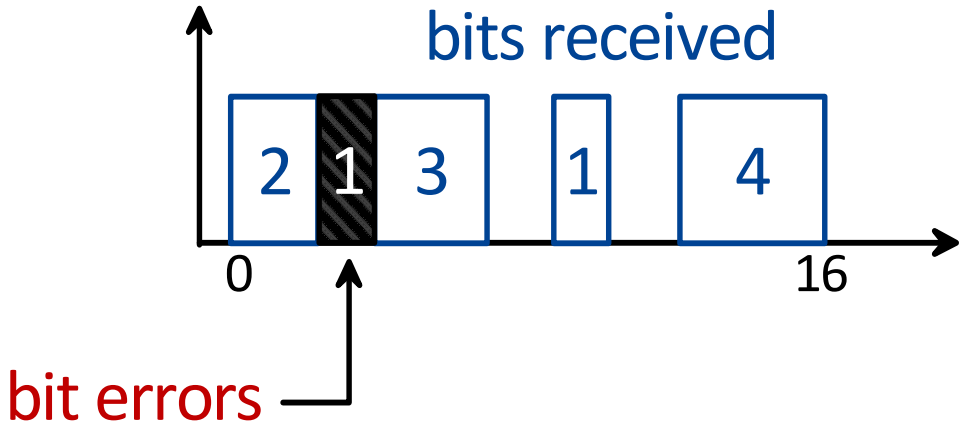
$$\text{throughput} = \frac{\text{error-free data received per second}}{\text{capacity}}$$

Takes account of

- Utilization = % time transmitter sending
- Errors ⇒ re-transmission ⇒ more data on same capacity
- Delays ⇒ less data received on same capacity

utilization = 11 / 16 = 68.75%

throughput = 10 / 16 = 62.5%



Baud Rate
Symbols per second

Symbol

Physical signal that encodes bits

Symbol rate (Baud rate)

Symbols transmitted per second

Bit transmission rate

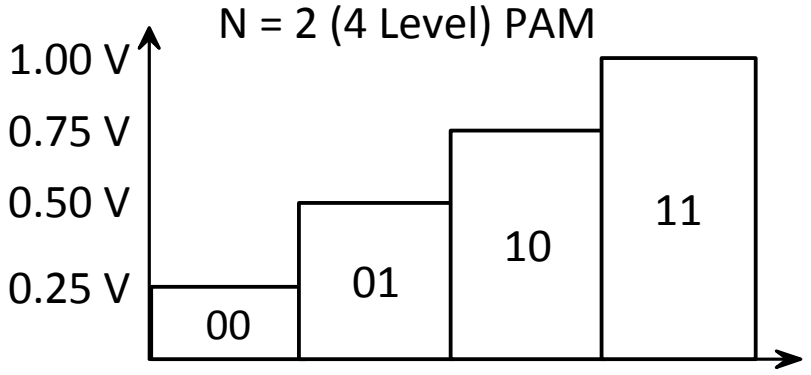
Bits transmitted per second = (symbols / second) × (bits / symbol)

Example

Pulse amplitude modulation (PAM)

Define 2^N electrical levels from 0 to 11…1

Each symbol (level) transmits N data bits



Baud Rate
Symbols per second

33 kbps dial-up modem

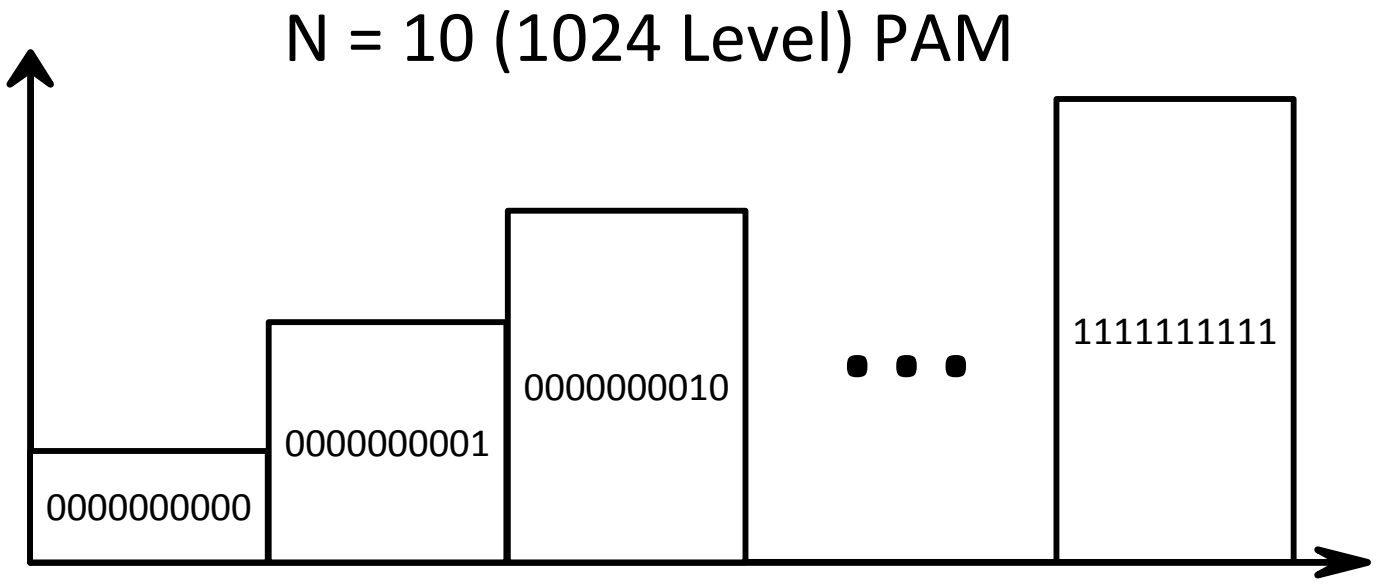
Define 2^{10} = 1024 electrical symbols (max for SNR on phone line)

Baud rate = 3300 symbols / second

Bits transmitted per second

Data rate = (3300 symbols / second) × (10 bits / symbol)

= 33,000 bps



Modeling Information

Information

- Set of possible answers (outcomes) to questions (tests)
  - Finite set (yes/no, day of week, 256 pixel colors, etc)
  - Infinite set ("Guess what happened today!")

Communication — transmission of symbol to receiver

- Before transmission receiver has limited knowledge of symbol
  - Permitted range of symbols (universe of outcomes)
  - Statistical distribution of symbols within range
- After transmission receiver has better knowledge of outcome
  - Receiver tests message to decides on most likely symbol (outcome)
  - Decision accuracy limited by noise

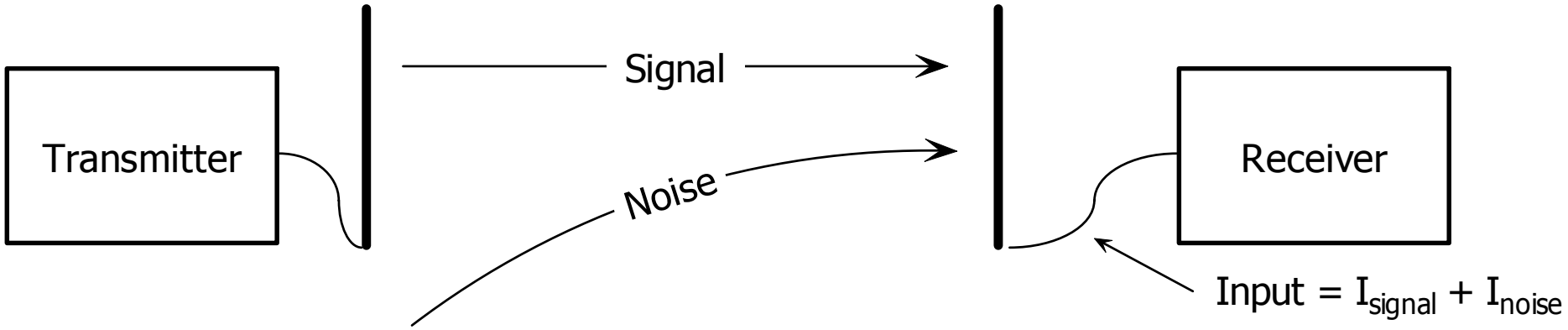
Noise

- Interference, rounding-off errors, resolution of detector, etc.
- Communication does not determine unique outcome

Modeling Noise

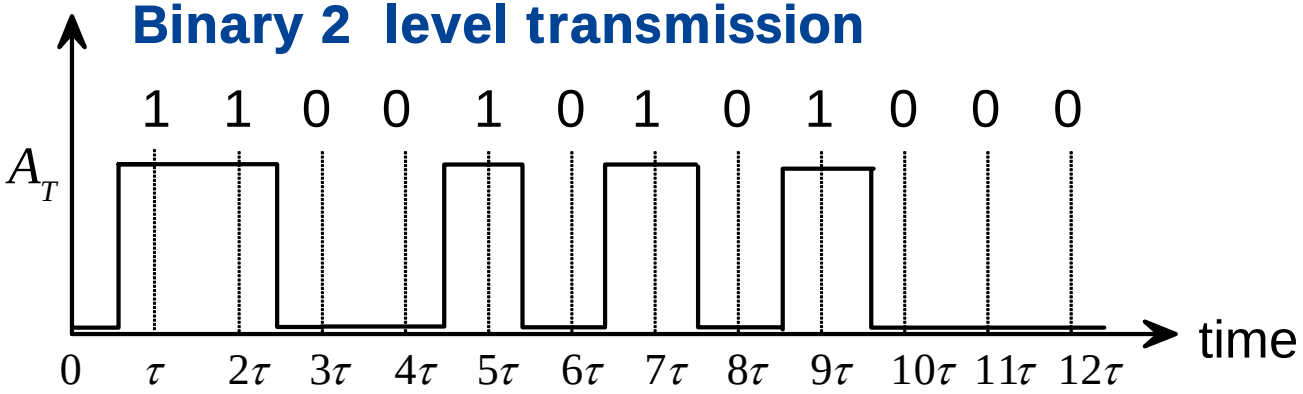
Receiver detects

- Signal from transmitter
- Noise sources
  - Other transmitters
  - Resolution errors
  - Electrical cables and devices
  - Lightening
- Input
  - Electrical current or voltage
  - Sum of Signal and Noise



Signal and Noise Inputs in 2-Level Transmission

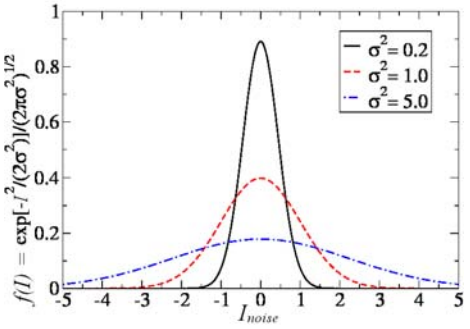
Binary 2 level transmission



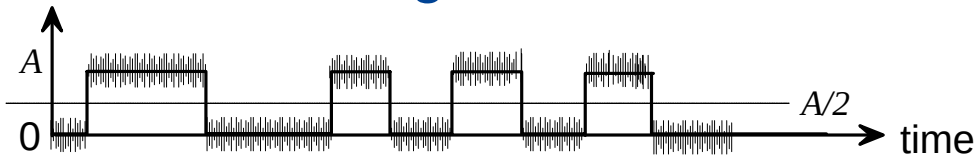
Gaussian additive noise



$$\langle I_{noise} \rangle = 0 \qquad \langle I_{noise}{}^2 \rangle - \langle I_{noise} \rangle^2 = \langle I_{noise}{}^2 \rangle = \sigma^2$$



Received signal = faded transmission + added noise



$$\langle I_{signal} \rangle = \tfrac{1}{2} A \qquad \langle I_{signal}{}^2 \rangle = \tfrac{1}{2} A^2$$

Decision

Signal < $A / 2$ $\Rightarrow$ binary 0

Signal > $A / 2$ $\Rightarrow$ binary 1

Signal to Noise Ratio

$$SNR = \frac{\langle I_{signal}{}^2 \rangle}{\langle I_{noise}{}^2 \rangle} = \frac{A^2}{2\sigma^2}$$

Probability of Error in 2 Level Transmission

$$P(\text{error}) = P(\text{error}\,|\,0)\cdot P(0) + P(\text{error}\,|\,1)\cdot P(1)$$

$$= P(\text{error}\,|\,0)\cdot\frac{1}{2} + P(\text{error}\,|\,1)\cdot\frac{1}{2}$$

$$= \frac{1}{2}\left[P(\text{error}\,|\,0) + P(\text{error}\,|\,1)\right]$$

$$= \frac{1}{2}\left[P\left(I_{noise} > \frac{A}{2}\right) + P\left(I_{noise} < -\frac{A}{2}\right)\right]$$

$$= \frac{1}{2}\left[\frac{1}{\sqrt{2\pi\sigma^2}}\int_{A/2}^{\infty} e^{-I^2/2\sigma^2}\,dI + \frac{1}{\sqrt{2\pi\sigma^2}}\int_{-\infty}^{-A/2} e^{-I^2/2\sigma^2}\,dI\right]$$

$$= \frac{1}{2}\left[1-\text{erf}\left(\frac{A}{2\sqrt{2}\sigma}\right)\right] = \frac{1}{2}\left[1-\text{erf}\left(\frac{1}{2}\sqrt{\frac{A^2/2}{\sigma^2}}\right)\right] = \frac{1}{2}\left[1-\text{erf}\left(\frac{1}{2}\sqrt{\frac{\langle I_{signal}^2\rangle}{\langle I_{noise}^2\rangle}}\right)\right]$$

$$= \frac{1}{2}\left[1-\text{erf}\left(\frac{1}{2}\sqrt{SNR}\right)\right],$$

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-y^2}\,dy$$

$SNR = 1 \rightarrow P(\text{error}) = 0.24$

$SNR = 25 \rightarrow P(\text{error}) = 2.1\times10^{-4}$

$SNR = 1023 \rightarrow P(\text{error}) = 1.2\times10^{-113}$

Quantifying Information

Set of possible outcomes (received symbols)

$K = 2^k$ different symbols

Symbol label = k-bit binary integers

Communication content

One symbol of K possible symbols

One label: $k = log_2 K$ bits

Information Rates

System transmits one symbol in τ seconds

Baud rate $W = (1/\tau)$ symbols / second

Bit rate $b = (W$ symbols / second) × (k bits / symbol)

$= W \times k$ bits / second

$= W \, log_2 K$ bps

Shannon's theorem

$K_{\text{maximum}} = 1 + SNR \quad \Rightarrow \quad \text{Maximum Capacity} = W \times \log_2 [1 + SNR]$

Shannon's Theorem — Examples

Average signal power = average noise power

$$\text{Capacity} = W \times \log_2\left[1 + SNR\right] = W \times \log_2\left[1+1\right] = W$$

V.34 modem

Capacity (Bit Rate) = 33,000 bps

Symbol rate W = 3,300 Hz

$$33{,}000 = 3{,}300 \times \log_2\left[1 + SNR\right]$$

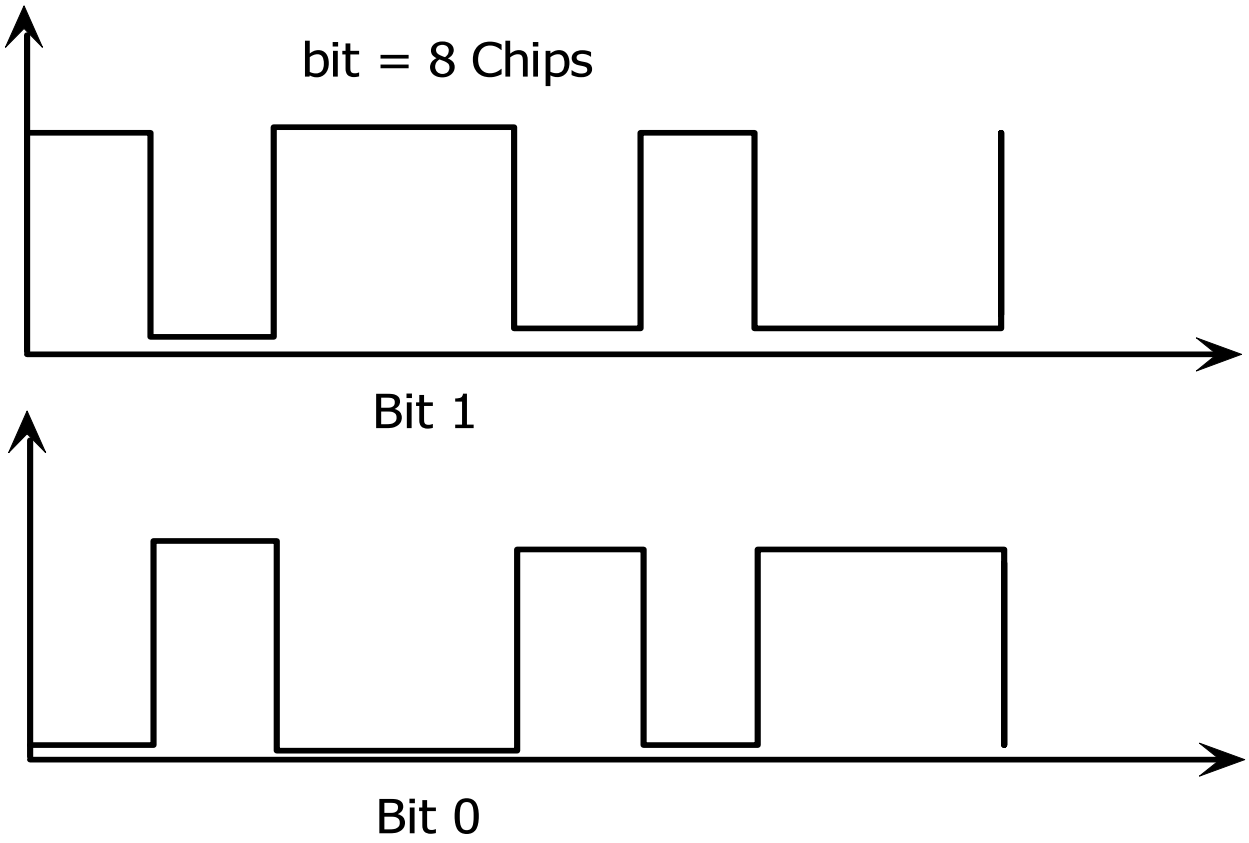
$$\log_2\left[1 + SNR\right] = 10 \quad \Rightarrow \quad \text{Required} \quad SNR = 2^{10} - 1 = 1023$$

Probability of Error in DSSS

Direct Sequence Spread Spectrum (DSSS)

Encode 1 data bit as an m-bit "chip" sequence

1 data bit error $\Rightarrow$ $m/2$ bit errors in one chip



DSSS Lowers Probability of Error

$$P(\text{bit error}) = P(\text{at least } m/2 \text{ chip errors})$$

$$= P(m/2 \text{ chip errors}) + P(m/2+1 \text{ chip errors}) + \ldots$$

$$= \sum_{k=\frac{m}{2}}^{m} \binom{m}{k} \left[P(\text{chip error})\right]^k \left[1 - P(\text{chip error})\right]^{m-k}$$

$$= \binom{m}{m/2} P(\text{chip error})^{\frac{m}{2}} + \cdots$$

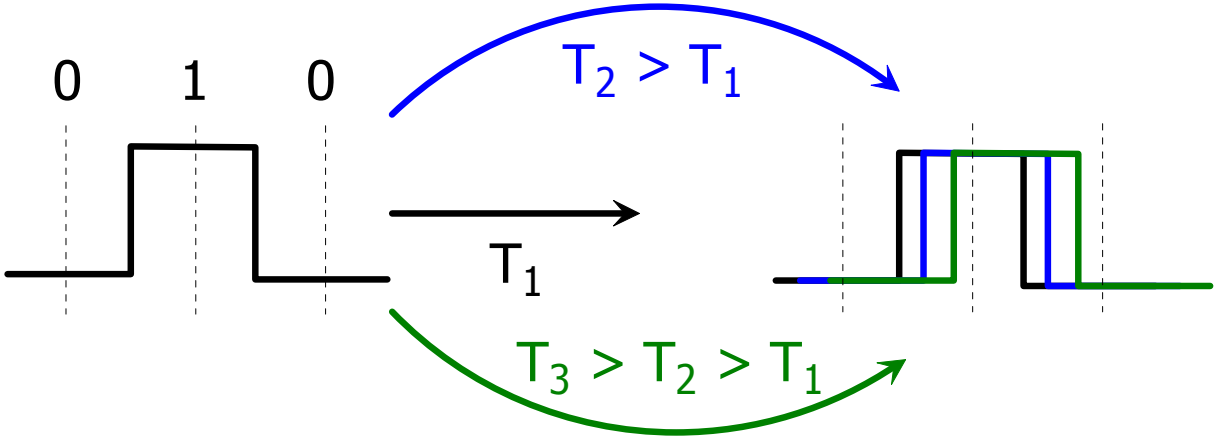
Example

$$m = 8 \text{ bits per chip}$$

$$P(\text{chip error}) = 10^{-2}$$

$$P(\text{bit error}) = \binom{8}{4} \left(10^{-2}\right)^4 \left(1 - 10^{-2}\right)^4 + \ldots \approx 10^{-6}$$

Inter-Symbol Interference



Transmitted signal undergoes multipath delay

Received signal is sum of delayed contributions

Inter-Symbol Interference (ISI)

Interference caused by overlap between sequential bits

Causes bit errors

Jitter

Delay varies from bit to bit

Difficult to determine proper sampling clock

Data Concentration

High capacity link

No single node can utilize link capacity

Example

Optical fiber cable with 4 fibers at 25 Gbps = 100 Gbps

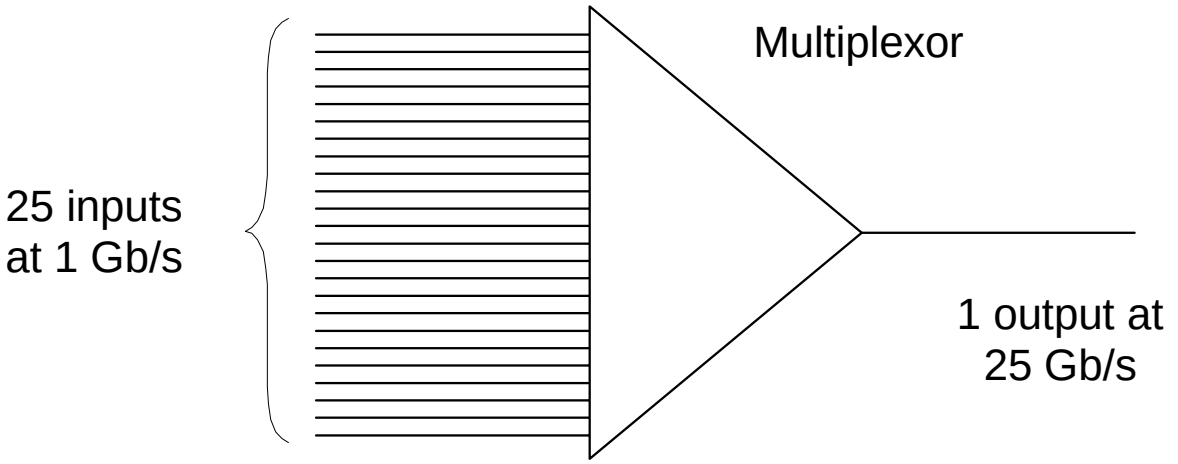
Multiplexing

Combine multiple nodes onto one link

Example

Optical fiber with 25 Gbps data rate

Combine 25 nodes transmitting at 1 Gbps



Multiplexing Methods

Frequency Division Multiplexing (FDM)

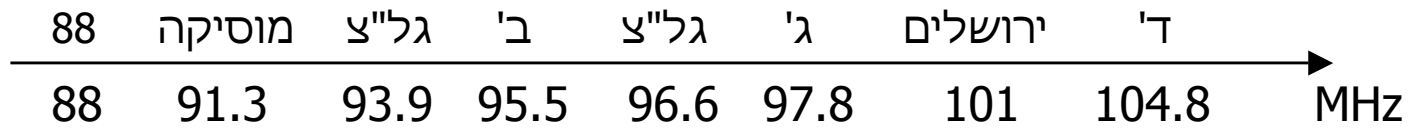
Divide available frequencies (bandwidth) among nodes

Nodes transmit simultaneously on different frequencies

Example

FM radio uses 88 MHz to 108 MHz = 20 MHz bandwidth

Divide 20 MHz into 100 channels = 200 kHz per FM channel



Multiplexing Methods

Time Division Multiplexing (TDM)

Divide capacity into time slots

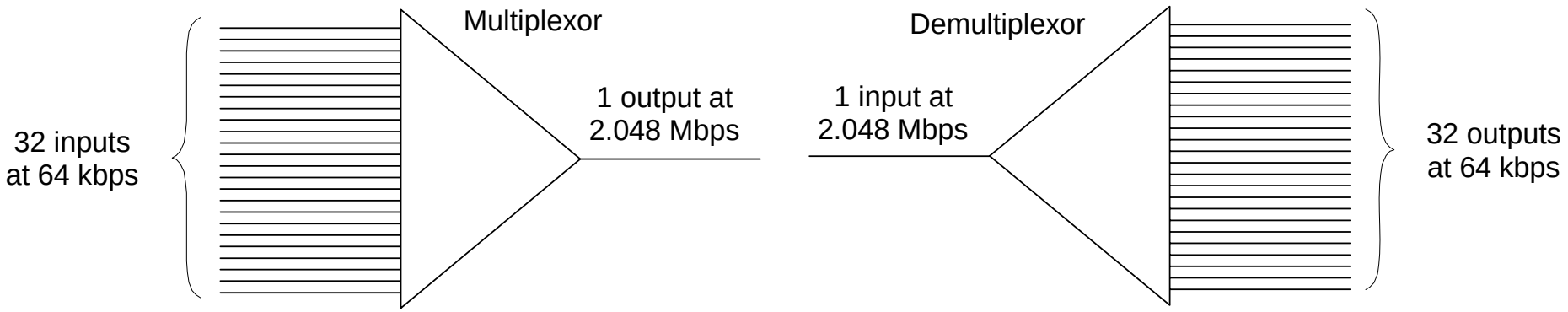
Node transmits in assigned time slot

Example

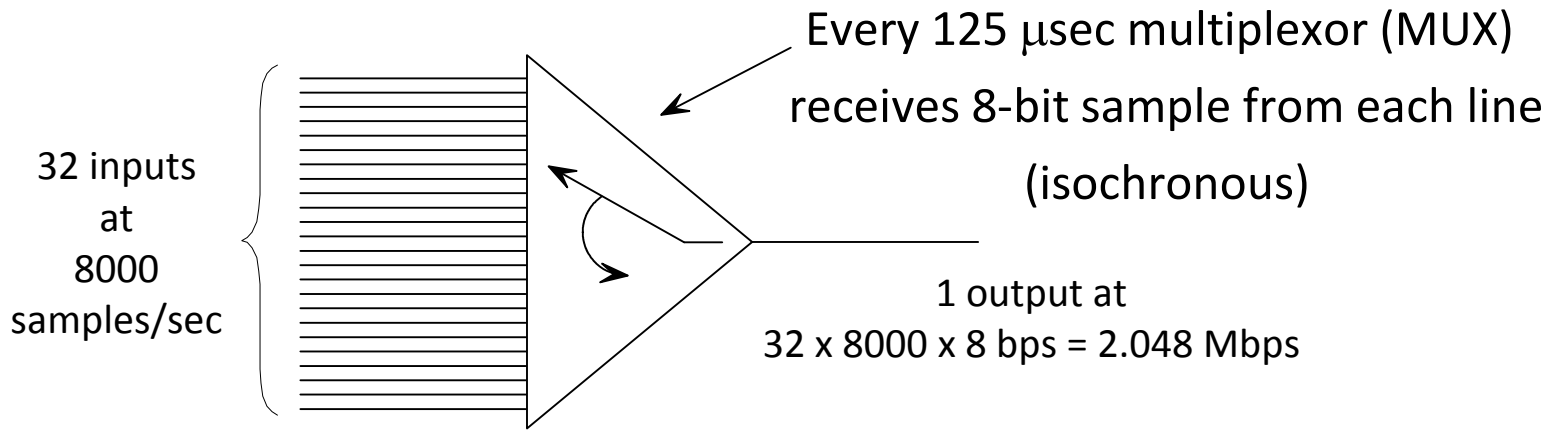
E1 digital line transmits at 2048 kbps

Divide 2048 kbps line into 32 time slots = 64 kbps per node

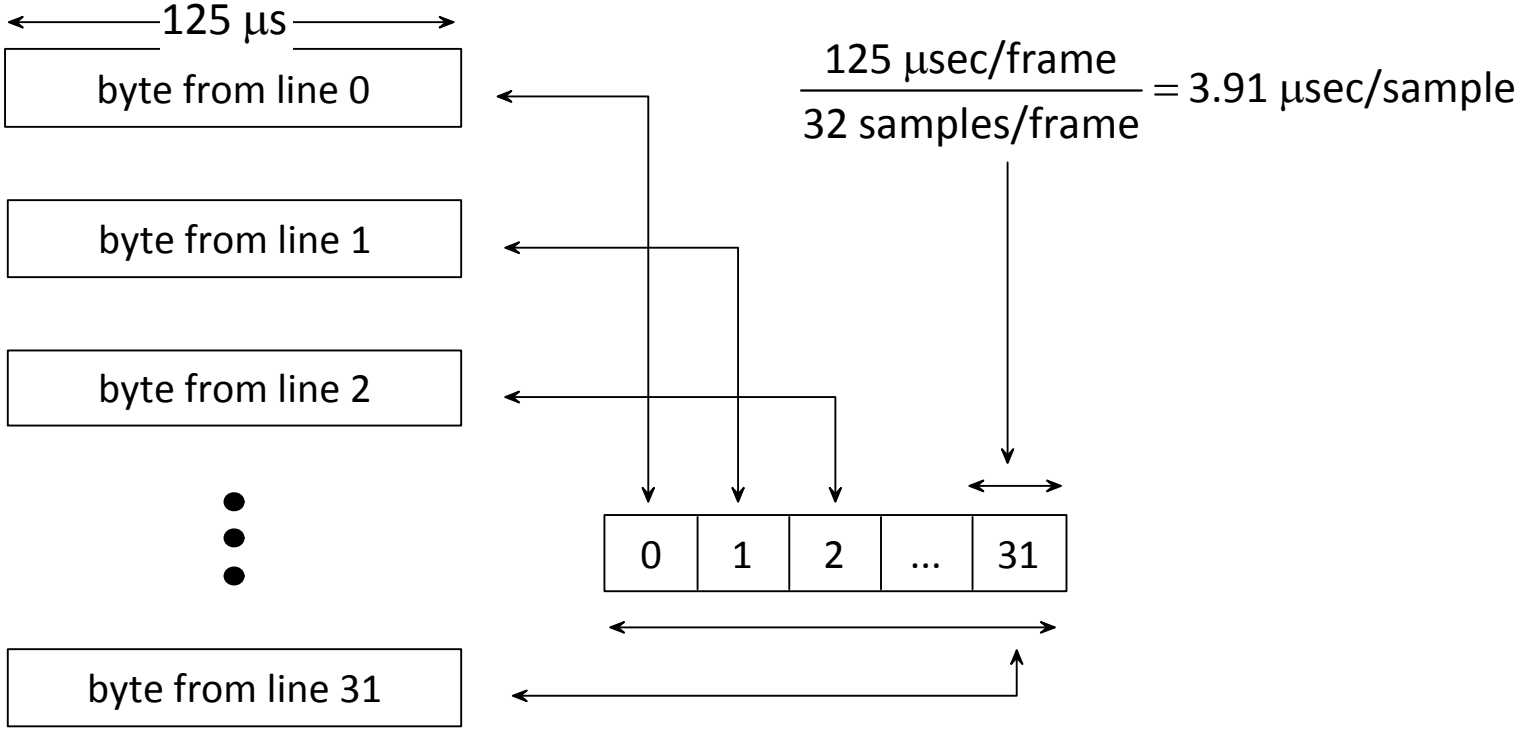
32 x 64 kbps = 2048 kbps = 2.048 Mbps



E1 Multiplex



$$\frac{1}{8000 \text{ samples/second}} = 125 \text{ μs/sample}$$



Mixed Multiplexing
Cellular telephone

Time Division Multiple Access (TDMA)

Used on GSM (2G) and UMTS (3G) phones

Combines FDM and TDM

Frequency Division Multiplexing (FDM)

GSM bands = 25 MHz

Divide 25 MHz into 125 channels = 200 kHz per channel

Transmit 270 kbps over 200 kHz channel

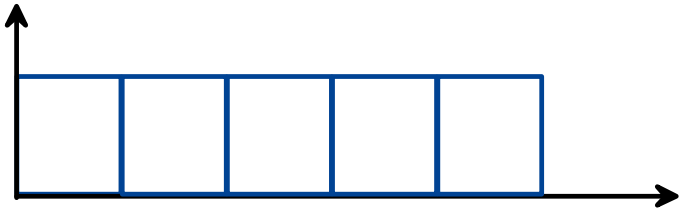
Time Division Multiplexing (TDM)

Divide 270 kbps into 8 times slots = 33 kbps per user

33 kbps = 23 kbps for voice + 10 kbps control

Data Statistics — CBR

Constant Bit Rate (CBR)



Isochronous data

- Equal time interval between bits
- Bits per second = constant

Average data rate

Average data rate = peak data rate = minimum data rate

Example

Uncompressed digital audio

Sample analog signal every T seconds

Round-off sample to n-bit number

Digital audio stream at n / T bps

Multiplexing Statistics

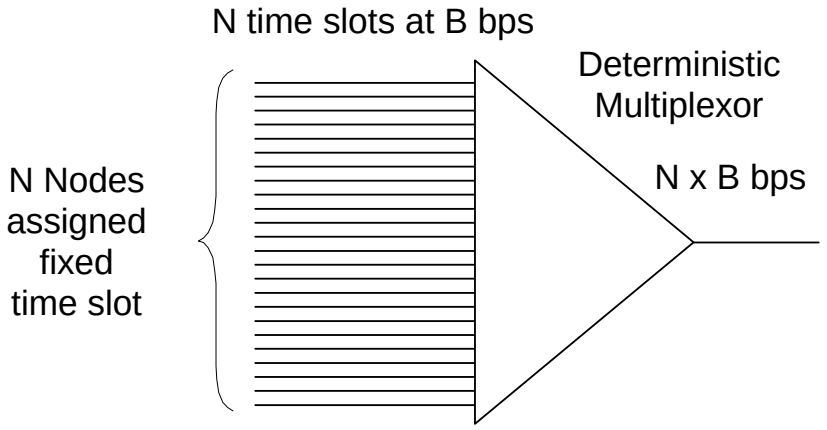
Deterministic multiplexing (CBR)

N Nodes = N time slots

Node reserves fixed time slot

Guaranteed transmission capacity

Node transmits in assigned time slot



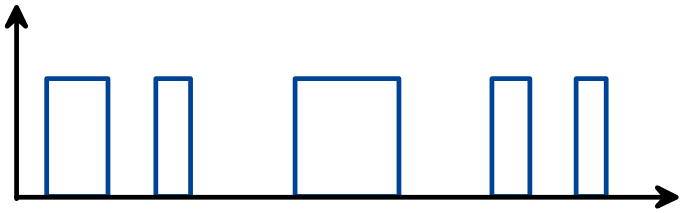
Example

PDH		
E1	32 x DS0	2.048 Mbps
E2	4 x E1	8.448 Mbps
E3	4 x E2	34.368 Mbps
E4	4 x E3	139.264 Mbps

SDH		
STM-0	21 x E1	51.84 Mbps
STM-1	E4	155.52 Mbps
STM-4	4 x E4	622.08 Mbps
STM-16	16 x E4	2.48832 Gbps
STM-64	64 x E4	9.95328 Gbps
STM-256	256 x E4	39.81312 Gbps

Data Statistics — VBR

Variable Bit Rate (VBR)



Bursty data

Peak data rate B > average data rate λ

Assume packets are independent (Poisson statistics)

$P(k,T,\lambda)$ = probability of k bits arriving

in T seconds when average rate = λ

$$P(k,T,\lambda) = \frac{(\lambda T)^k}{k!} e^{-\lambda T}$$

Example

Data sent by time-of-day client

Request time (1000 bits) once every hour (3600 seconds)

Average data rate = 1000 bits / 3600 seconds = 0.28 bps

Peak data rate = 55 Mbps on 802.11g WiFi

Peak data rate B = 55 Mbps > average data rate λ = 0.28 bps

$$P(1 \text{ bit}, 1 \text{ second}, \lambda = 0.28) = \frac{(0.28)^1}{1!} e^{-0.28} = 0.21$$

Multiplexing Statistics

Statistical multiplexing (VBR)

M nodes > N time slots

Bursty data

Average data rate λ < peak data rate B

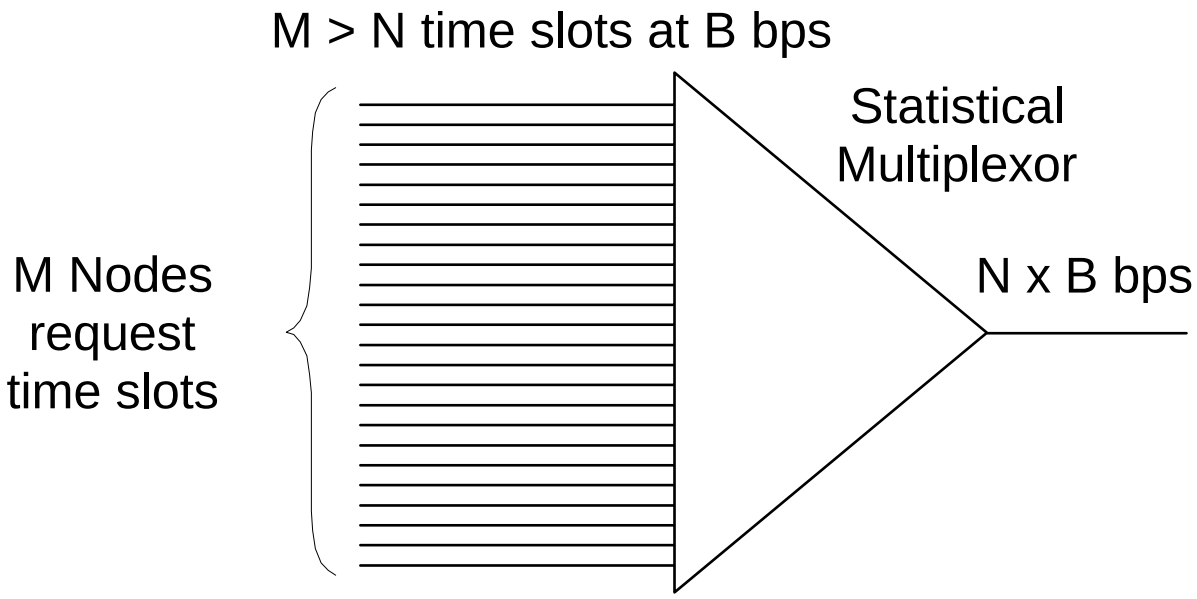
Average traffic = M x λ < capacity = N x B

Actual traffic < capacity $\Rightarrow$ OK

Actual traffic > capacity $\Rightarrow$ data delayed or lost

Example

Internet routers



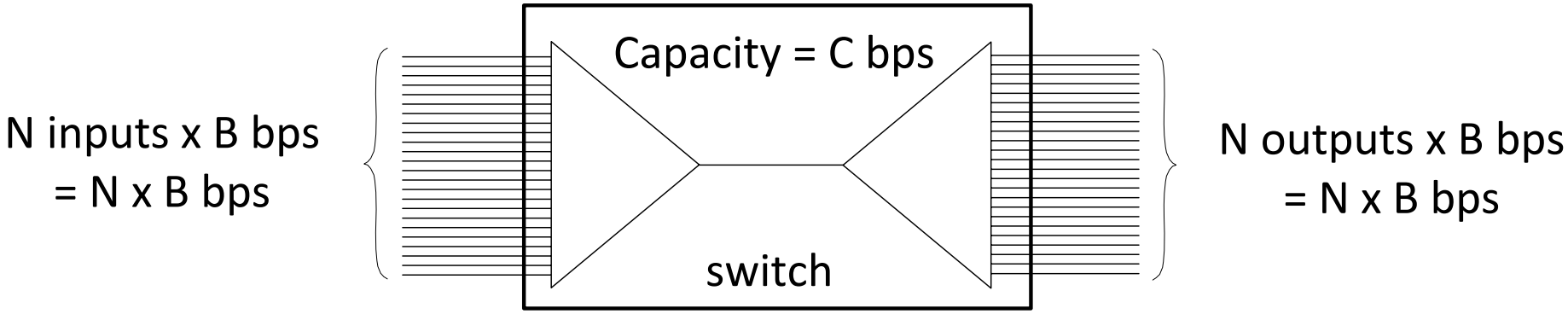
Switching

Switch

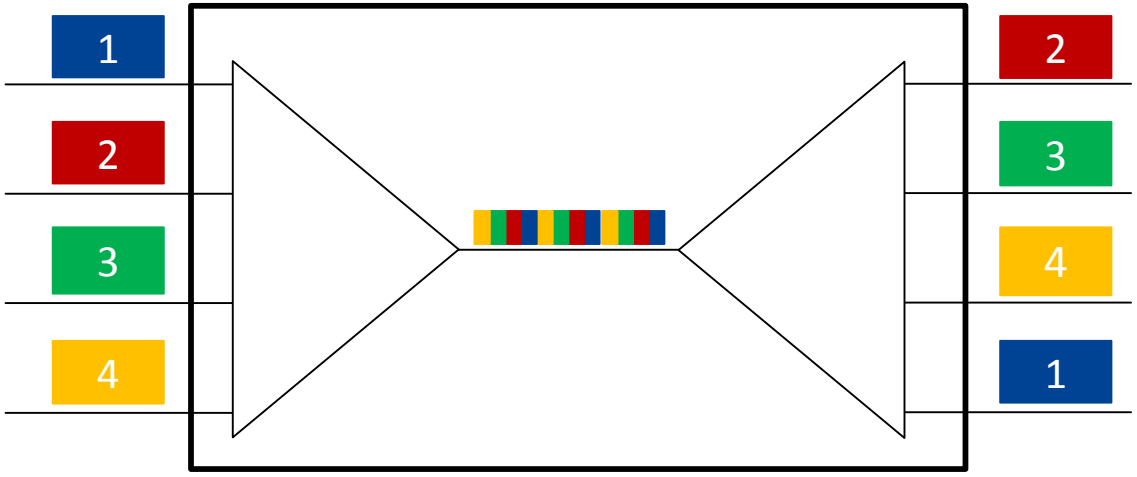
Multiplexor + Demultiplexor

Data at $\texttt{input_port}_i \rightarrow \texttt{output port}_j$

$i, j = 0, 1, 2, \ldots, N - 1$



Example



Circuit Switching — Circuit Mode Connection

Deterministic multiplexing

Capacity C = N × B

Dedicated (reserved) link

$\texttt{input_port}_i \rightarrow \texttt{output port}_j$

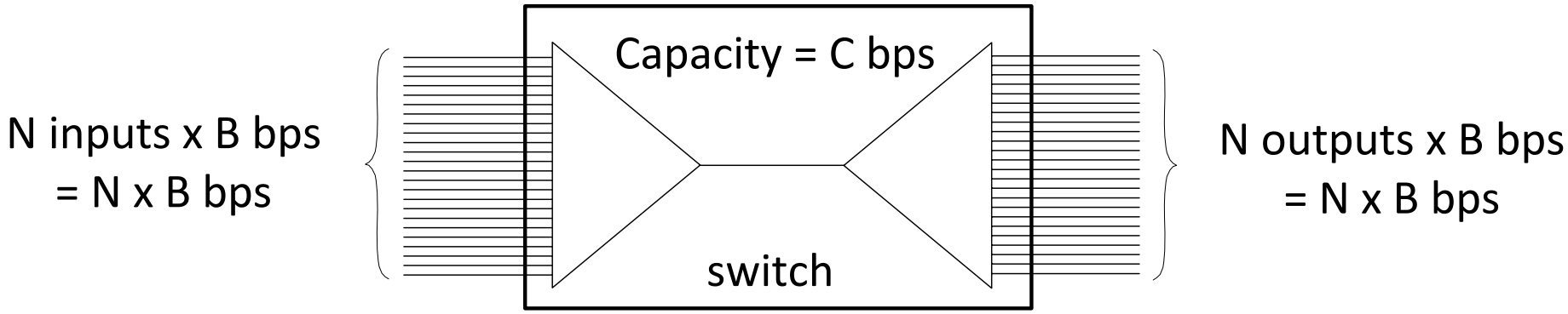
No competition (M nodes = N time slots)

Guaranteed capacity B — if used or not

Example

Bezeq phone call

64 kbps from telephone to telephone (even if no one speaks)



Packet Switching — Packet Mode Connection

Statistical multiplexing

Capacity C = N × B < total possible demand = M × B

Dynamical time slot assignment (on request)

$\texttt{input_port}_i \rightarrow \texttt{output port}_j$

Competition

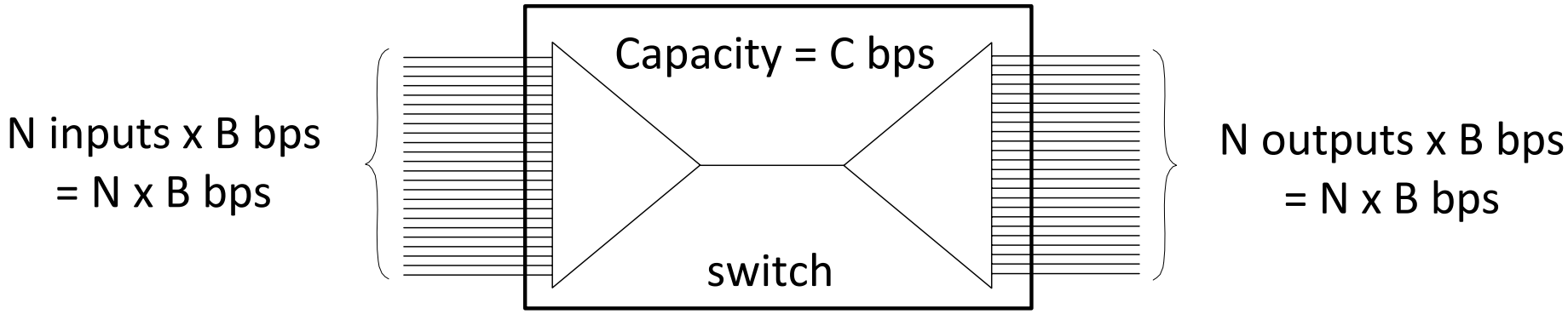
More ports than capacity (M > N)

Demand > capacity ⇒ delay

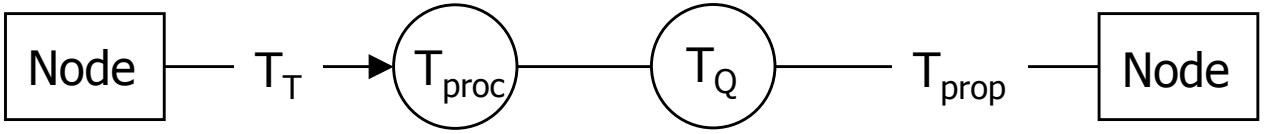
Example

Internet router

Packet queue — first come first served



Message Delay



Transmission delay T_T

T_T = Time to inject bits into line = (bits in packet) / (bits per second)

Example: 1000 Mb / 100 Mbps = 10 sec

Processing delay T_{proc}

Packet processing time in intermediate node

Propagation delay T_{prop}

T_{prop} = (length of cable) / (signal speed)

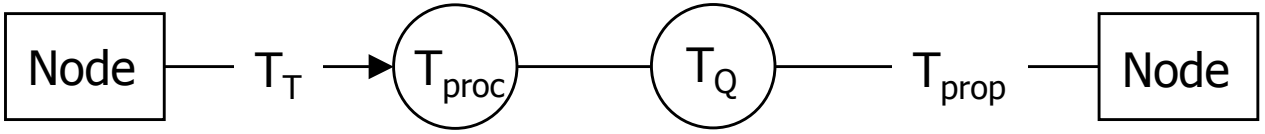
Example: 4 km / (2×10^8 km/s) = 2×10^{-8} sec << 10 sec

Queuing delay T_Q

Time packet waits in buffer for previous packets (congestion)

T_Q = (service time per packet) × (packets waiting in buffer + 1)

Example of Queuing Delay



Queuing delay T_Q

Packets waiting in buffer = utilization / (1 – utilization)

T_Q = (service time per packet) / (1 – utilization)

Queuing delay example

Service time per packet = 10 ms / packet

Service rate = 100 packets / second

Average traffic = S = 85 packets / second

Utilization = (85 packets / second) / (100 packets / second) = 0.85

Buffer level = 0.85 / (1 – 0.85) = 5.7

T_Q = (10 ms / packet) / (1 – 0.85) = 67 ms

Switch capacity C = 100 packets / second

Demand > 100 buffer ⇒ overflow ⇒ excess delay

$$P(\text{demand} > C) = \sum_{k=C+1}^{\infty} P(\text{demand} = k) = \sum_{k=C+1}^{\infty} \frac{S^k}{k!} e^{-S} = \sum_{k=101}^{\infty} \frac{85^k}{k!} e^{-85} = 0.05$$

Error Control

Bit error

Data 1 received as 0 or data 0 received as 1

$$\text{Bit Error Rate (BER)} = \frac{\text{bit errors in received data}}{\text{bits in received data}}$$

Packet Loss

Congestion or buffer overflow → packet discarded

$$\text{Packet loss rate} = \frac{\text{packets lost}}{\text{packets transmitted}}$$

Error detection

Error correction code / redundancy code / checksum

Checksum transmitted with data in header / trailer

Receiver compares independent hash with transmitted code

Error control

Required

- Discard corrupt packet

Optional

- Retransmit discarded / missing packets

Flow Control and Congestion Control

Flow control

Sender avoids overflow of receiver buffer

Congestion control

All senders avoid overflow of intermediate network buffers

Buffer arrival rate

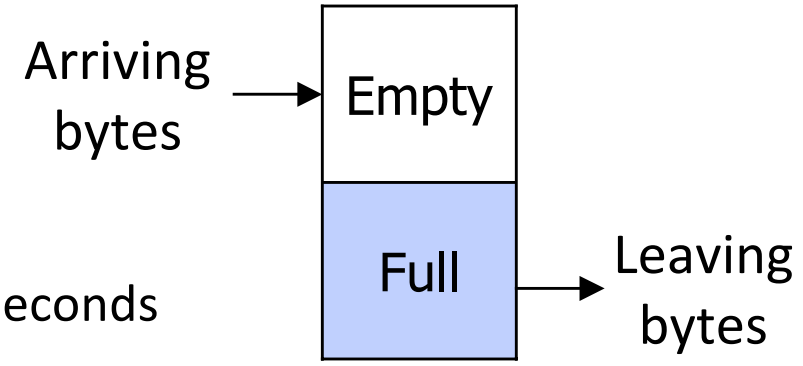
Bytes / second arriving from network

Buffer empty rate

Bytes / second leaving to network or application layer

Buffer file time

$$T_{overflow} = \frac{\text{buffer size}}{\text{buffer arrival rate} - \text{buffer empty rate}}$$



Example

$$T_{overflow} = \frac{64\ \text{KB}}{8\ \text{KB/sec} - 4\ \text{KB/sec}} = \frac{64\ \text{KB}}{4\ \text{KB/sec}} = 16\ \text{seconds}$$

Congestion Control

Queuing theory

Assumptions

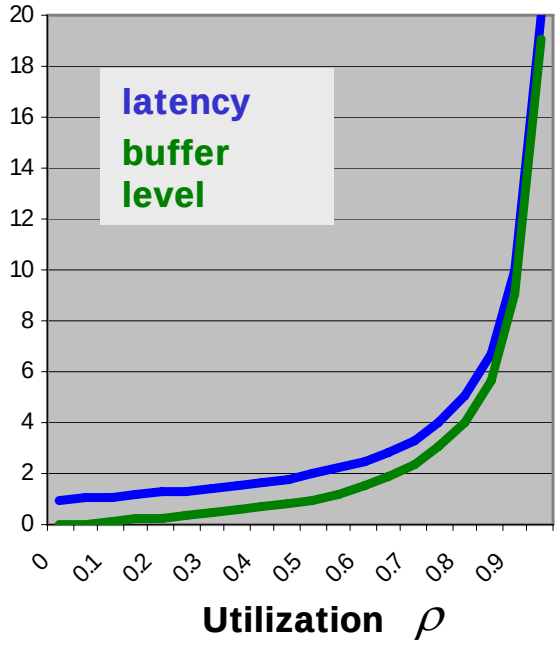
- Data packets arrive independently (Poisson statistics)
  - Random length (bytes)
  - Average arrival rate in steady state
- Data packets leave independently (Poisson statistics)
  - Average emptying rate in steady state

Results

$$\rho = \text{Utilization} = \frac{\text{arrival rate}}{\text{empty rate}}$$

$$\text{Latency} = \frac{1}{\text{empty rate} - \text{arrival rate}} = \frac{1}{\text{empty rate}}\left(\frac{1}{1-\rho}\right)$$

$$\text{Buffer Level} = \text{Latency} \times \text{arrival rate} = \frac{\rho}{1-\rho}$$



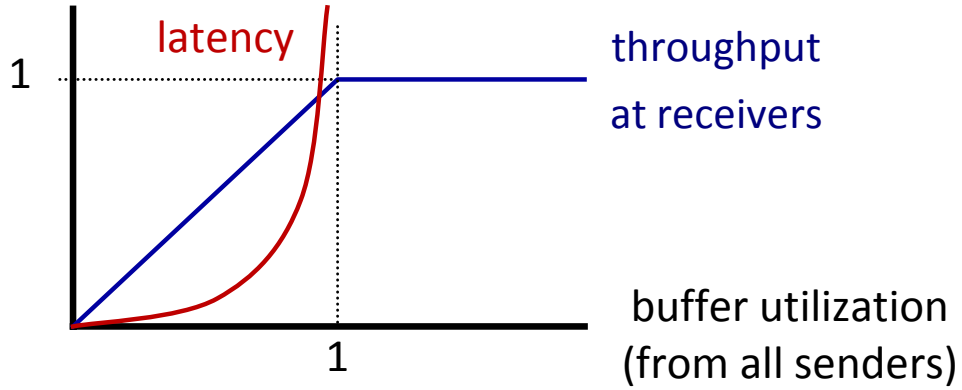
Congestion Control

Buffer throughput

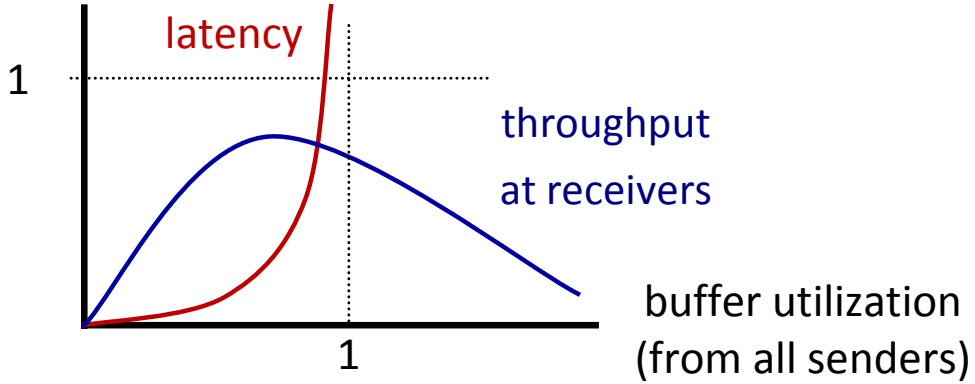
(Over)-simplified throughput model

$$\text{throughtput} = \frac{\text{receive rate}}{\text{maximum receive rate}}$$

$$\text{buffer utilization} = \frac{\text{arrival rate}}{\text{empty rate}}$$



Realistic throughput behavior



High arrival rate at buffer

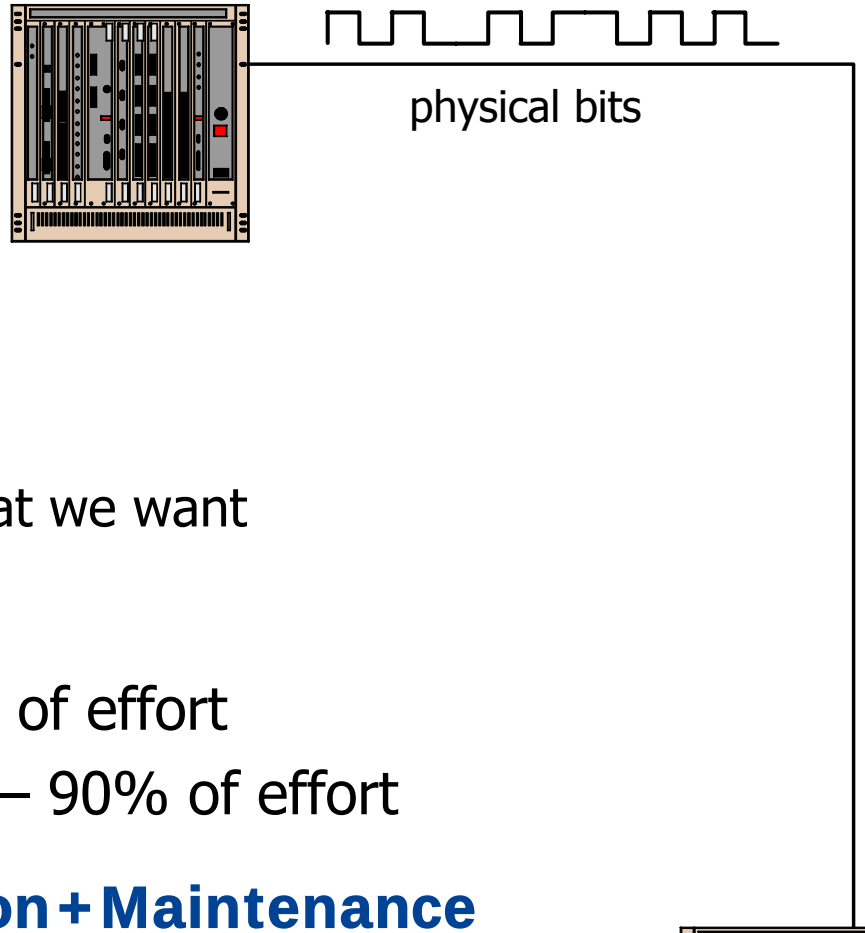
Longer latency + overflow

Sender timeouts

Re-transmit ⇒ more segments ⇒ higher arrival rate at buffer

Data Link

Infrastructure layers



Physical layer (**PHY**)

Physical transmission of bits

Data Link layer (**DL**)

Management of **PHY**

Make physical technology do what we want

Infrastructure management

Delivering data messages — 10% of effort

Making hardware work correctly — 90% of effort

OAM = Operations+Administration+Maintenance

Application assumes infrastructure "just works"

"Just works" ⇒

Reliability, availability, stability, serviceability, growth

Layered Protocol Model

Layered communication

Communication task divided into layers

Protocol stack

Specific peer-to-peer protocol defined at each layer

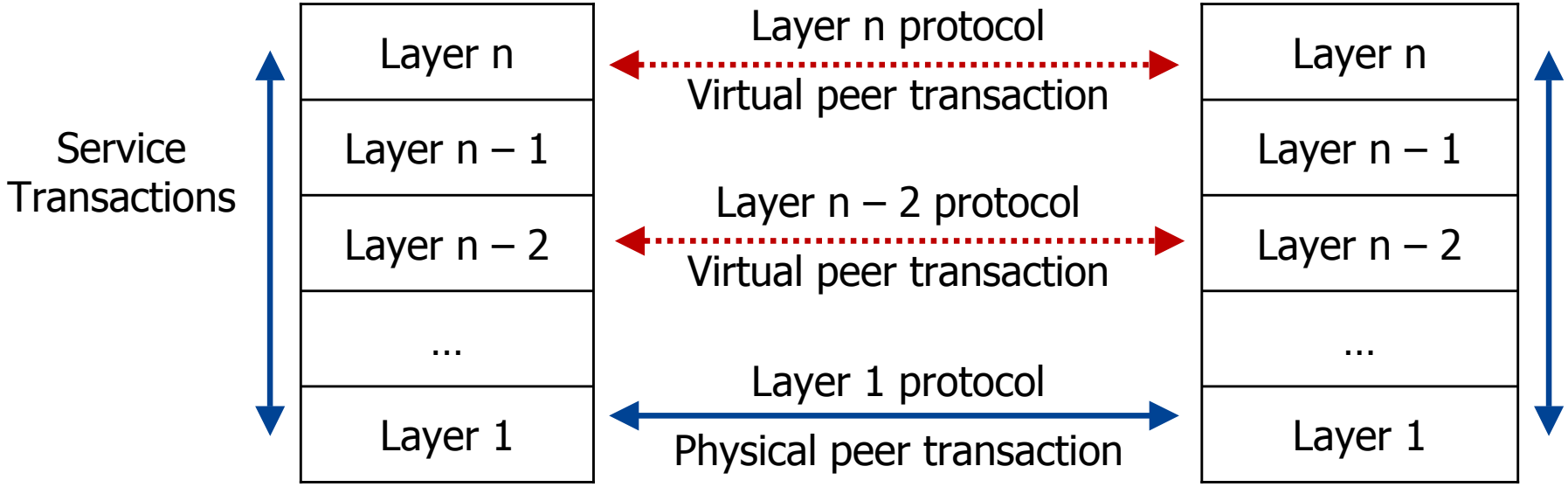
Layer n protocol

Performs **VIRTUAL COMMUNICATION** between layer n peers

Processes only layer n information

Passes request to layer n – 1 for communication service

Receives response from layer n – 1



Encapsulation — Protocol Headers

Layer n – 1 protocol

Receives service request from layer n

Request = message to layer n peer agent

Adds layer n – 1 **HEADER**

Header = message to layer n – 1 peer agent

Service Data Unit (SDU) at layer n – 1

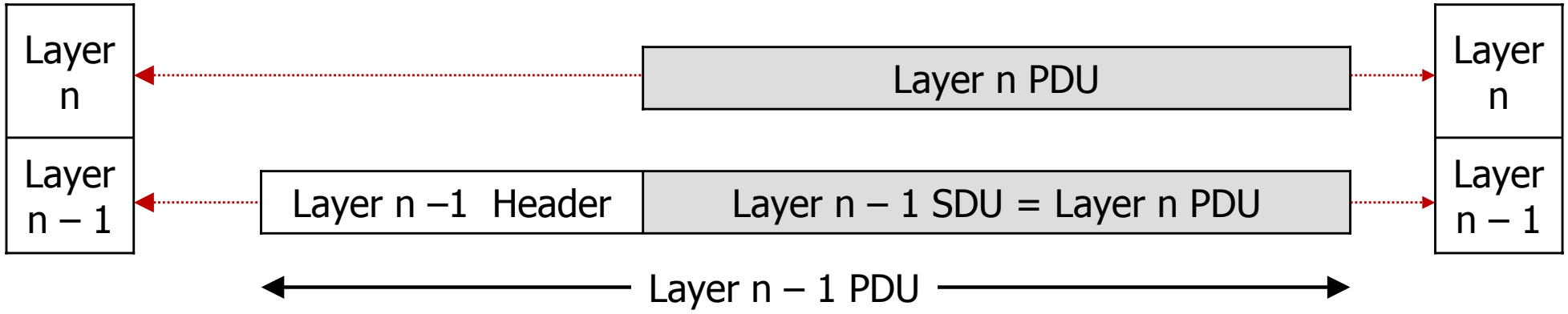
Message received from layer n

Treated as meaningless data by layer n – 1

Protocol Data Unit (PDU) at layer n – 1

Message sent by layer n – 1 protocol

Includes layer n – 1 SDU = layer n – 1 header + layer n PDU



Functional Analysis of Communication

Open System Interconnection Model (OSI)

Layer	Function	Description
7	Application	Exchange of data between user applications
6	Presentation	Syntax and semantics of exchanged data
5	Session	Identification, separation, and continuity of multiple ongoing data transactions between software agents
4	Transport	Reliable end-to-end data exchange between host nodes Prevents data loss, errors, repetitions, ordering errors
3	Network	End-to-end data routing between host nodes via multiple hops
2	Data Link	Control of data transmission between neighboring hardware agents (one hop)
1	Physical	Data transmission between neighboring hardware agents on physical channels (electrical, optical, radio, …)

Internet Functional Model

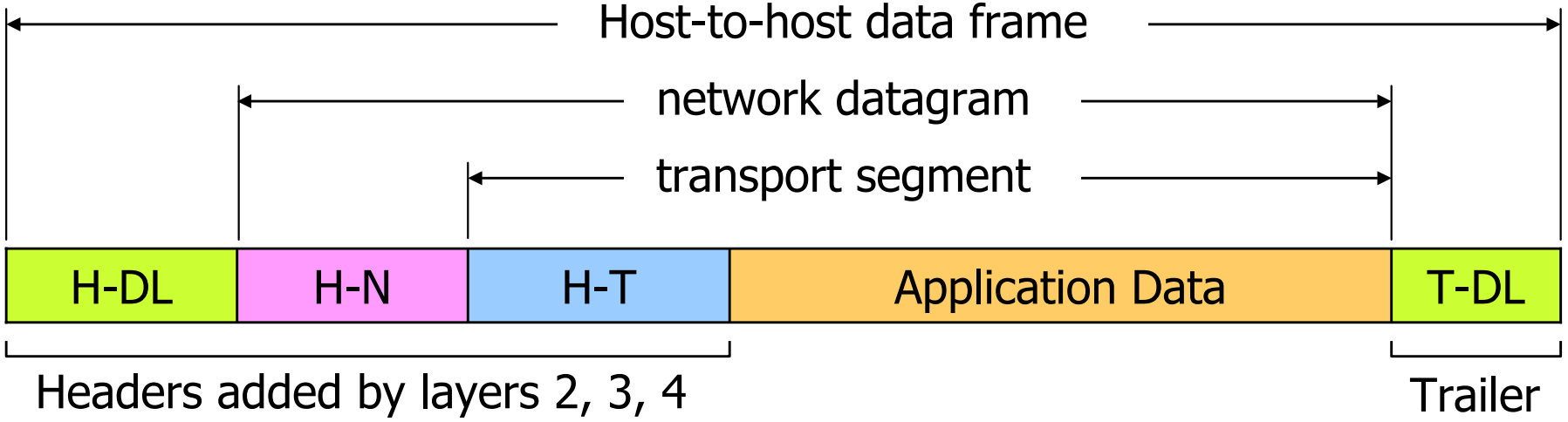
OSI Layer	OSI Function	Internet Layer	Comment
7	Application	Application	Application provides presentation service and some session service (transactions)
6	Presentation		
5	Session	Transport	Internet session management can be: Reliable — with transport service Unreliable — without transport service
4	Transport		
3	Network	Network	End-to-end data routing as in OSI
2	Data Link	Infrastructure	Internet protocols do not discuss physical data transmission
1	Physical		

Ref: http://tools.ietf.org/html/rfc4949

Internet PDUs

Protocol Data Unit (PDU)

Layer	Message	PDU	ID
Application	Data	Message	Socket
Transport	Header	Segment	Port
Network	Header	Datagram	Network (IP) Address
Data Link	Header + Trailer	Frame	Hardware Address
Physical	Bits	Signal	Attachment



Endpoint

Network Address + Port

Data Link Sublayers

Logical Link Control (LLC) sublayer

- Multiplexing of data sources / destinations
- Packet type identification
- Error correction
- Flow control

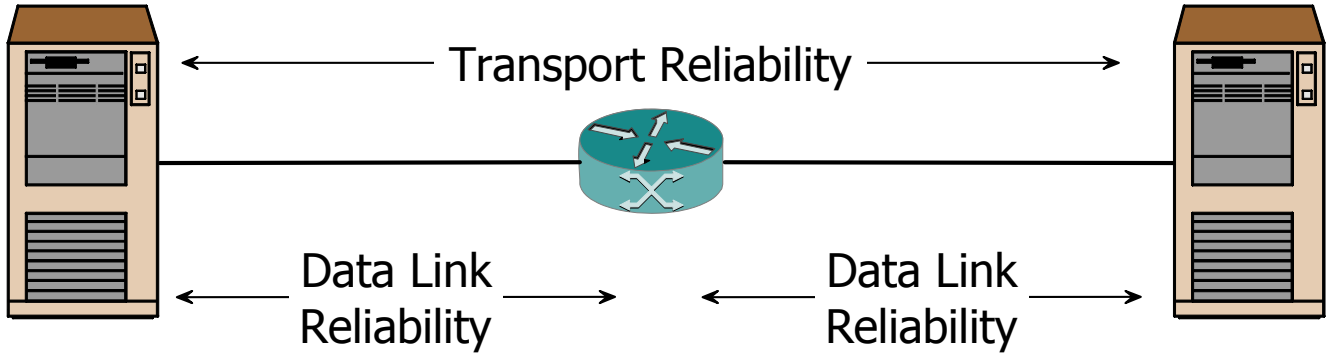
Medium Access (MAC) sublayer

- Network topology
- Medium access management
  - Sharing medium among nodes
  - Permission to transmit
- Data frame structure
- Hardware (MAC) addressing
- Error detection

<table>
<tr><td>LLC Sublayer</td><td rowspan="2">Data Link Layer</td><td rowspan="2">2</td></tr>
<tr><td>MAC Sublayer</td></tr>
<tr><td colspan="2">Physical Layer</td><td>1</td></tr>
</table>

Data Link Functions

Similar to transport layer functions



Framing

Assemble network PDUs into hardware packets

Attach header + trailer for Data Link and Physical layers

Medium access + flow control + congestion control

When / how transmitter sends data onto link

Transmitter avoids overflow of receiver buffer

Transmitters avoid interfering with other transmitters

Error control

Detect / correct transmission bit errors

Local addressing

Transmit locally using hardware MAC addresses

Medium Access Sharing

Time division

- Each host granted full bandwidth in allocated time slot
  - Time slot allocated deterministically or statistically
- Examples
  - Deterministic — telephone switching
  - Statistical — Ethernet, WiFi, …

Frequency division

- Each host granted partial bandwidth in all time slots
- Examples
  - Commercial radio / TV
  - Bluetooth

Code division

- Each host granted full bandwidth in all time slots
- Each host transmits using different coding scheme
- Example
  - Cellular CDMA

Common Shared Medium Networks

Ethernet	IEEE 802.3 local area network Wired time-division access
WiFi	IEEE 802.11 local area network Wireless time-division access
Bluetooth	IEEE 802.15 personal area network Wireless frequency-division access
WiMAX	IEEE 802.16 metropolitan area network Wireless time/frequency-division access
GSM / UTMS	ITU 2G / 3G cellular network Wireless time/frequency-division access
CDMA / CDMA2000	ITU 2G / 3G cellular network Wireless code-division access

Connection Types

Connection

State machine associated with data exchange

- Set-up channel before data exchange
- Monitor channel state during data exchange
- Multiple transactions associated with connection state
- Close channel after data exchange

Connection-oriented

Example — **phone call**

- Connection at application layer
- Enter number → answer call → extended conversation → disconnect

Connectionless

Example — **email message**

- No connection at application layer
- Send email → hope message arrives → hope message is found / read

High-Level Data Link Control (HDLC)

Family of data link protocols

Based on IBM SDLC

- Layer 2 protocol in mainframe SNA
- Originally for communication between CPUs and peripherals

Link Access Protocol (LAP)

- Versions of HDLC used in public network architectures

X.25	LAPB
Frame Relay	LAPF
ISDN	LAPD
Ethernet Logical Link Control (LLC)	IEEE 802.2
Internet point-to-point	SLIP, PPP

HDLC Frame Structure

General HDLC frame

01111110	Address	Control	data	CRC	01111110
8	8	8	≥ 0	16 / 32	8

Address

8 bit address $\Rightarrow$ 256 hardware addresses

Control field

Specifies frame type / control

Information (data)

7	6	5	4	3	2	1	0
0	SEQ N(S)			p/f	NEXT N(R)		

Supervisory (flow control)

7	6	5	4	3	2	1	0
1	0	type		p/f	NEXT N(R)		

Unnumbered (management / connectionless)

7	6	5	4	3	2	1	0
1	1	type		p/f	subtype		

HDLC Internet Protocols

Serial Line Internet Protocol (SLIP)

RFC 1055 (old and rarely used)

0xC0	IP datagram with byte stuffing (C0 → DB DC, DB→ DB DB)	0xC0

Point-to-Point Protocol (PPP)

Layer 2 protocol used between

- Internet routers
- Host and Internet service provider (ISP)

01111110	11111111	11000000	Protocol	Data	CRC	01111110
8	8	8	8 or 16	≥ 0	16 / 32	8

Address = 11111111 = broadcast

HDLC control = 11000000 = Unnumbered (connectionless data)

Protocol

- Protocol in data field
- Network protocol or link negotiation protocol (layer 2 control sublayer)

PPP Protocol Options

01111110	11111111	11000000	Protocol	Data	CRC	01111110
8	8	8	8 or 16	≥ 0	16 / 32	8

Standard network protocols

- IP, IPX, AppleTalk, …
- Datagram in data field

Control sublayers

- Link Control Protocol (LCP)
  - PPP options
    - Header compression (remove control / address fields)
    - Size of protocol / CRC fields and data
  - Test
  - Terminate
- Network Control Protocol (NCP)
  - Network layer options
    - Protocol
    - Address
    - Header compression (encode header fields)
  - Authentication (ISP user / password exchange)

Ethernet Frame

Preamble	Start	Dest Address	Src Address	Type or Length	Data	CRC
7 bytes	1 byte	6 bytes	6 bytes	2 bytes	46 – 1500 bytes	4 bytes

Preamble	7 bytes of 10101010 for sync of receivers	
Start	10101011	
Address	Hardware (**MAC**) address of node 48-bit MAC addresses assigned by OEM and fixed in hardware Broadcast address FF:FF:FF:FF:FF:FF (frame read by all STAs)	
Length	Length of data field (<1500)	
Type	Code identifying protocol in data field Used in most Ethernet systems Type codes > 1536 =0x600	IP = 0x0800 AppleTalk = 0x809B ARP = 0x0806
CRC	CRC-32	

Ethernet Switch (Hub)

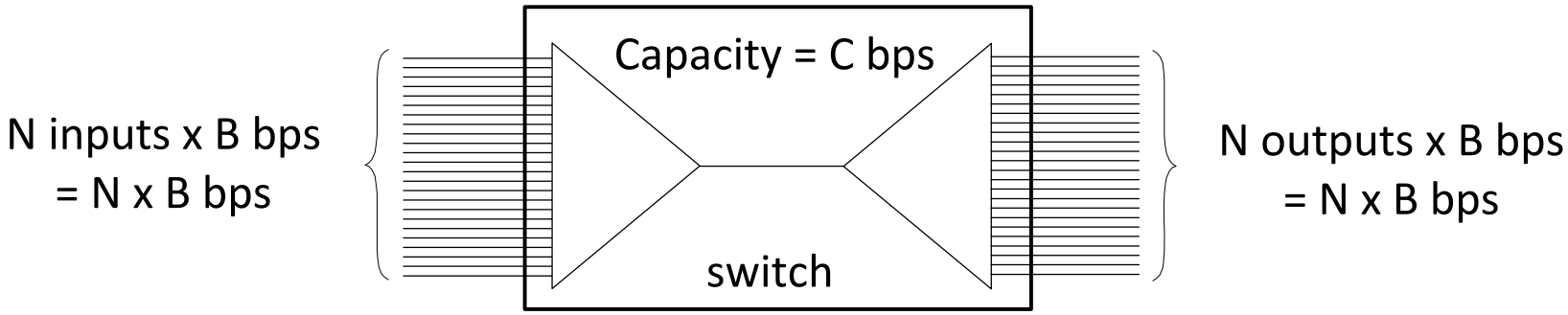
Implements Ethernet medium access (MAC) protocol

Simplest layer 1 + 2 packet switch architecture

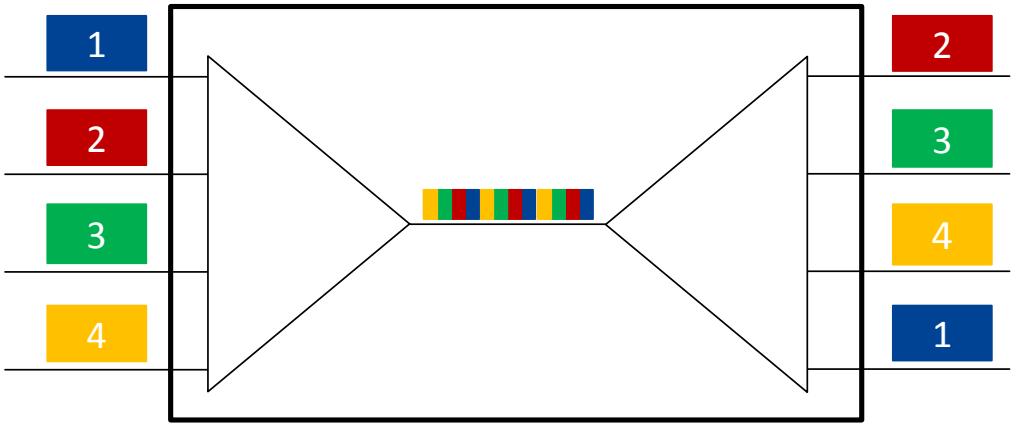
Frame at input port → output port by DEST address

Learning mode

SRC address in frame → associates MAC address with port number



Switch Fabric: typically C = N x B ⇒ no blocking or buffering



Asynchronous Transfer Mode (ATM)

Complex layer 1 + 2 packet switch infrastructure

Path + Circuit Switching

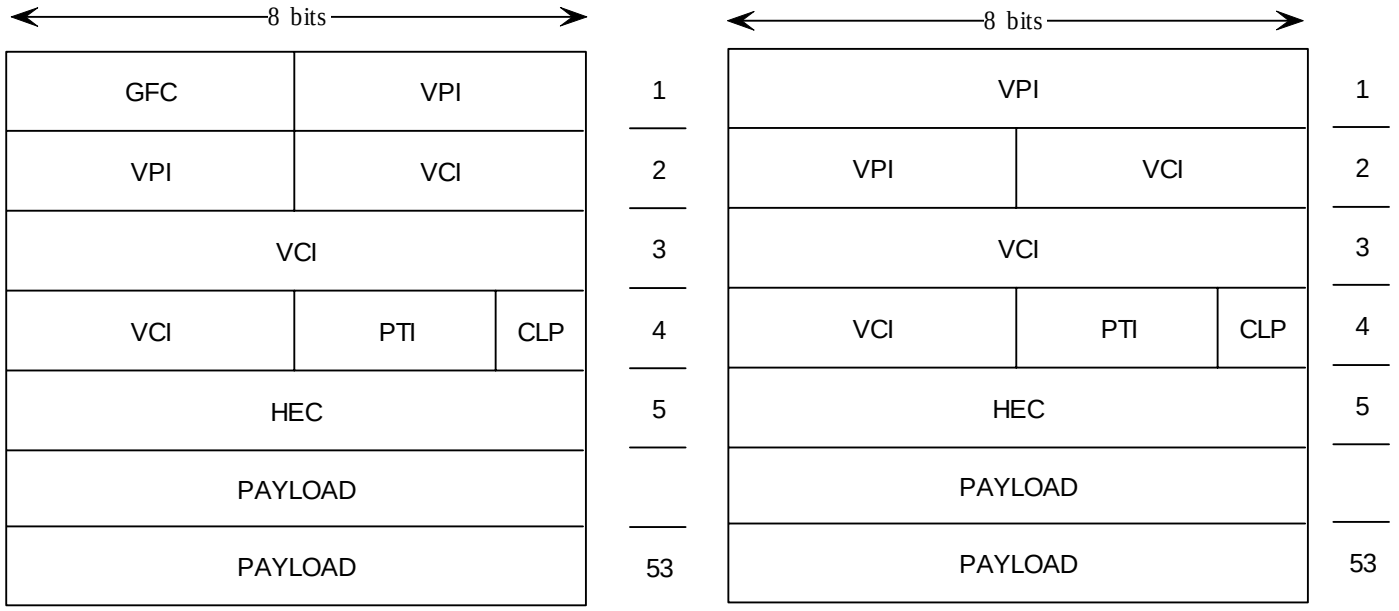
- VPI — virtual path identifier
- VCI — virtual circuit identifier

Capacity

- 2.5 Gbps (STM-16)

Small frames (cells) ⇒ fast priority switching + low latency (delay)

- 53 byte cell = 5 byte header + 48 byte data



at User-to-Network Interface (UNI)

at Network-to-Network Interface (NNI)

GFC
Generic Flow Control

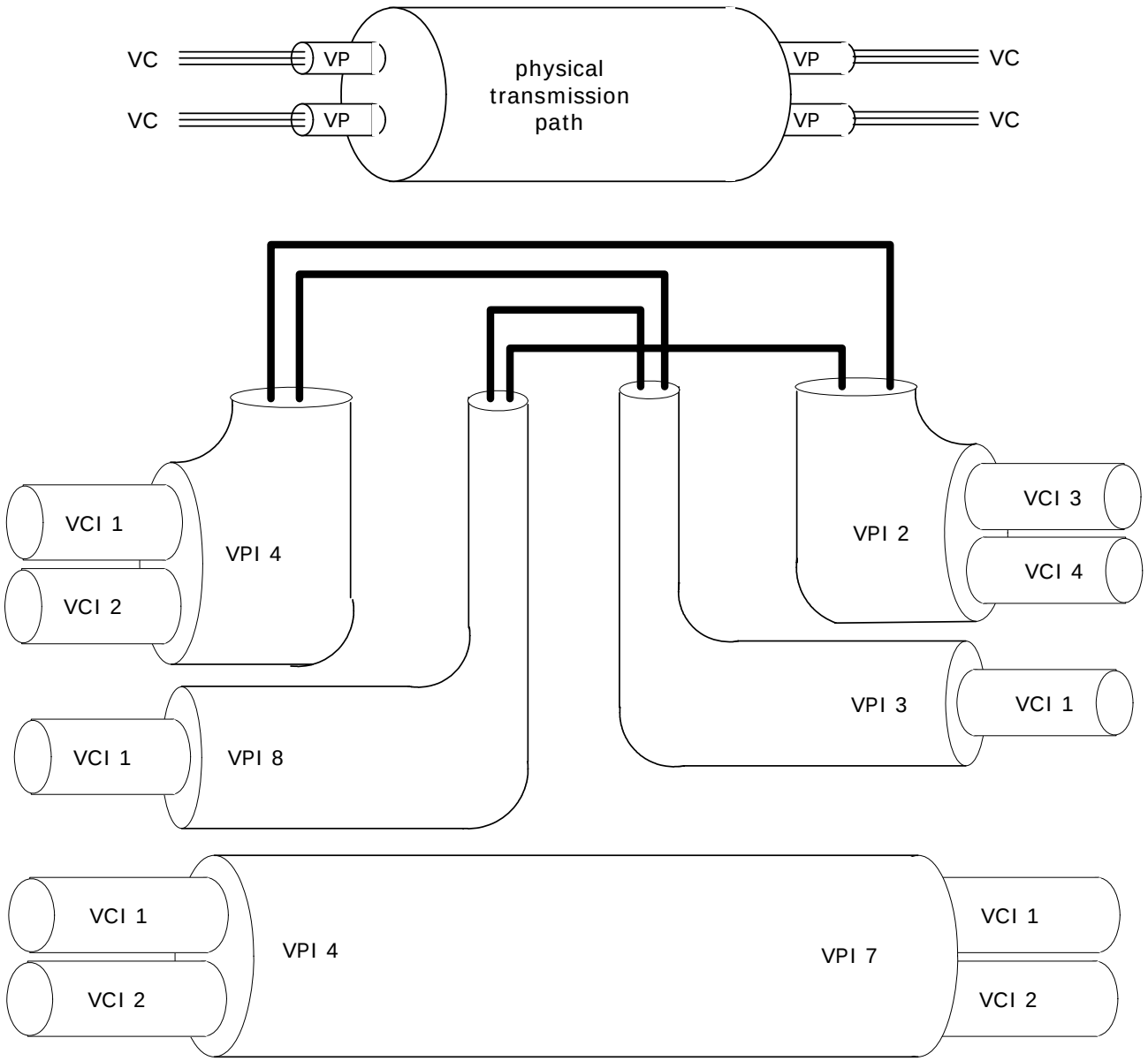
PTI
Payload Type Indicator

CLP
Cell Loss Priority

HEC
header error check

Virtual Path:Virtual Circuit

Path = set of circuits



Quality of Service (QoS) Parameters

Service Level Agreement (SLA)

Defines parameters for each service flow

Service implemented in switching operations

Peak Cell Rate (PCR)

Maximum instantaneous transmission rate

Sustained Cell Rate (SCR)

Average transmission rate measured over time

Minimum Cell Rate (MCR)

Minimum required cell rate

Cell Loss Ratio (CLR)

Percentage of cells lost to error / congestion

Cell Transfer Delay (CTD)

Total in system delay due to propagation + queuing + service delay

Cell Delay Variation (CDV)

Variance of CTD (jitter)

Burst Tolerance (BT)

Maximum burst size (cells) permitted at peak rate

Quality of Service (QoS)

Traffic categories

- Constant Bit Rate (CBR)
  - Emulates isochronous circuit mode
- Variable Bit Rate (VBR)
  - Statistical multiplexing
  - Specified QoS parameters
- Available Bit Rate (ABR)
  - Cheaper statistical multiplexing
  - No guaranteed minimum loss or delay
- Unspecified Bit Rate (UBR)
  - Cheapest statistical multiplexing
  - No guaranteed QoS

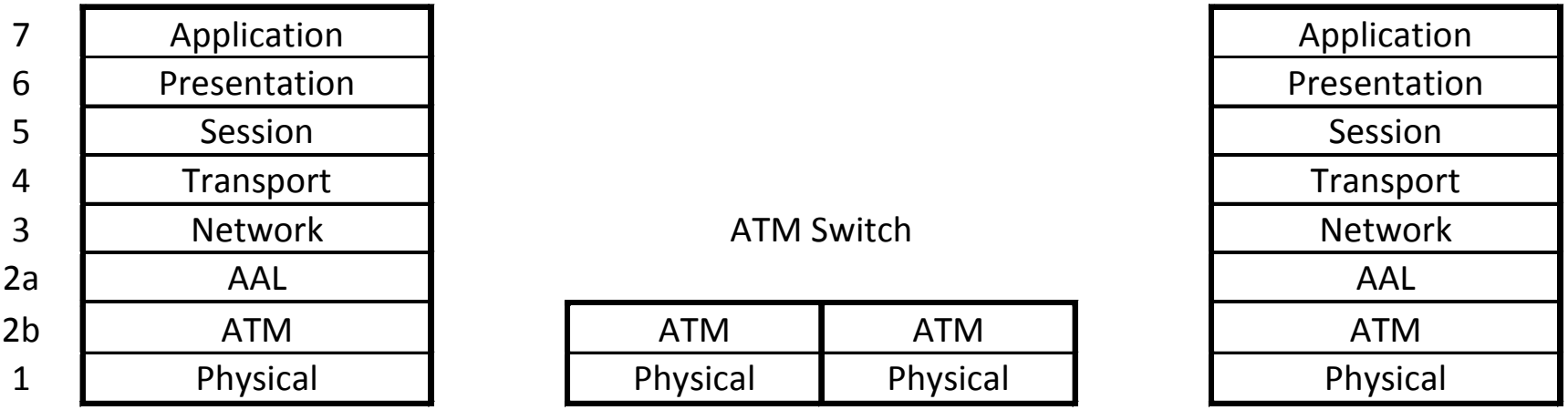
Protocol Layers

ATM Adaptation Layer (AAL)

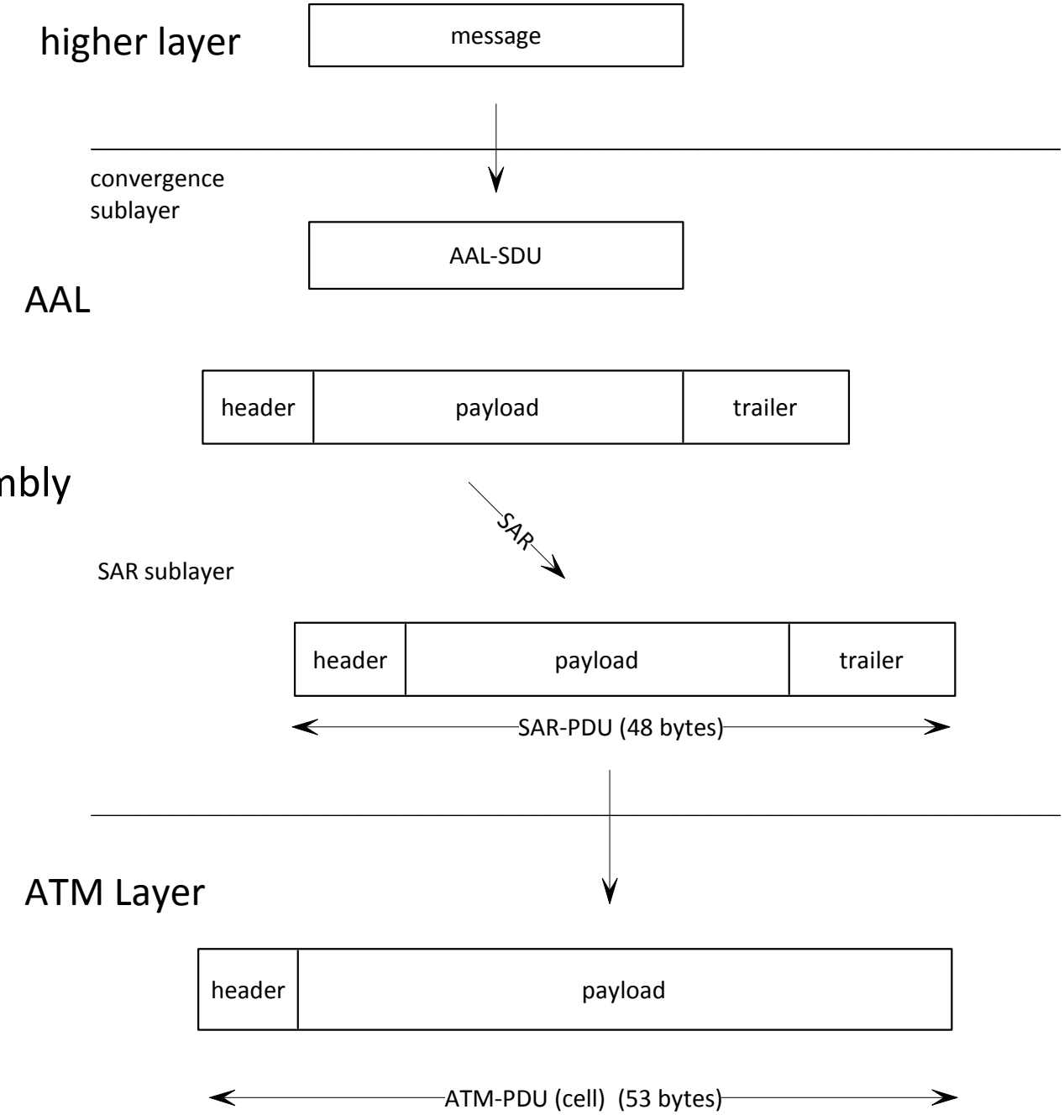
- Convergence sublayer
  - SAP to higher layers
  - Provides service specific functions
- Segmentation And Reassembly (SAR) sublayer
  - Package higher layer data into / from ATM cell
  - Cell loss detection

ATM Layer

- Addressing / Switching
- QoS



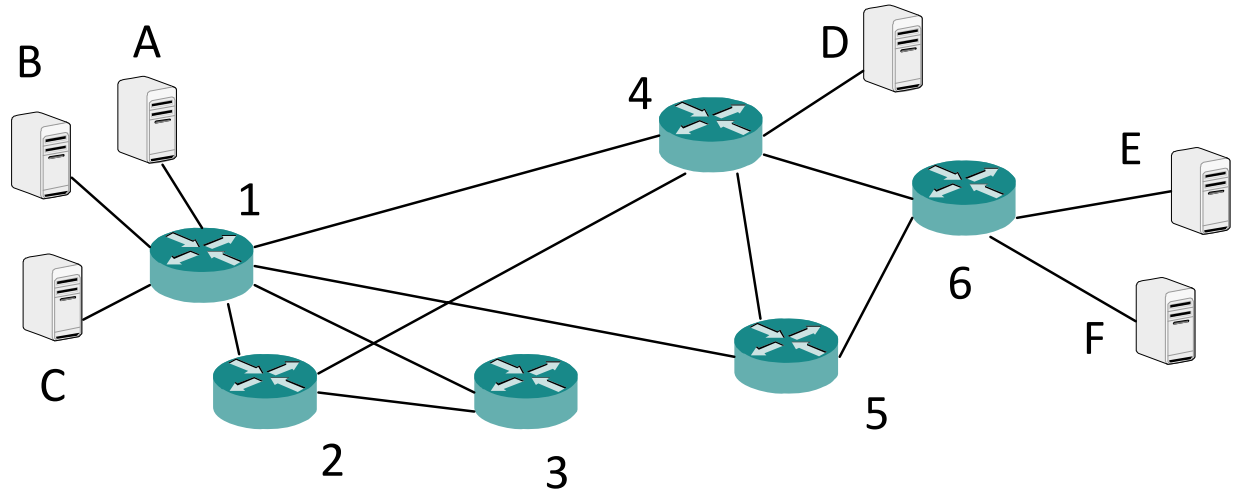
ATM Adaptation Layer



Connection-Oriented Routing

Network of switches and links

Circuit switching or packet switching



Switched Virtual Circuit (SVC)

Set-up / close messages carry source and destination addresses

Example

Set-up VC – 1: B → 1 → 4 → 6 → F

Packet routing by VC ID in header (layer 2 or layer 3)

Every packet follows same VC route

Example

VC – 1	data

Virtual Circuit Labeling

Switch topology

Mapping of node to interface number

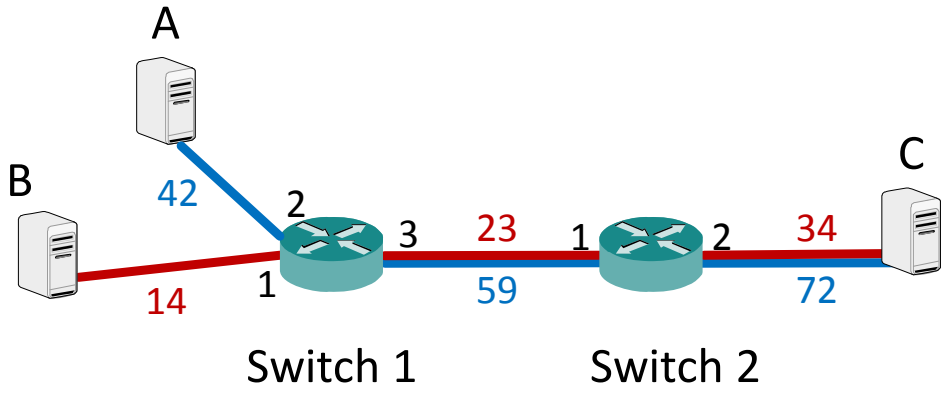
Example

At Switch 1

Interface 1 = Node B

Interface 2 = Node A

Interface 3 = Switch 2



VC number

Labels src-to-dest path

Assigned at routers

Can change at each switch

Example

ATM network packet

VPI	VCI	Type	Priority	CRC	Data

Routing table

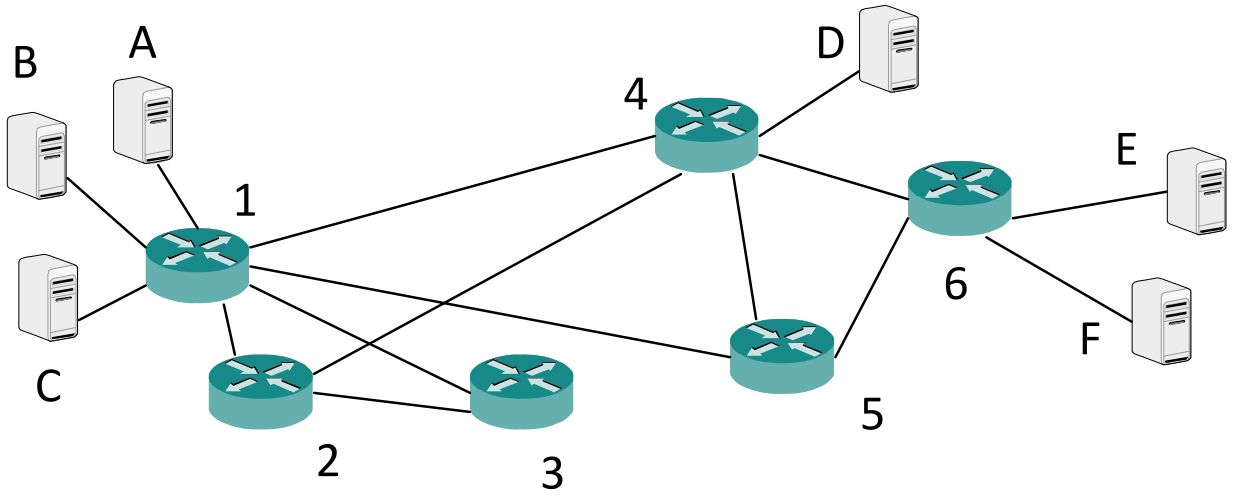
Mapping between interface / VC pairs

Switch 1 Routing Table			
In		Out	
Intfc	VC	Intfc	VC
1	14	3	23
2	42	3	59
3	23	1	14
3	59	2	42

Connectionless Routing

Network of routers and links

Packet switching



Each datagram

Routed individually through network

Has source and destination address in header

Data Link header or Network header

Datagrams may follow separate routes

Example

src = B dest = F	data

B → 1 → 4 → 6 → F

B → 1 → 5 → 6 → F

Datagram Forwarding

Router topology

Mapping of node to interface number

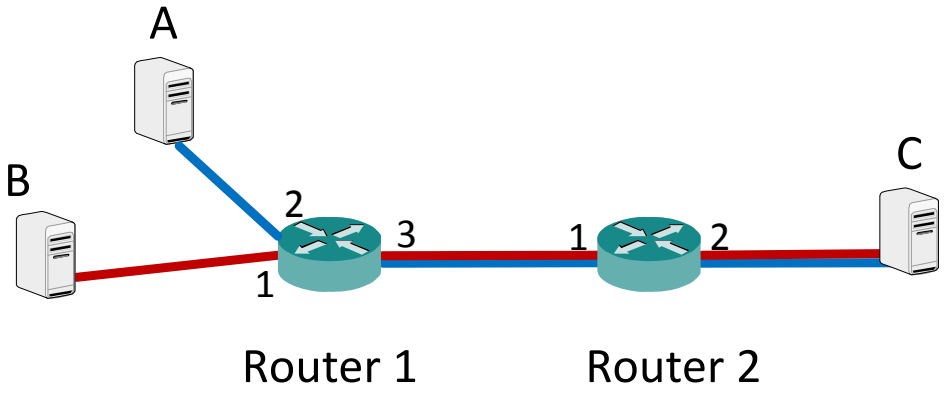
Example

At Router 1

Interface 1 = Node B

Interface 2 = Node A

Interface 3 = Router 2



Node address

Labels node

Router handles next hop

Src + dest address in IP header

src	dest	other fields	data

Routing table

Mapping between dest addr and interface

Switch 1 Routing Table	
In	Out
Dest Addr Range	Intfc
A	2
B	1
other	3

Internet Protocol version 4 (IPv4)

IP datagram format

4 bits	4 bits	8 bits	16 bits
Version	Hlen	Service Type	Total Length (header + data in bytes)
Identification		Flags	Fragment Offset (13 bits)
Time to Live		Protocol	Header Checksum
Source IP Address			
Destination IP Address			
Options			
Data			

Hlen	Header length in 32-bit multiples
Service type (see chapter 5)	Differentiated Services Code Point (DSCP) Explicit Congestion Notification (ECN)
Identification	Provides a unique ID to each datagram
Flags	**0** | **DF** (Don't Fragment) | **MF** (More Fragments — all frags but last)
Fragment Offset	Offset in 8-byte units from start of original datagram (fragmented)
Time To Live (TTL)	Routers perform: **`{if (--TTL == 0) delete datagram}`**
Protocol	Protocol of data carried by datagram (usually TCP or UDP)

Internet Protocol version 4 (IPv4)

IPv4 Addressing

Internet address

32-bit address

Written as 4 octets (8-bit numbers in decimal) separated by dots

Example

www.hadassah.ac.il = 212.179.79.228

Networks / Subnets / Hosts

Network.Host

Network number = 212.179.79.0

Host number = 228

Network.Subnet.Host

Subnet 0 = 212.179.79.1 – 212.179.79.127

Host address range = 212.179.79.X

X = 0xxxxxxx (binary)

Subnet 1 = 212.179.79.128 – 212.179.79.254

Host address range = 212.179.79.X

X = 1xxxxxxx (binary)

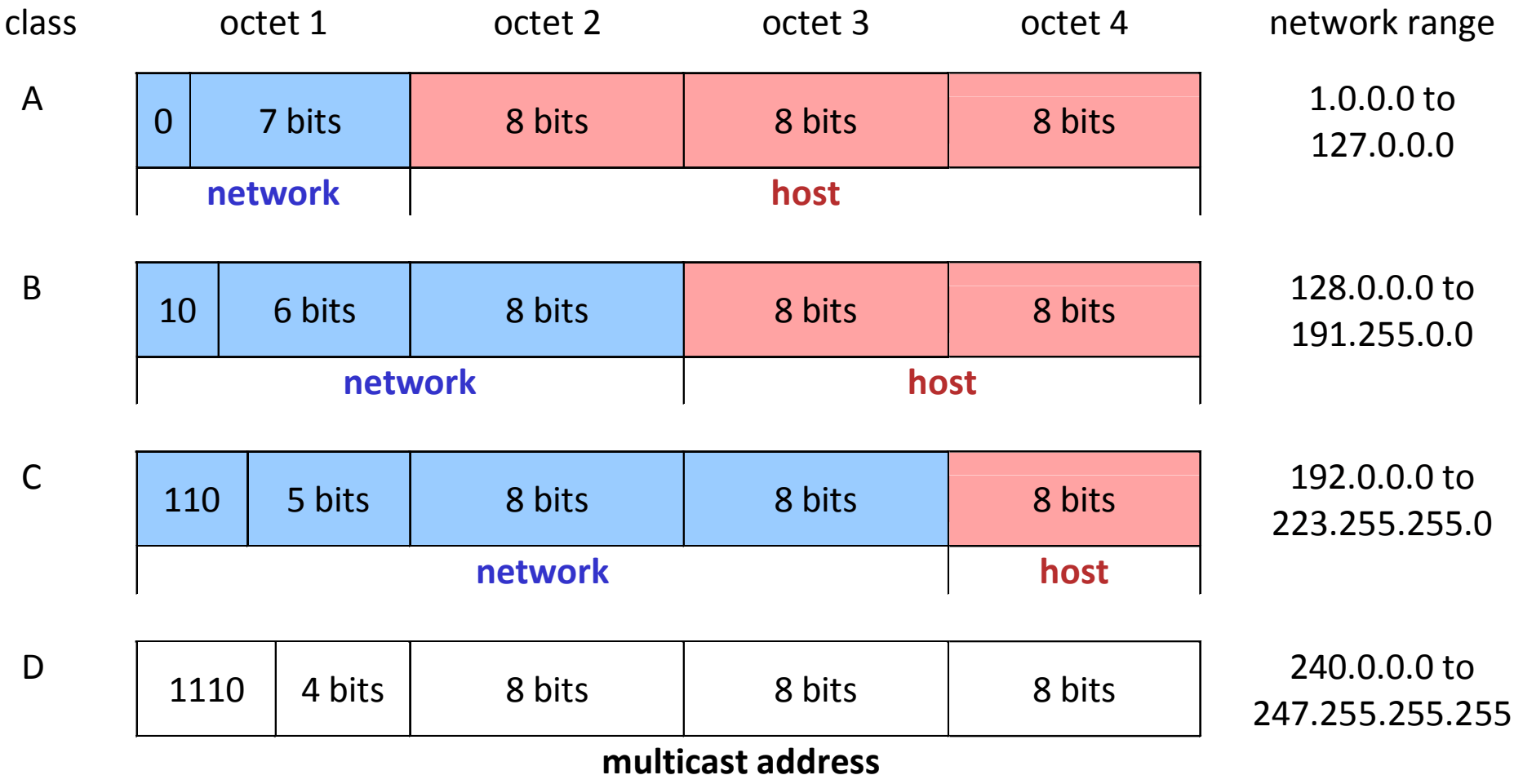
Internet Protocol version 4 (IPv4)

Address classes

3 main address classes

A — small number of large networks (up to 2^{24} = 16 Mhosts)

C — large number of small networks (up to 2^{8} = 256 hosts)



Internet Protocol version 4 (IPv4)

Classless Inter-Domain Routing (CIDR)

Length of network number = 1, 2, 3, … , 31

Not restricted to 7, 16, 24

Address format

`Octet1.Octet2.Octet3.Octet4/bits_in_network_number`

Examples

Class A address — **`10.0.1.5/8`**

8-bit network number = **`10.0.1.0`**

Class C address — **`192.168.0.37/24`**

24-bit network number = **`192.168.0.0`**

General node address — **`192.168.0.33/27`**

192	168	0	33	
11000000	10101000	00000000	001	00001
27-bit network address				host

Network address = **`192.168.0.32`**

Host addresses **`192.168.0.32 – 192.168.0.63`**

Internet Protocol version 4 (IPv4)

Subnet mask

Forming subnet mask

1 in all bits of network number

0 in all bits of host number

Encodes same information as number of bits in network number

Example

General node address — **192.168.0.33/27**

27-bit network number

Mask

11111111.11111111.11111111.11100000

255.255.255.224

Using mask

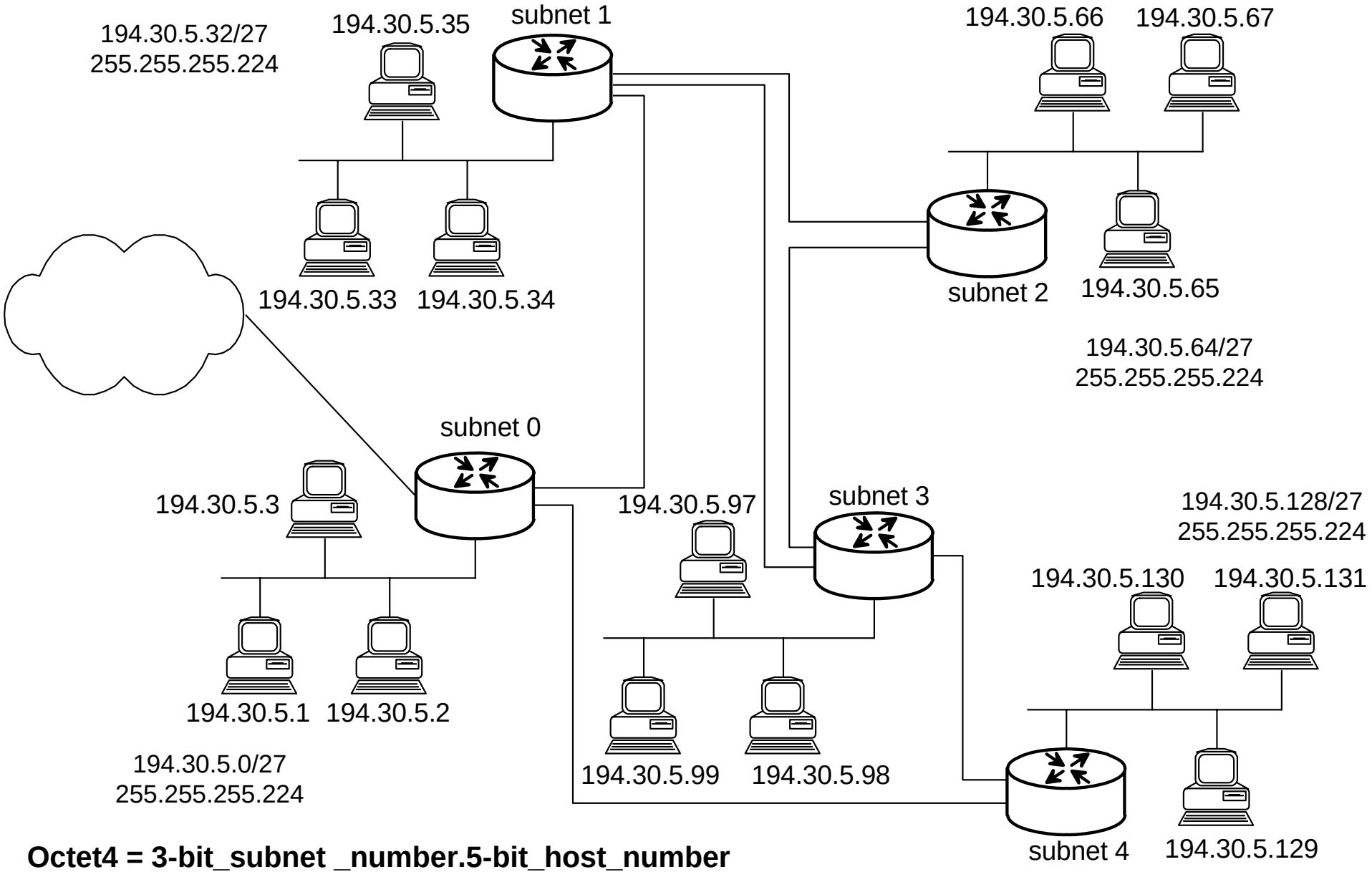
Mask AND IP address = network number

Example

255.255.255.224 AND 192.168.0.33 = 192.168.0.32

Internet Protocol version 4 (IPv4)

Subnet example



Octet4 = 3-bit_subnet _number.5-bit_host_number

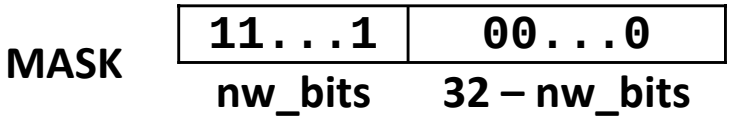
Internet Protocol version 4 (IPv4)

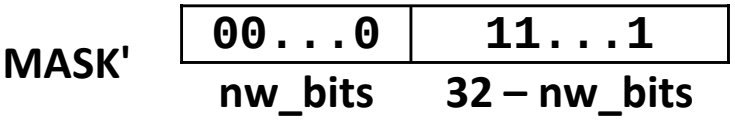
Reserved addresses

CIDR address block	Description
0.0.0.0/8	Current network (source address)
10.0.0.0/8	Private network
127.0.0.0/8	Loopback (destination = this node)
172.16.0.0/12	Private network
192.168.0.0/16	Private network
224.0.0.0/4	Multicast (Class D)
240.0.0.0/4	Reserved

Broadcast on CIDR network

IP address = octet1.octet2.octet3.octet4/nw_bits





Network number = MASK AND IP

Broadcast = MASK' OR IP

Internet Routing

Autonomous System (AS)

Nodes managed by one organization

Hierarchical routing

Interior Gateway Protocols (IGP)

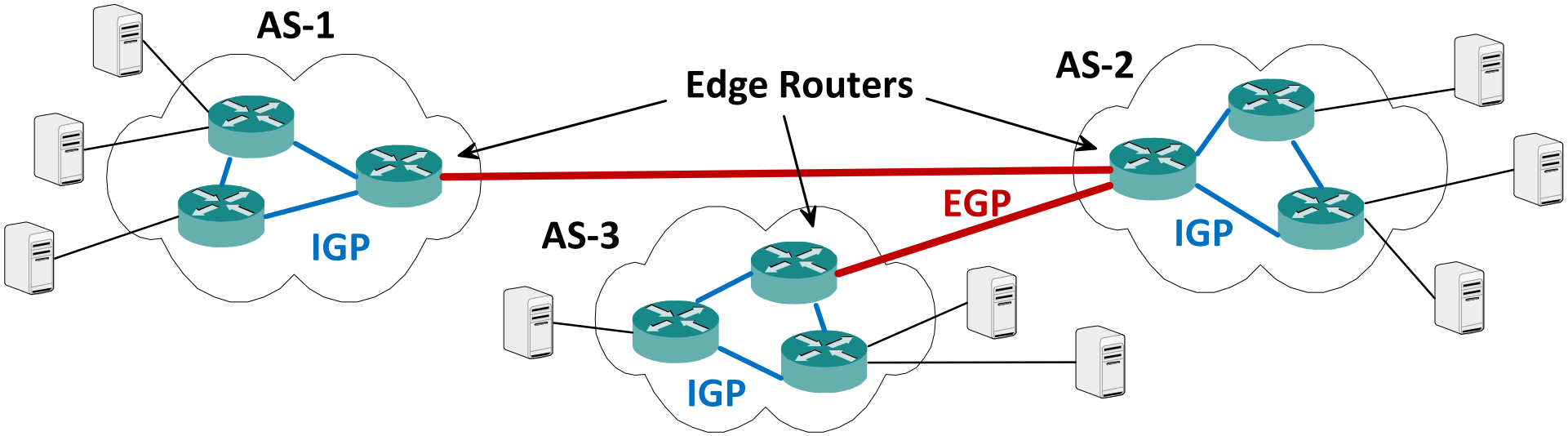
Routing protocols within one AS (Intra-AS)

Exterior Gateway Protocols (EGP)

Routing protocols between ASs (Inter-AS)

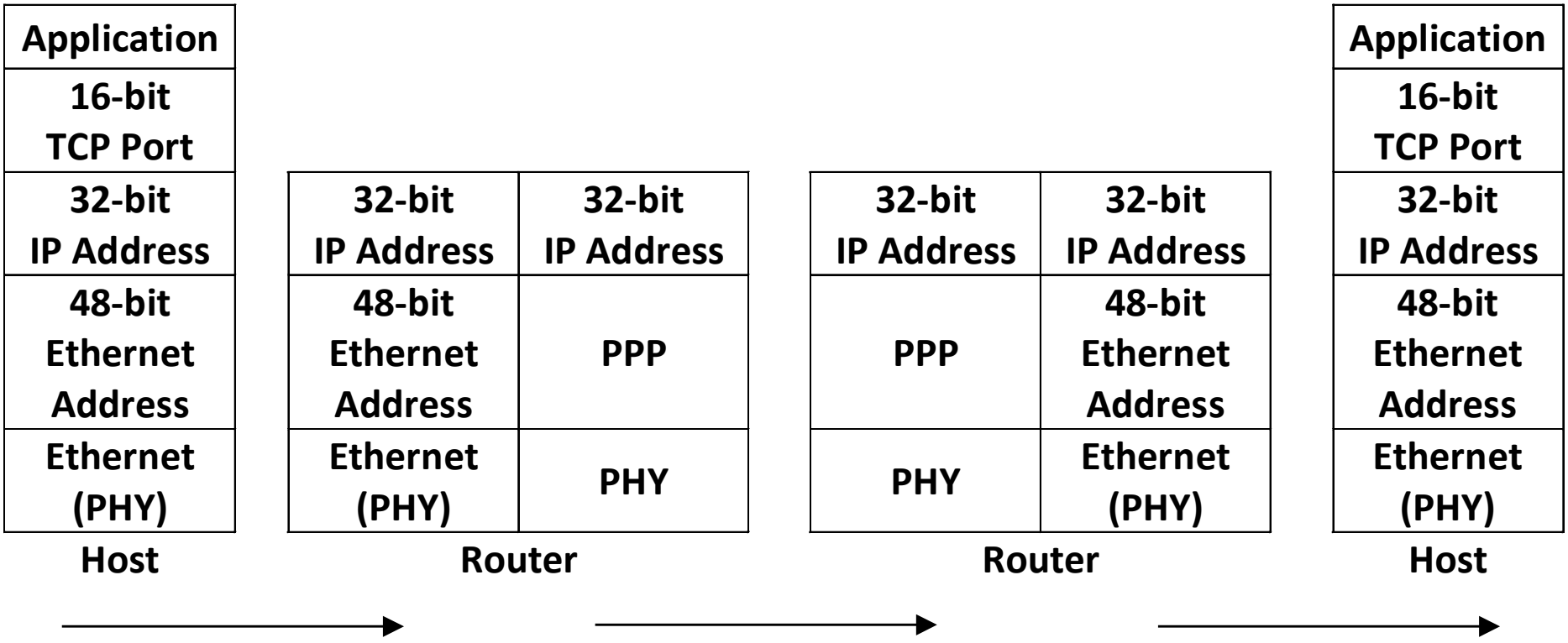
Edge router (gateway router)

Router within AS linked to router in different AS



Relationship of Protocol Layers

Typical network



Locate router by IP address
(uses default gateway)
Send to router by MAC address

Point-to-point

Locate host by IP address
Send to host by MAC address

Ethernet always uses source / destination Ethernet addresses — not IP addresses

Host finds MAC address for router using an **address resolution protocol (ARP)**

Network Address Translation (NAT)

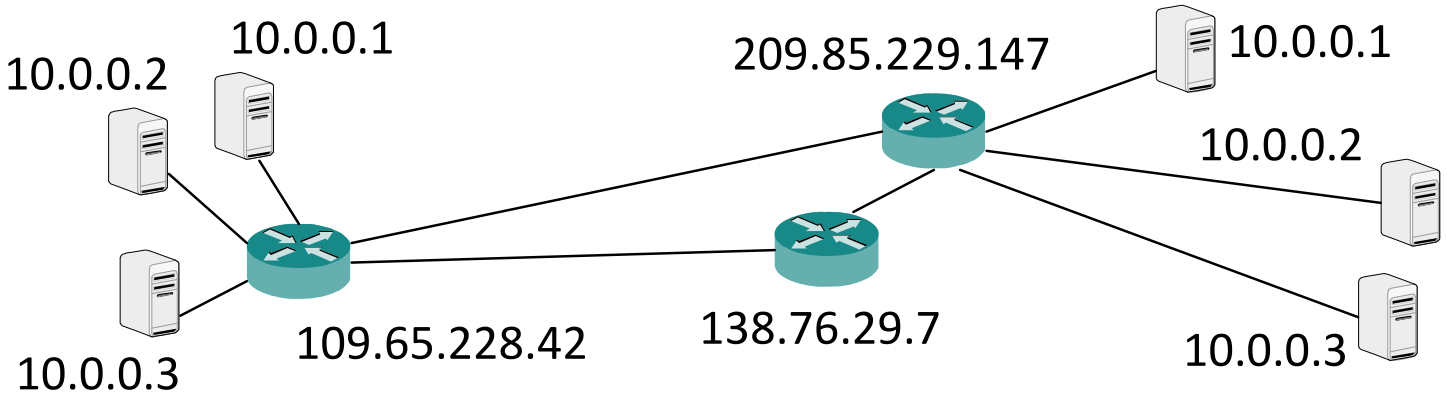
Router

- Receives IP datagram
- Exchanges IP source / destination address in header
- Forwards datagram
- Maintains list of translations

Autonomous systems

- Allocate private network addresses internally
  - 10.0.0.0 and 192.168.0.0
- Only gateway and backbone devices require unique IP addresses
  - Local nodes on AS re-use private addresses

Example



Network Address Translation (NAT)

Translation details

No translation required on traffic internal to AS

- Local addresses = real IP addresses

Local node to external node

- Outgoing packet
  - Source endpoint = local IP address + application source port number
  - Dest endpoint = remote IP address + well-known port number
- Gateway router NAT
  - Replaces
    - Local IP address with gateway IP address
    - Local application source port with unique unused NAT port
  - Records mapping
    - NAT source port → local IP address + source port number
  - Forwards packet

External node to local node

- Gateway router replaces NAT port → local IP + app source port

Network Address Translation (NAT)

Example

Host A HTTP request to Server S

Source port = **1025** Source IP = **10.0.0.2**

Destination port = 80 Destination IP = 209.85.229.147

Outgoing NAT at Gateway router B

Source port = **3745** Source IP = **109.65.228.42**

Destination port = 80 Destination IP = 209.85.229.147

Server S HTTP response to Host A

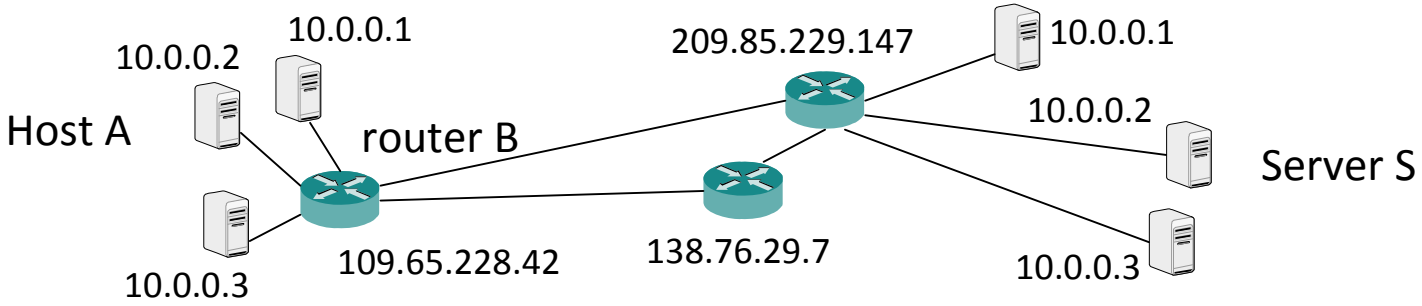
Source port = 80 Source IP = 209.85.229.147

Destination port = **3745** Destination IP = **109.65.228.42**

Incoming NAT at Gateway router B

Source port = 80 Source IP = 209.85.229.147

Destination port = **1025** Destination IP = **10.0.0.2**



Network Address Translation (NAT)

Traversal problem

Problem

How to operate service behind NAT

Solutions

Static router referral

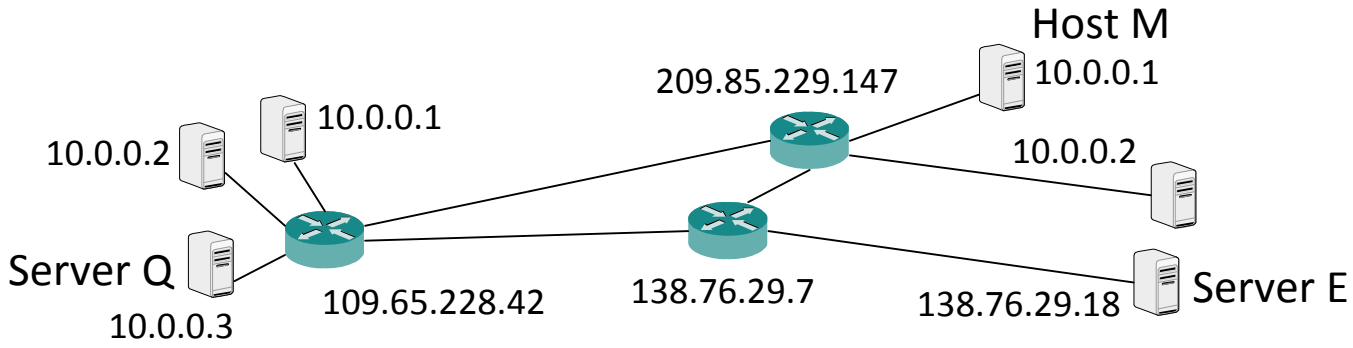
- Define service port on Server Q
- Refer all incoming traffic at gateway for service port to Server Q

Relay server

- Server Q connects to Server E with public address (no NAT)
- Host M requests service from Server Q via Server E
- Server Q replies via Server E

Universal Plug and Play (UPnP)

- Specialized protocols for NAT traversal

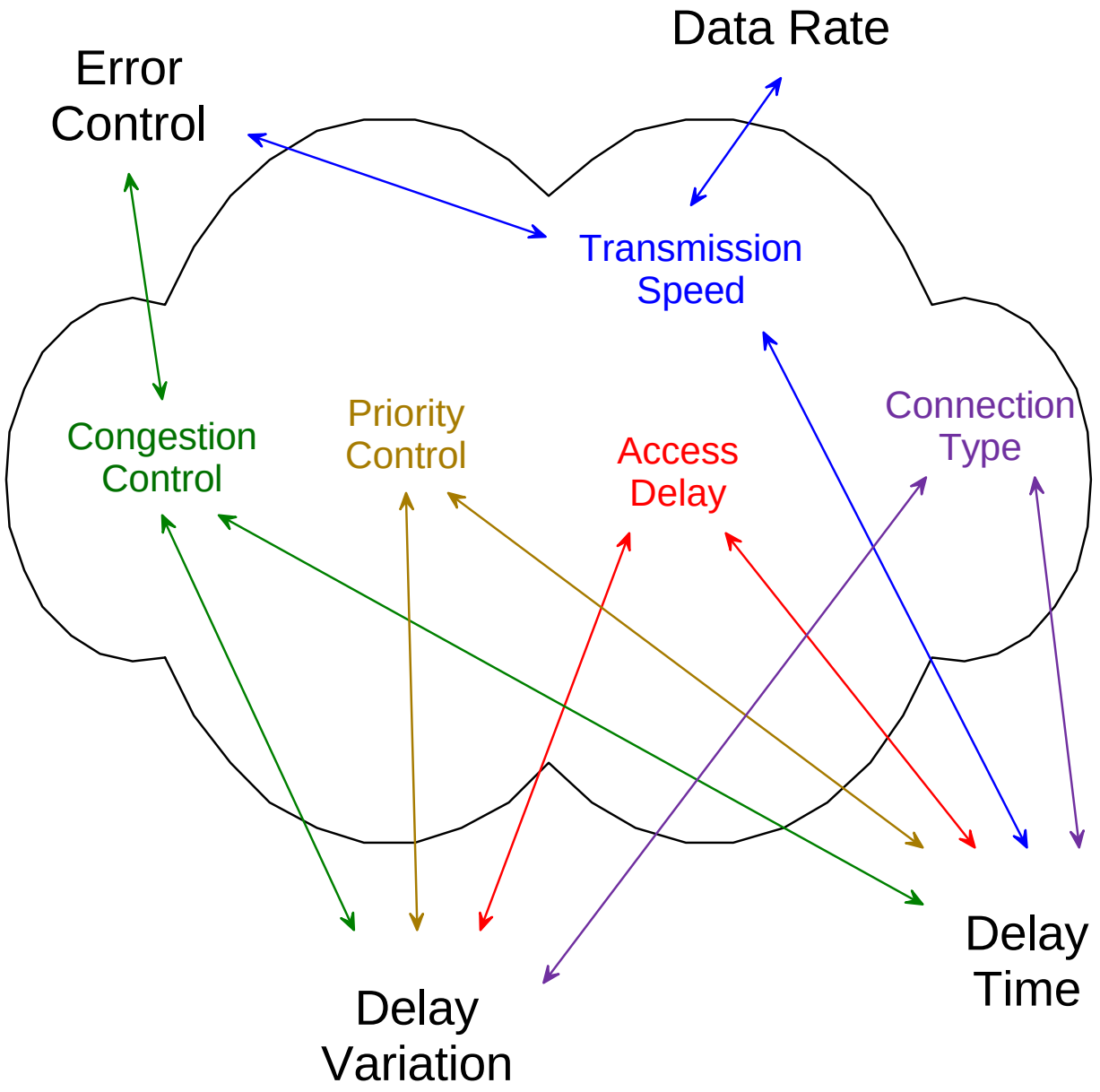


Quality of Service (QoS)

Network parameters

Data rate	Physical transmission speed in bps
Error control	Bit error rate (BER) $< 10^{-9}$ on fiber optic cable $< 10^{-3}$ on wireless Packet loss rate Depends on congestion control policy
Delay	End-to-end transmission time for one bit High data rate $\Rightarrow$ lower transmission delay T_{trans} Congestion / priority $\Rightarrow$ longer queuing delay T_Q Datagram service $\Rightarrow$ longer processing delay T_{proc}
Delay variation	Dynamic variations in Data rate / propagation delay (jitter) ΔT_{trans} Queuing delay ΔT_Q Processing delay ΔT_{proc}

QoS and Network Policies



QoS Requirements for Various Services

Service	Speed	Error Control	Delay	Delay Variation
e-mail	—	good	—	—
file transfer	reasonable	maximum	reasonable	reasonable
NFS				
database access				
voice	fast	good	very small	very small
video	very fast	good	very small	very small
real time control				

Internet Transport Layer Connections

Reliable transport (TCP)

- Connection-oriented
  - TCP connection established before data transfer
- Error-free delivery
  - Data delivered
    - In original order
    - No errors, duplications, omissions
- Flow control
  - Control sender rate to prevent buffer overflow in receiver
- Congestion control
  - Control sender rate to prevent buffer overflow in network

Unreliable transport (UDP)

- Connectionless
- Lower overhead ⇒ faster but no guarantees
- Segments with errors discarded with no warning to application

Transport Layer — QoS Trade-Off

	Reliable Transport	Unreliable Transport
Protocol	TCP	UDP
Error control	Error-free	No error correction
Connection	Connection-oriented	Connectionless
Socket type	Stream	Datagram
Processing overhead	Error correction, packet ordering, congestion control, session state management	None

Reliable Transport

Generally preferred when possible

Unreliable Transport

Used when

- Some data loss tolerable
- Delay or jitter intolerable
- Example — video delivery

Source / Destination Ports

Client

Opens socket to send requests

Client / OS binds port number to socket

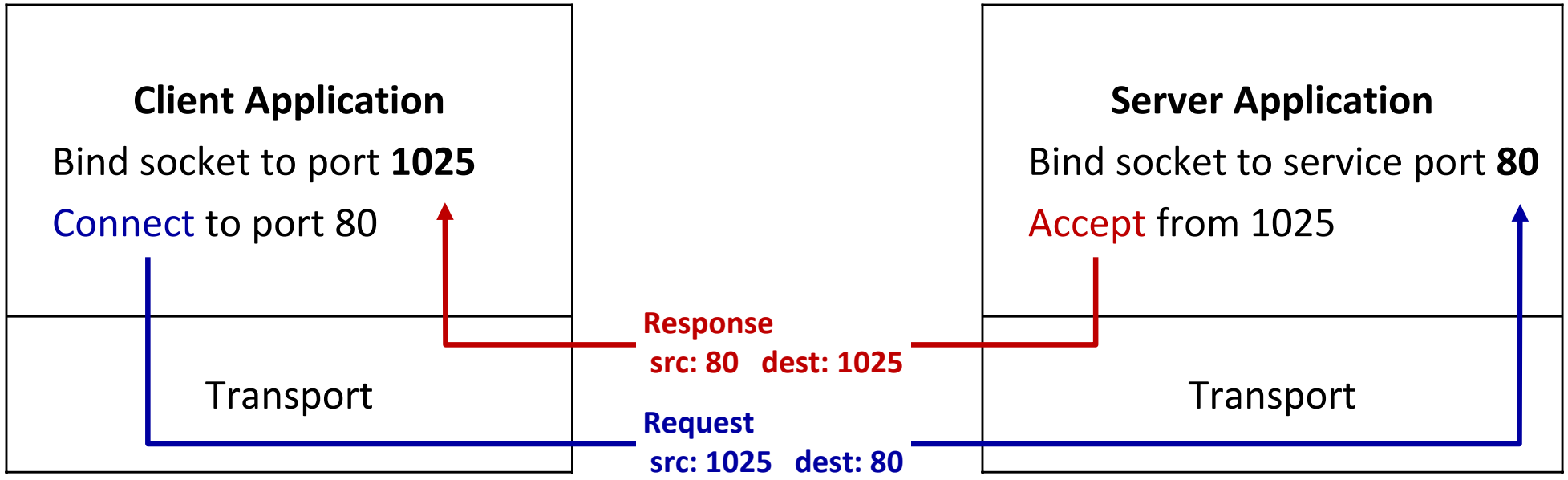
1024 ≤ client port ≤ 65,535 identifies client application

Server

Opens listen socket mapped to accept sockets for requests

Binds well-known port to service socket

0 ≤ well-known port ≤ 1023 identifies service application



Multiplexing / Demultiplexing

Applications send / receive data on sockets

Multiple sockets ⇒ multiple conversations

Transport layer segment

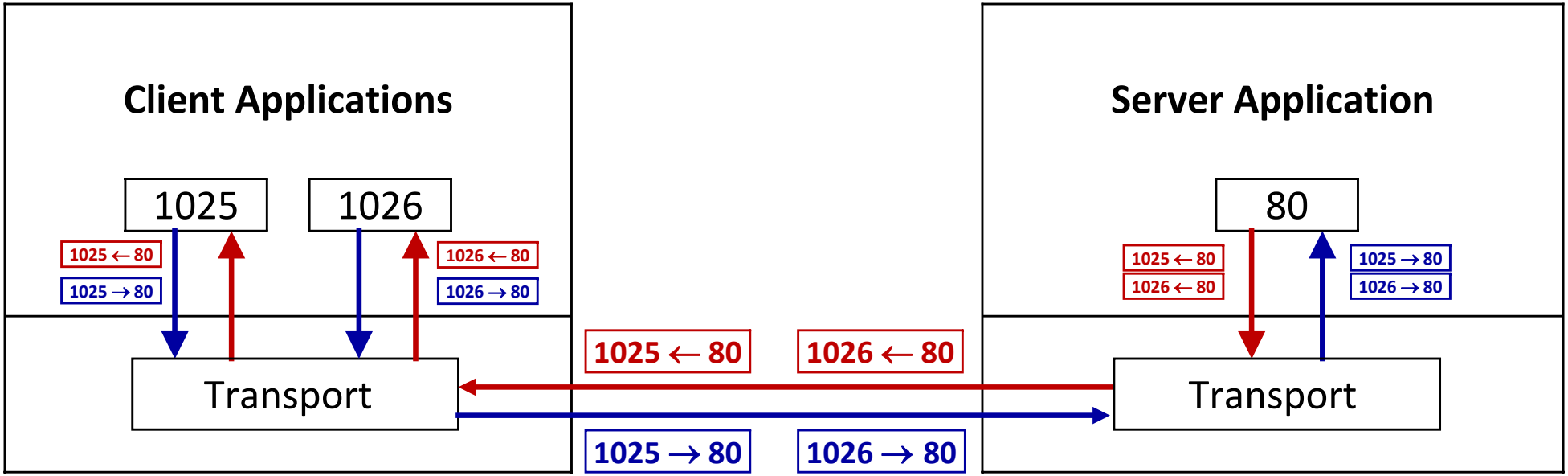
Transport header + application data (PDU)

TCP / UDP headers carry source + destination ports

Multiplexing / demultiplexing

Segments transmitted on same infrastructure

Sorted by destination port at destination



User Datagram Protocol (UDP)

Internet unreliable transport protocol

Defined in RFC 768

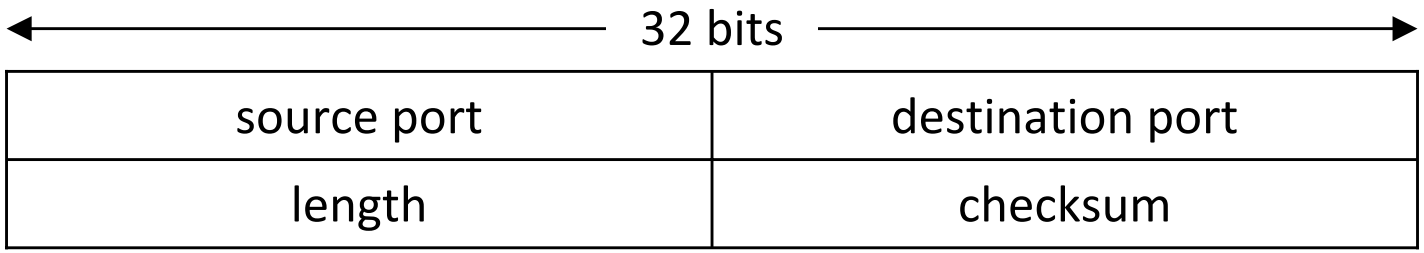
Used when low delay / jitter more important than error control

Streaming multimedia, multiplayer games, ...

UDP segment



UDP header

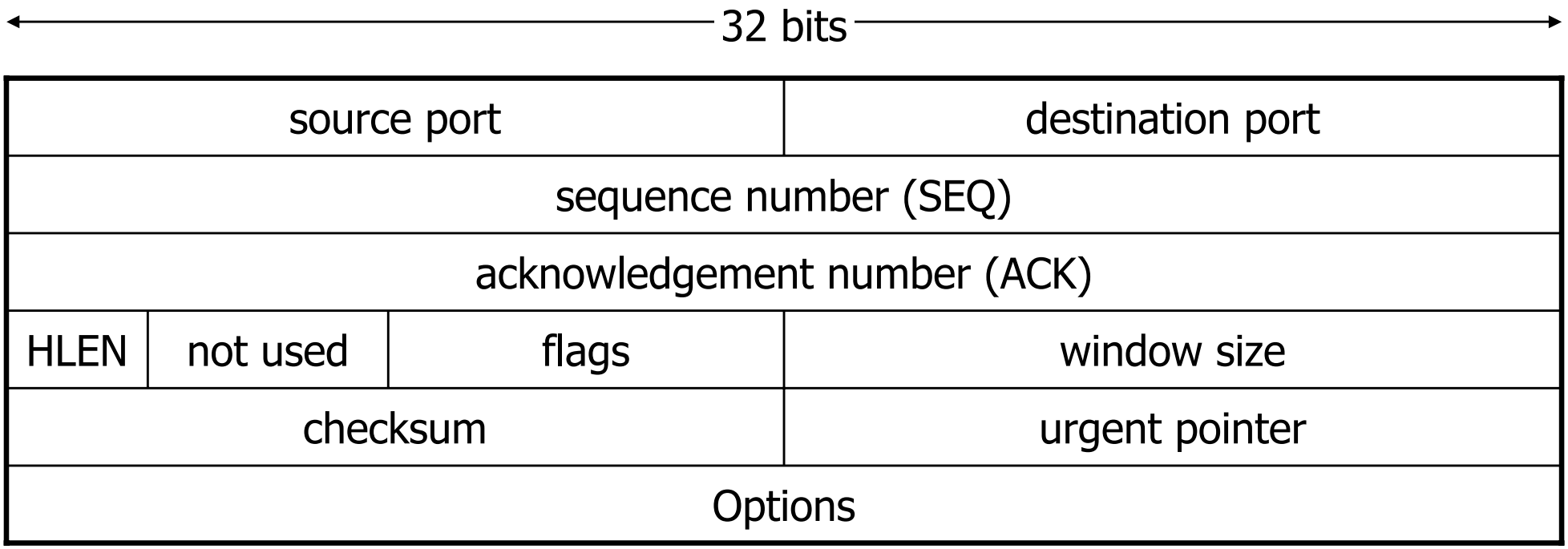


Length

Number of bytes in UDP segment < 2^{16} = 65,536

Maximum length of data = 64 KB – lengths of all headers

TCP Header



HLEN (data offset)	4 bits	Length of TCP header in 32-bit words
Reserved	3 bits	Not used
Flags	9 bits	Control bits
Window size	16 bits	Number of bytes receiver can receive now
Urgent pointer	16 bits	Offset from SEQ points to last urgent data byte
Options	10 – 320 bits	Options fields + padding for multiple of 32 bits

TCP Header
Flags

NS	ECN-nonce concealment protection
CWR	Congestion Window Reduced (CWR) flag Sender indicates receiving segment with ECE flag set
ECE	ECN-Echo If SYN = 1 peer is ECN capable If SYN = 0 packet with Congestion Experienced flag in IP header received during normal transmission
URG	Urgent pointer field valid
ACK	**Acknowledgment field valid**
PSH	Push buffered data to receiving application
RST	**Reset connection**
SYN	**Synchronize sequence numbers**
FIN	**No more data from sender**

TCP Connection Set-up
Three-way handshake

Client SYN segment

SYN flag = 1

SEQ = random number x

No data

Server SYN-ACK segment

SYN flag = ACK flag = 1

SEQ = random number y

ACK = x + 1

No data

Client ACK segment

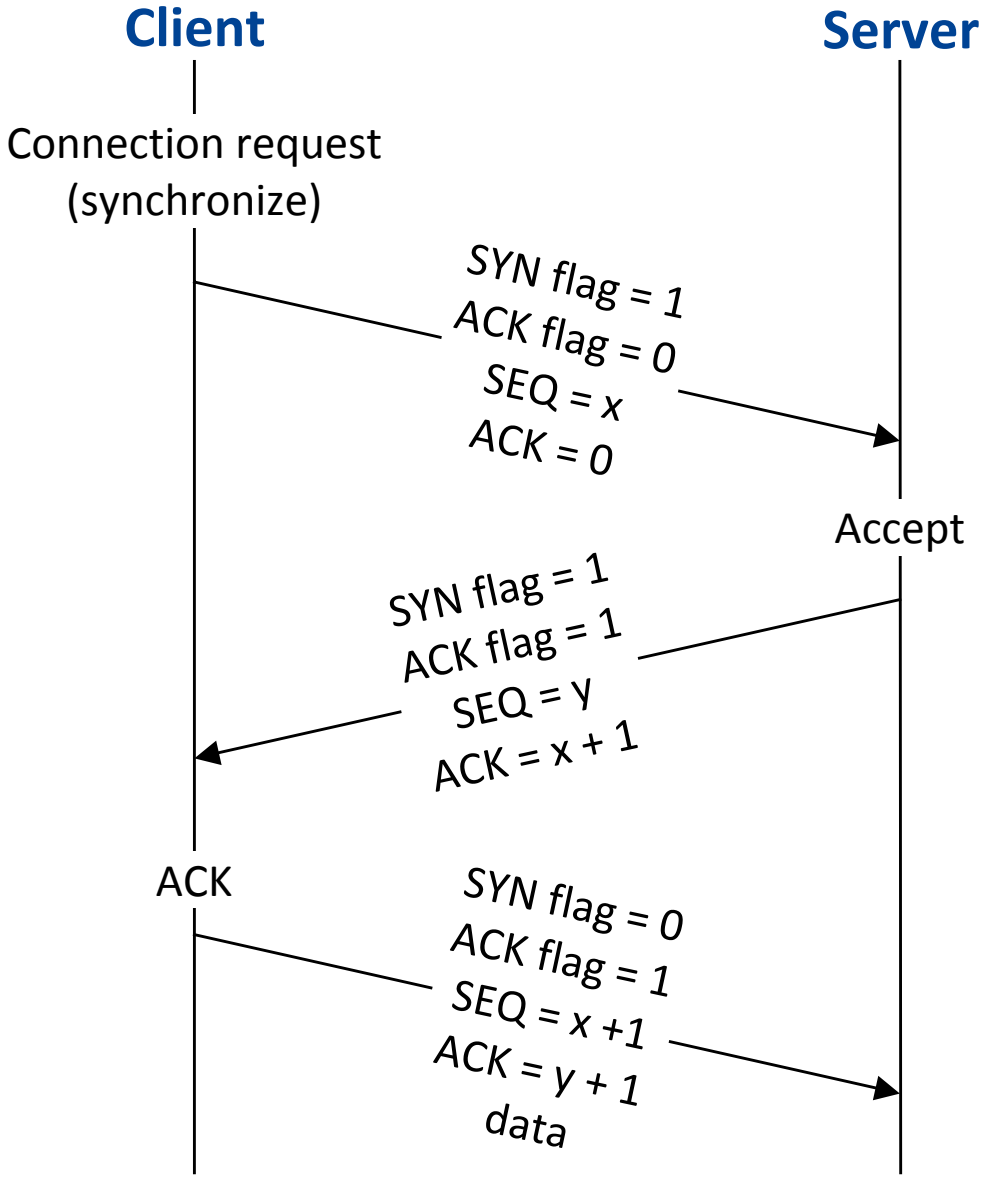
SYN flag = 0

ACK flag = 1

SEQ = random number x + 1

ACK = y + 1

May contain data



TCP SEQ + ACK

Initial sequence number (ISN)

Random SEQ in 3-way handshake

Prevents counterfeit segments

At end of handshake SEQ = ISN + 1

SEQ — byte sequencing

SEQ = previous SEQ + length(data bytes in previous segment)

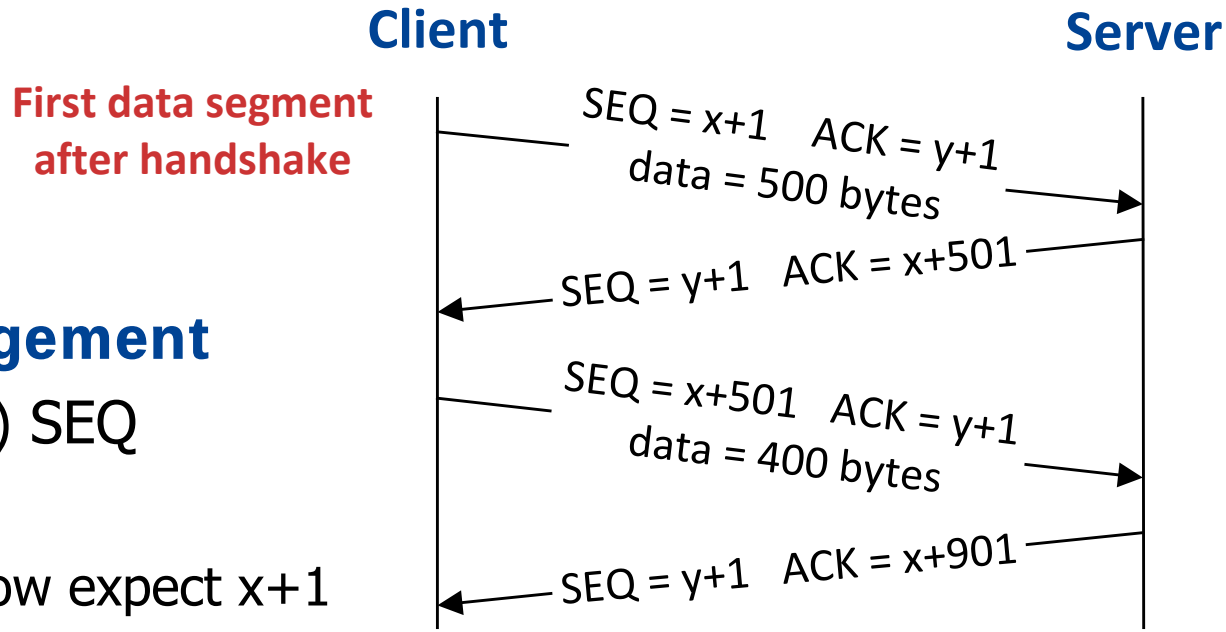
= ISN + 1 + data bytes sent in all **previous** segments

ACK — byte acknowledgement

ACK = next (expected) SEQ

ACK = x + 1

Acknowledge x — now expect x+1



Send and Receive Windows

Send buffer

Holds sent segments until ACKed

On timeout resend segments from send buffer

Send window

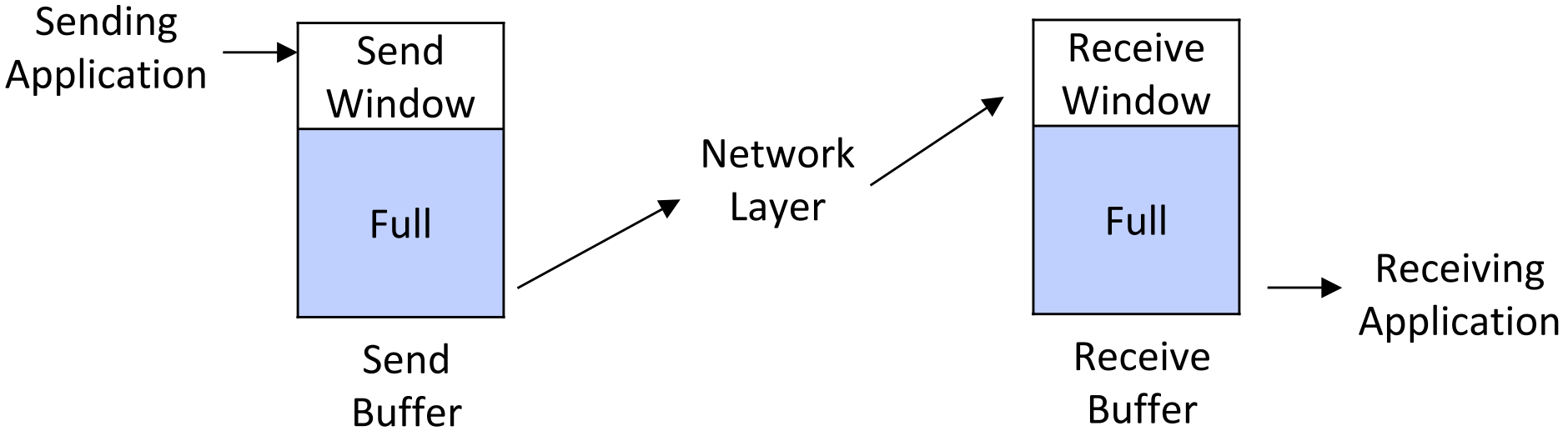
Available space in send buffer

Receive buffer

Holds received segments until requested by application

Receive window

Available space in receive buffer



Simplified TCP Sender — 1

```
// initialize
    SEQ = ISN + 1
    SendBase = ISN + 1                  // last byte ACKed
    InFlight = 0                        // unACKed bytes sent
    RTO                                 // timeout interval
    SendWindow, RecvWindow              // send & receive windows

// main loop
    if (new data from application)
        Prepare data segment
            sequence number for segment = SEQ
            SEQ = SEQ + length(data)
        if InFlight < min{SendWindow,RecvWindow)
            Pass segment to IP
            InFlight = InFlight + length(data)
            if !(timer running) timer = RTO
```

Simplified TCP Sender — 2

```
if (receive ACK = y)
    stop timer
    if (y > SendBase)
        newACKs = y – SendBase          // bytes ACKed
        SendBase = y
        InFlight = InFlight – newACKs
        if (InFlight > 0) timer = RTO
if (timeout)
        SEQ = SendBase = min{unACKed SEQ}
        resend
        timer = RTO
```

Simplified TCP Receiver — 1

```
// initialize
    Set RecvWindow = receiver buffer size
    expected = Sender ISN + 1
    ack_buffer = 0          // received unACKed segments
    ack_max                 // delayed ACK trigger
    ack_delay = 250 msec    // local policy: < 500 msec
    Start ACK delay timer = ack_delay

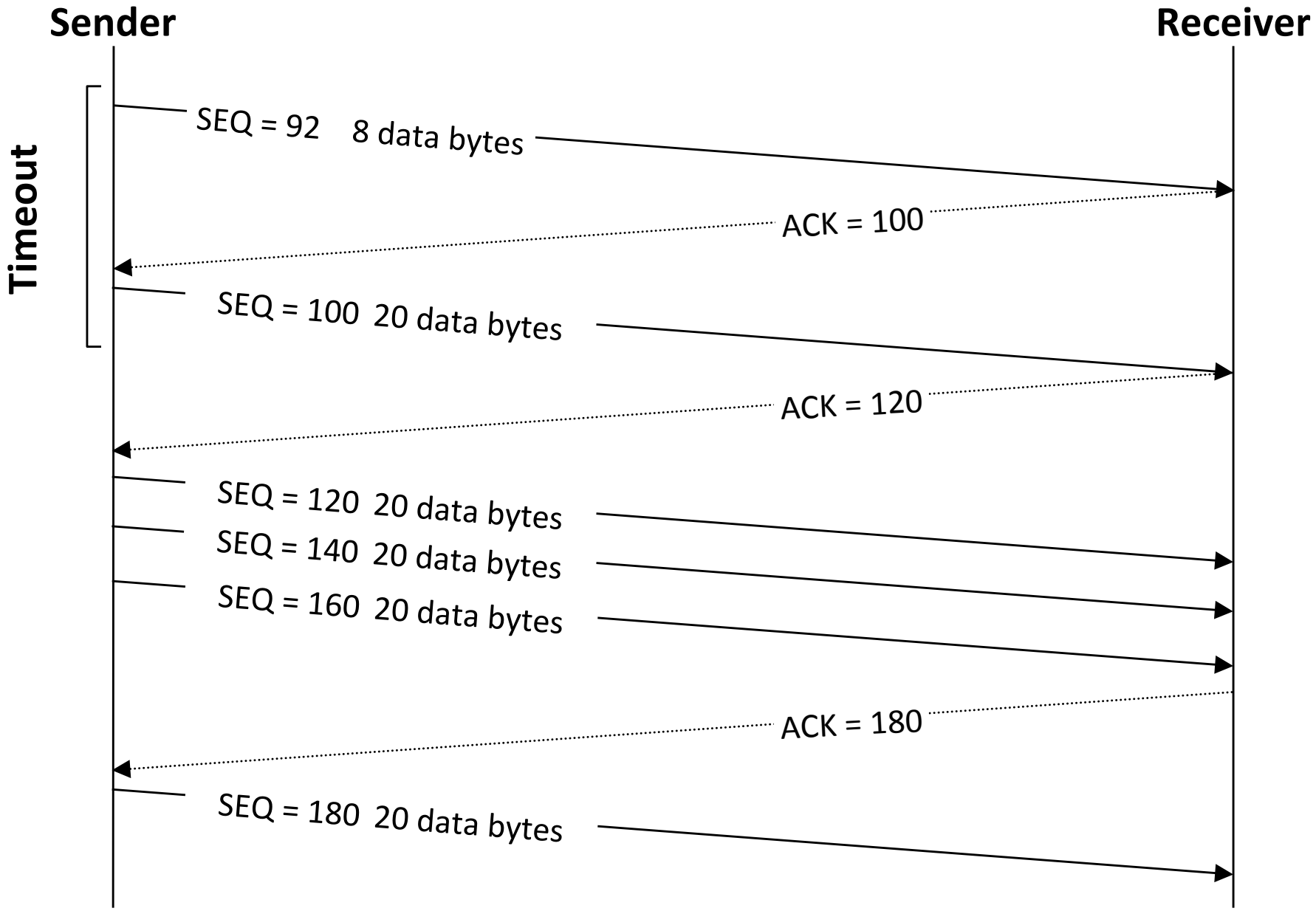
if (ACK delay timer = 0 && ack_buffer > 0)
    Send ACK = expected with updated RecvWindow
    ACK delay timer = ack_delay
    ack_buffer = 0
```

Simplified TCP Receiver — 2

```
if (receive SEQ = x)
    if (x = expected && error-free)
        expected = expected + length(data)
        if (NACK = 1)
            Send ACK = expected with updated RecvWindow
            ACK delay timer = ack_delay
            ack_buffer = 0
            NACK = 0
        else if (ack_buffer < ack_max)
            nextACK = expected
            ack_buffer++
        else if (ack_buffer = ack_max)
            Send ACK = expected with updated RecvWindow
            ACK delay timer = ack_delay
            ack_buffer = 0
    else
        Send ACK = expected with updated RecvWindow
        ACK delay timer = ack_delay
        NACK = 1
```

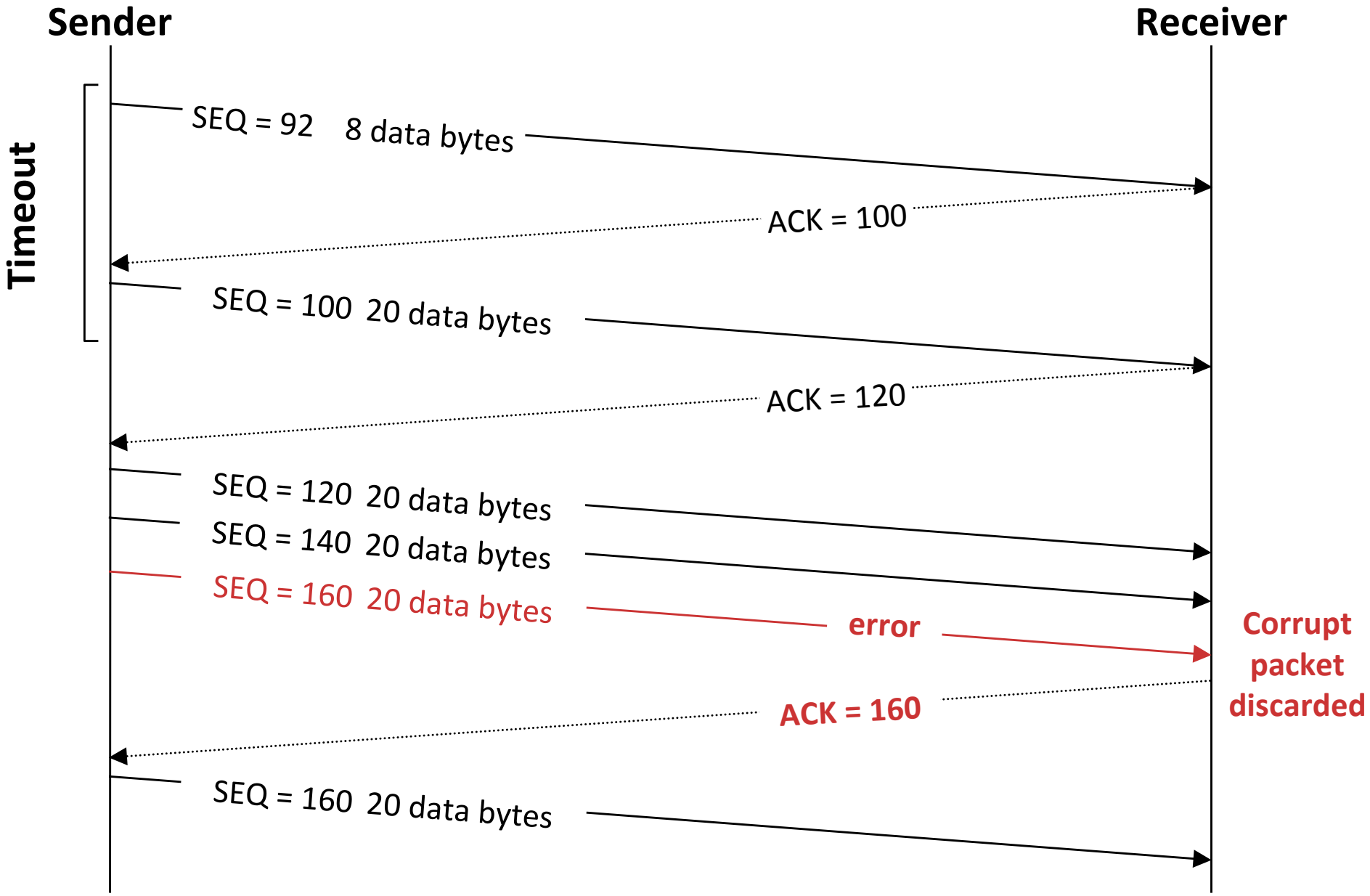
TCP SEQ + ACK

No errors



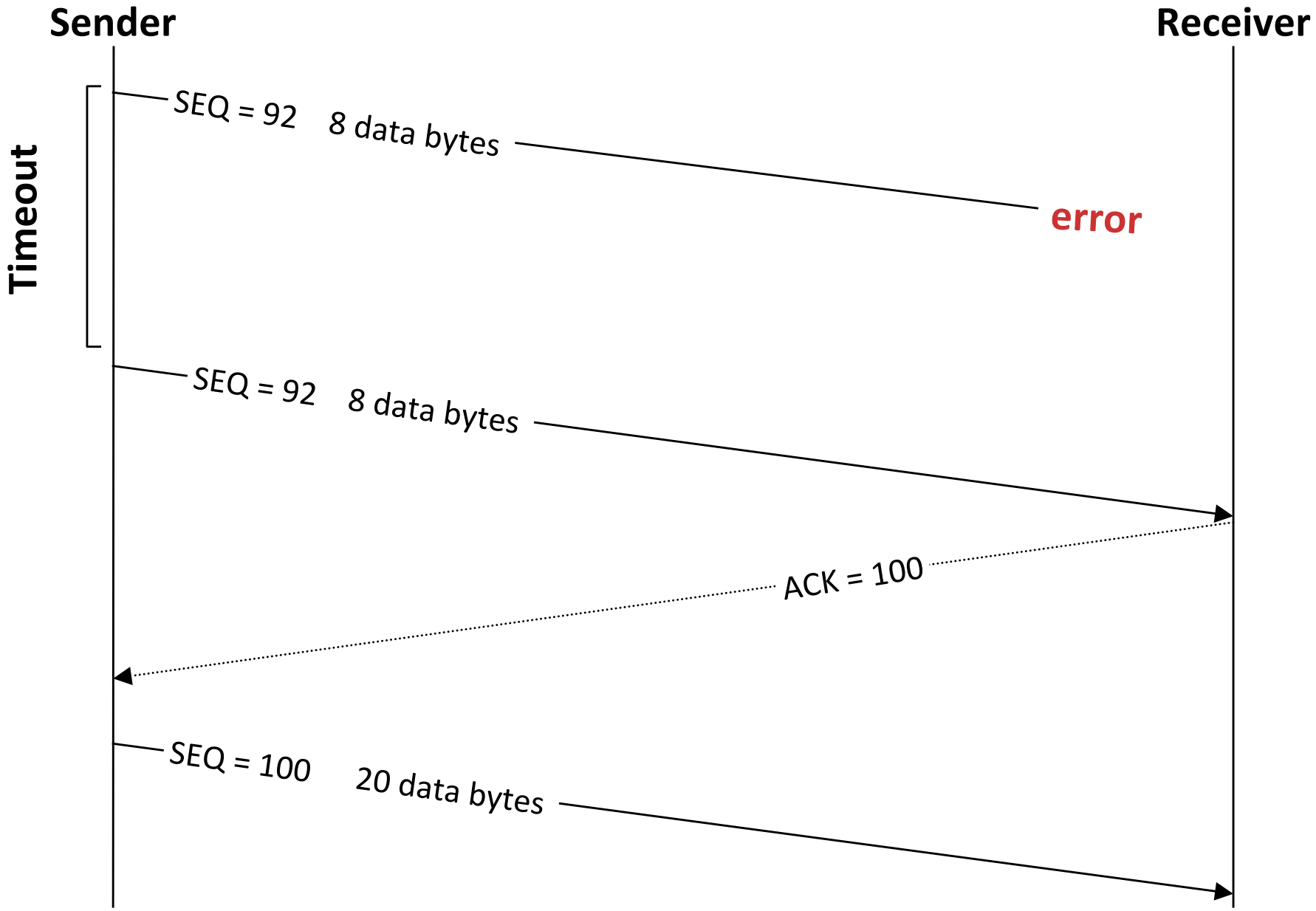
TCP SEQ + ACK

Bit errors



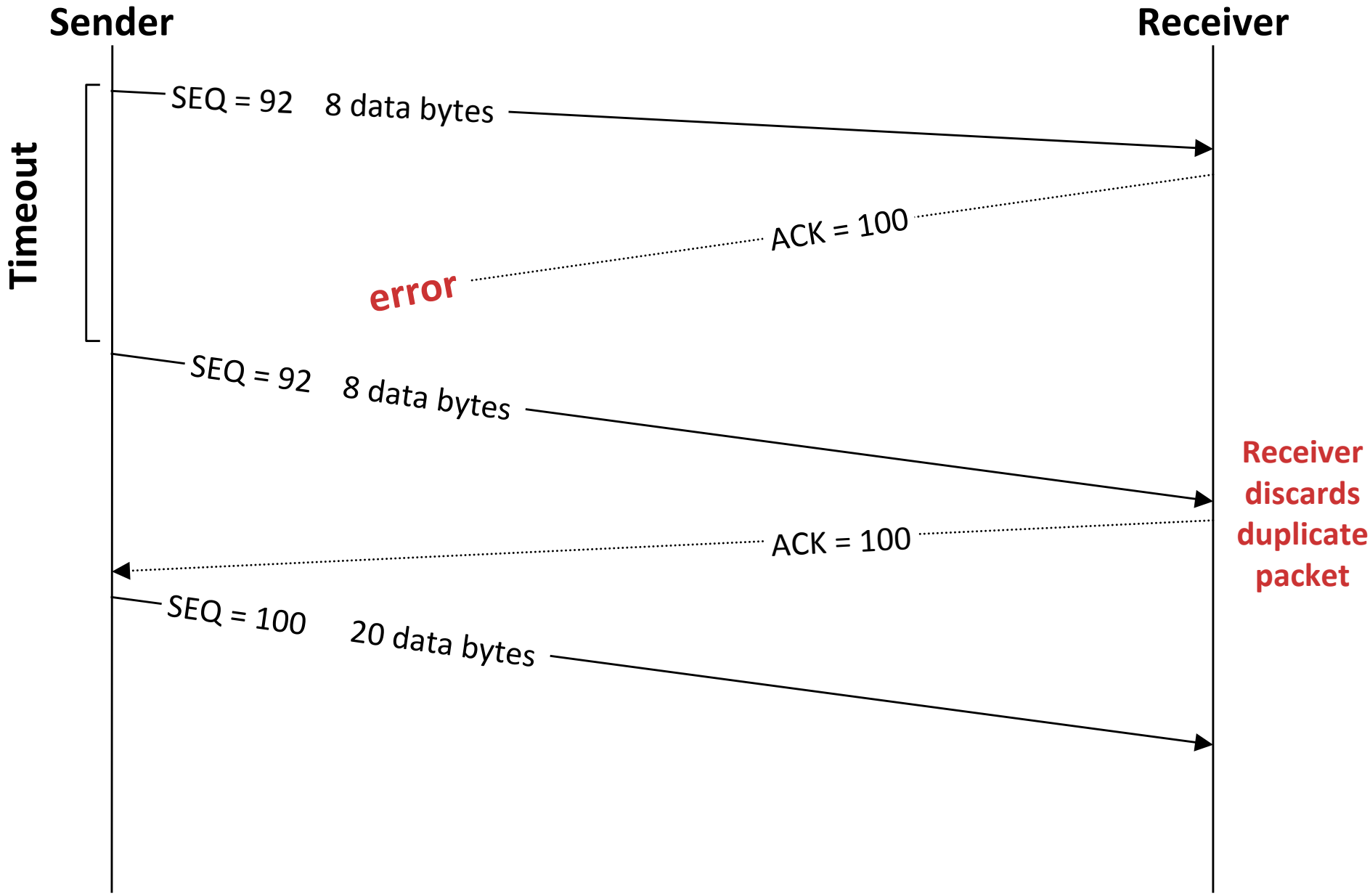
TCP SEQ + ACK

Timeout



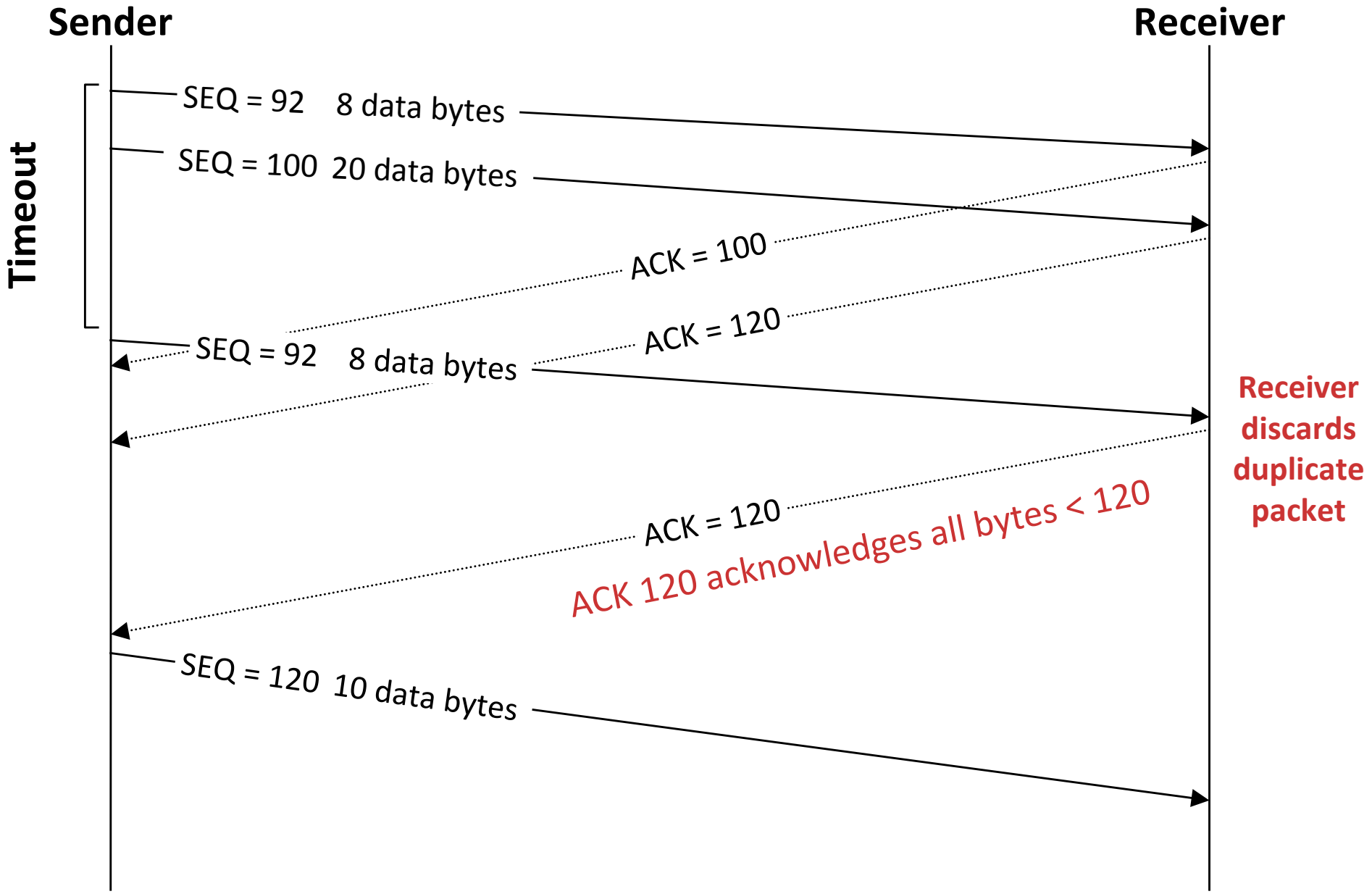
TCP SEQ + ACK

Lost ACK



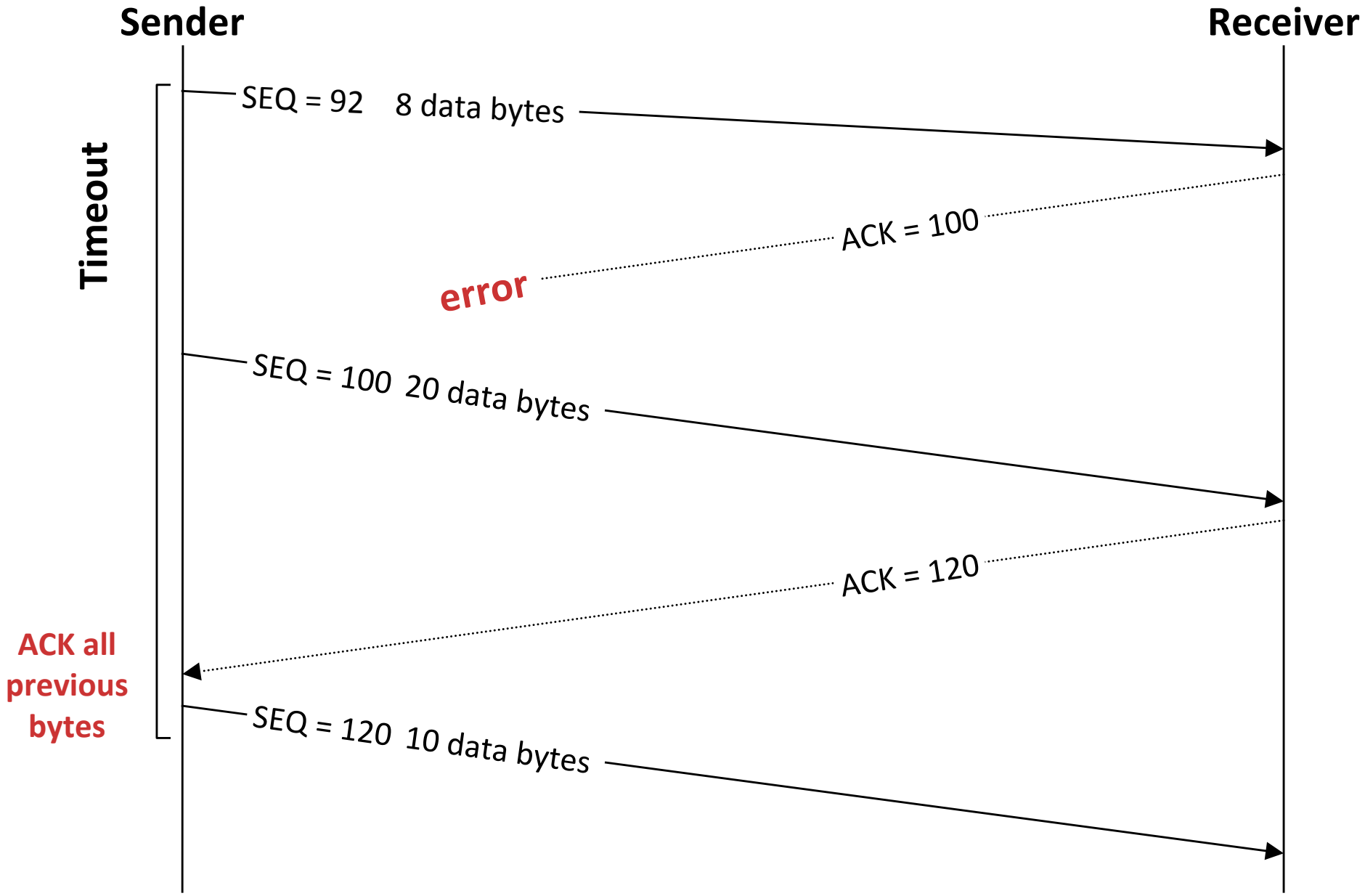
TCP SEQ + ACK

Missed ACK



TCP SEQ + ACK

Cumulative ACK



TCP Connection Close

Symmetric

Client or server may close connection

FIN segment

SYN flag = 1

SEQ = cumulative SEQ number

ACK segment

ACK flag = 1

ACK = SEQ + 1

FIN segment

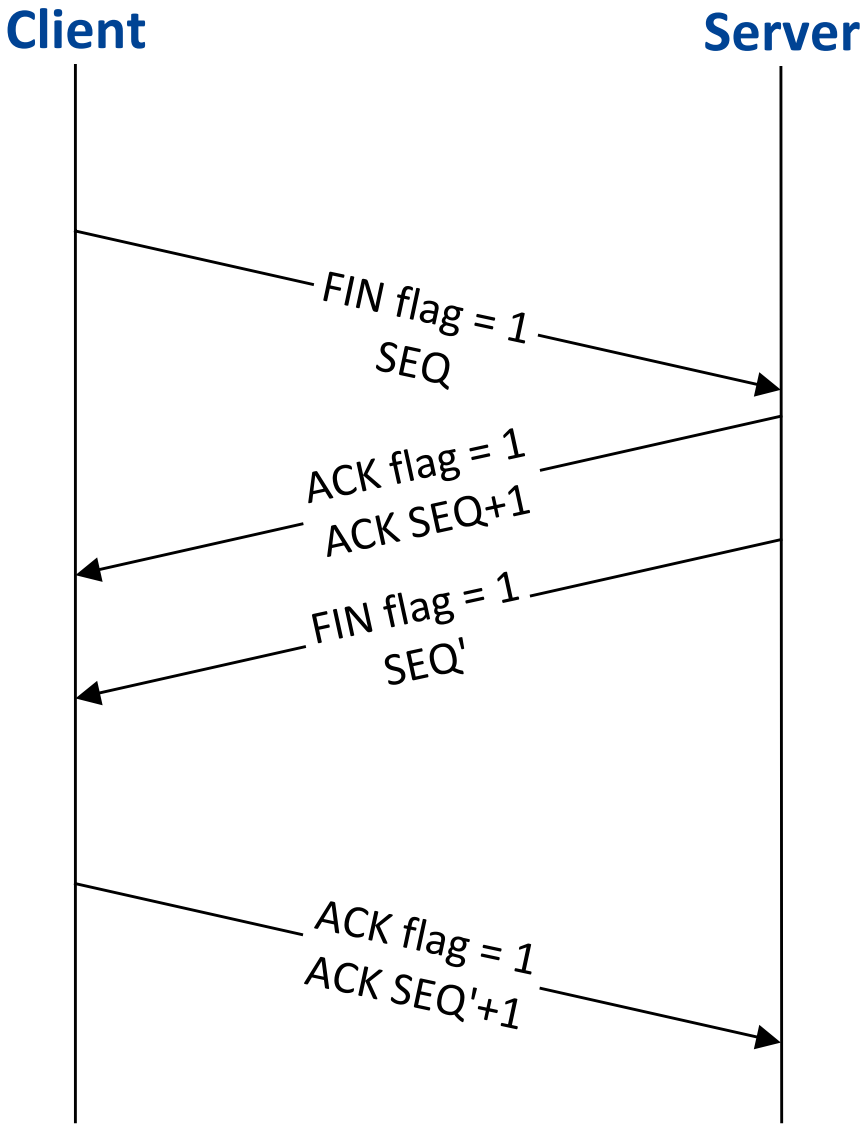
FIN flag = 1

SEQ' = cumulative SEQ number

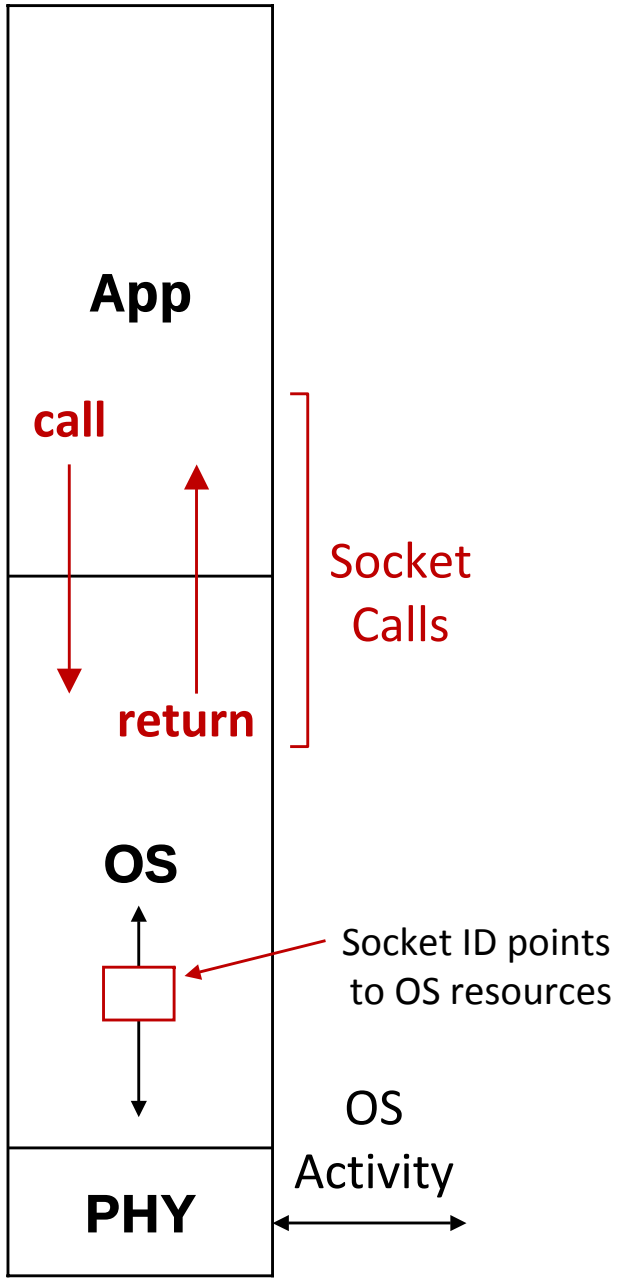
ACK segment

ACK flag = 1

ACK = SEQ' + 1



Socket Calls — Between App and OS



Socket, Bind

OS creates resources for network connection

OS returns to App socket descriptor (socket ID)

Listen

Server OS makes service available

Connect

Client OS attempts connection to service

Accept

Server OS creates new connection socket

Server OS returns new descriptor to App

Listen socket continues to listen

Send

App sends data to OS, pointed at socket

OS transmits data on associated connection

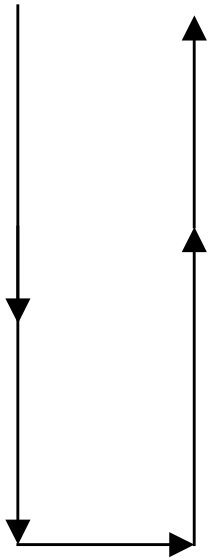
Receive

App requests data from OS socket buffer

OS returns data to App

Using UDP

Client	Server
Application Open socket Send data on socket to endpoint (node address + service port)	**Application** Open socket Bind service port Listen on socket Receive data from listen socket
Client UDP Agent Accept data Add header with checksum Send to server	**Server UDP Agent** Perform checksum Error Discard segment No error Pass data to socket by port



If required — add reliability features at client / server application level

Connection

Client **Server**

```
s_ID = socket(domain, type, protocol)
```
domain: address type; type: service type

```
bind(s_ID, s_endpoint, s_endpoint_len)
listen(s_ID, backlog)
```

connection set-up →

```
c_ID = socket(domain, type, protocol)
bind(c_ID, c_endpoint, c_endpoint_len)
connect(c_ID, service_endpoint)
```

If client skips **bind()**
OS sets a default port

```
s = accept(s_ID, c_endpoint, endpoint_len)
```

← data

```
send(s, data, len, flags)
receive(c_ID, buffer, len, flags)
```

Perl Server Socket Example

```
#!/usr/bin/perl
$flag = "1";
use IO::Socket;
$sock = IO::Socket::INET->new(
        LocalHost => '127.0.0.1',
        LocalPort => 1234,
        Listen => 1,
        Reuse => 1,
        Proto => 'tcp') || die "Error creating socket\n";
$client = $sock->accept();
while($flag == "1") {
  $line = <$client>;
  print $line;
  print $client "Received\n";
  if ($line =~ /bye/) {$flag = "0";}
}
close($sock);
```

In CLI run perl script

```
$perl servlet.pl
```

In second CLI enter:

```
$telnet 127.0.0.1 1234
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
```

Server

- Prints data from client Echoes `Received`
- Closes on data `bye`

LOOPBACK ADDRESS (for testing)

`127.0.0.1`

Local calls on this machine

Assistive Application Protocols used with IP Layer

Domain Name System (DNS)

- Convert between node name and network address

Address Resolution Protocol (ARP)

- Convert between network and hardware addresses
- Discover local subnet topology

Dynamic Host Configuration Protocol (DHCP)

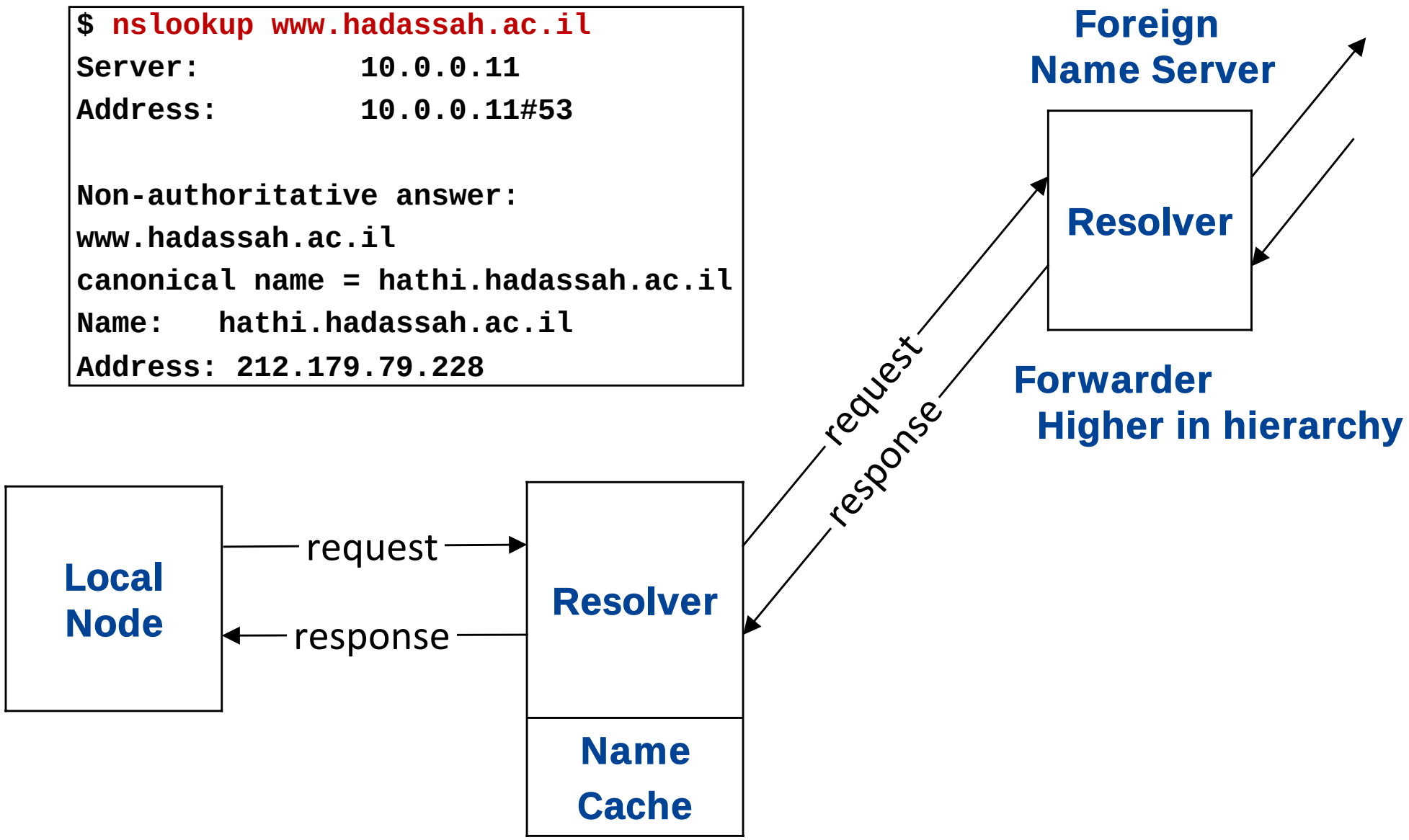
- IP address allocation on request from DHCP server
  - Server typically in gateway router
- Server allocates IP address from pool of available addresses
- DHCP message types
  - DHCP server discovery (host broadcast)
  - DHCP offer (server response)
  - DHCP request (host request)
  - DHCP ACK (server provides address)

Domain Name System (DNS)

Address Resolution Hierarchy

```
$ nslookup www.hadassah.ac.il
Server:         10.0.0.11
Address:        10.0.0.11#53

Non-authoritative answer:
www.hadassah.ac.il
canonical name = hathi.hadassah.ac.il
Name:   hathi.hadassah.ac.il
Address: 212.179.79.228
```



Address Resolution Protocol (ARP)

Look-up MAC address by IP address (RFC 826)

```
Q:  Who has IP = a.b.c.d ?  (MAC layer broadcast)
A:  I am IP = a.b.c.d with MAC = u:v:w:x:y:z
```

STAs store mappings in arp table

ARP packet fields

HTYPE	Hardware type — MAC protocol
PTYPE	Protocol type — network protocol
HLEN	Hardware length — length in octets of MAC address
PLEN	Protocol length — length in octets of network address
Operation	1= request / 2 = reply
SHA	Sender hardware address
SPA	Sender protocol address
THA	Target hardware address (ignored in requests)
TPA	Target protocol address

Internet Control Message Protocol (ICMP)

Control protocol

Network management information

Error reporting

Unreachable host / network / port / protocol

Echo request / reply

Packets carried in IP datagrams

Packet structure

Offset	0 – 7	8 – 15	16 – 23	24 – 31
0	Type	Code (subtype)	Checksum	
32	Rest of Header			
64	Padding			

Rest of Header — message specific

Padding — data field

Selected Message Types		
Type	Code	description
0	0	echo reply
3	0	dest network unreachable
	1	dest host unreachable
	2	dest protocol unreachable
	3	dest port unreachable
	6	dest network unknown
	7	dest host unknown
8	0	echo request
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Ping

Client application program

- Client sends echo request packets to destination IP address
  - Type = 8 Code = 0 16-bit ID number 16-bit sequence number
- OS dependent
  - Linux ping
    - Default = continuous packets with 56 padding bytes
  - Windows ping.exe
    - Default = 4 packets with 32 padding bytes

Ping server (OS service)

- Responds to each echo request packet with echo rely packet
  - Type = 0 Code = 0 16-bit ID number 16-bit sequence number

Ping client

- Measures round trip time (RTT)
- Reports packet losses, RTTs and average RTT

UDP ping

- Uses UDP instead of ICMP

Ping Example

```
c:\>ping www.hadassah.ac.il

Pinging hathi.hadassah.ac.il [212.179.79.228] with 32 bytes of data:

Reply from 212.179.79.228: bytes=32 time=32ms TTL=119
Reply from 212.179.79.228: bytes=32 time=70ms TTL=119
Reply from 212.179.79.228: bytes=32 time=135ms TTL=119
Reply from 212.179.79.228: bytes=32 time=83ms TTL=119

Ping statistics for 212.179.79.228:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 32ms, Maximum = 135ms, Average = 80ms
```

Traceroute

Client application program

- Multiple echo request packets to destination IP address
  - First packet TTL = 1
  - Each subsequent packet has TTL++
- OS dependent
  - Linux traceroute / tracepath
    - UDP to unlikely port number (port typically not defined)
  - Windows tracert.exe
    - ICMP echo request packets

Intermediate routers

```
if (--TTL == 0){delete ; ICMP type 11 to client}
```

- ICMP packet carries router name + IP address
- ICMP sequence = list of intermediate routers (hops = 1, 2, …)

Destination node

- UDP segment — ICMP port undefined (type 3 code 3) message
- ICMP packet — ICMP echo reply message

Traceroute Example

```
c:\>tracert www.hadassah.ac.il

Tracing route to hathi.hadassah.ac.il [212.179.79.228]
over a maximum of 30 hops:

  1     2 ms     1 ms     2 ms  10.0.0.138
  2    18 ms    19 ms    17 ms  bzq-179-37-1.static.bezeqint.net [212.179.37.1]
  3    18 ms    17 ms    17 ms  bzq-179-48-201.static.bezeqint.net [212.179.48.201]
  4    17 ms    17 ms    17 ms  bzq-179-80-241.static.bezeqint.net [212.179.80.241]
  5    17 ms    18 ms    17 ms  bzq-179-124-201.static.bezeqint.net [212.179.124.201]
  6    17 ms    17 ms    17 ms  bzq-179-124-138.static.bezeqint.net [212.179.124.138]
  7    19 ms    18 ms    18 ms  bzq-179-59-1.static.bezeqint.net [212.179.59.1]
  8    32 ms    36 ms    41 ms  10.20.110.18
  9   105 ms    84 ms    71 ms  bzq-179-79-228.static.bezeqint.net [212.179.79.228]
 10    73 ms    88 ms   150 ms  bzq-179-79-228.static.bezeqint.net [212.179.79.228]

Trace complete.

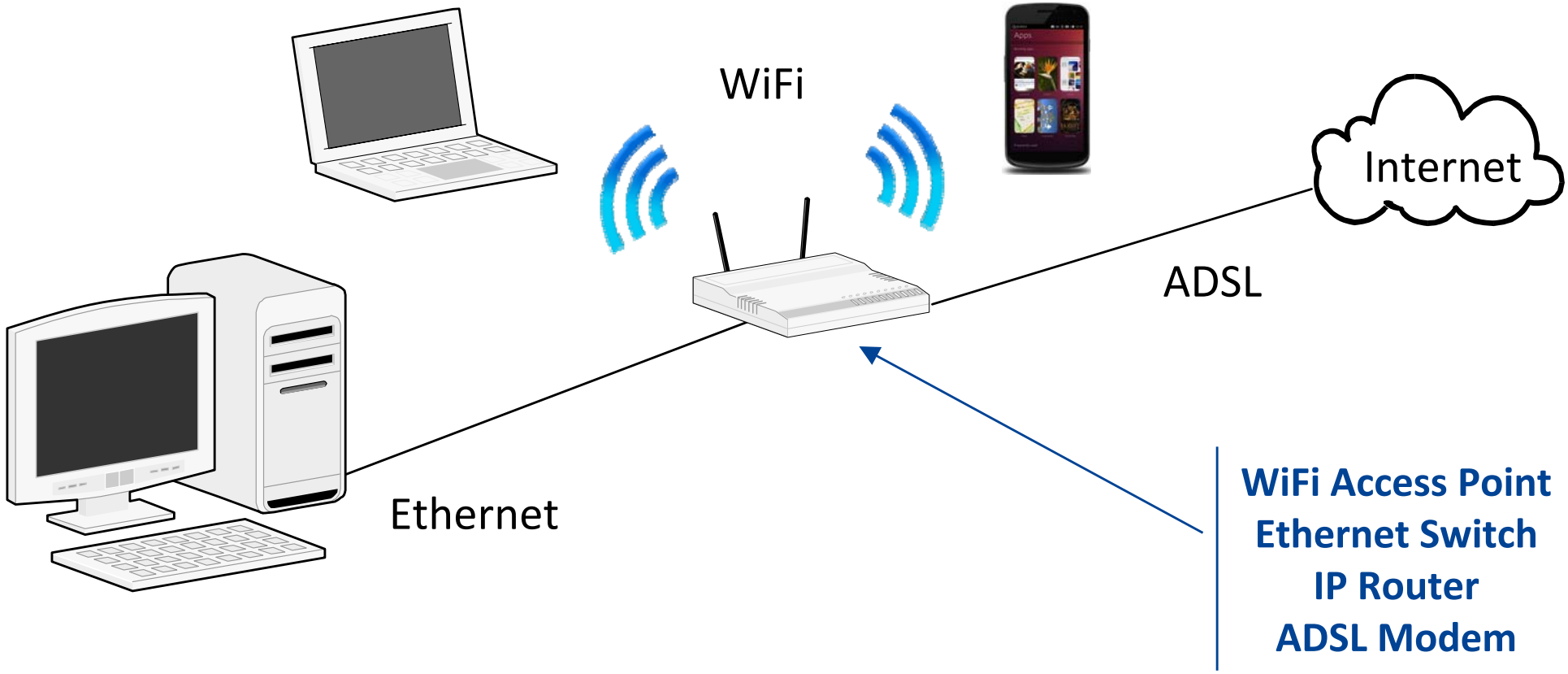
c:\>
```

Access Networking

Connecting the User to the World

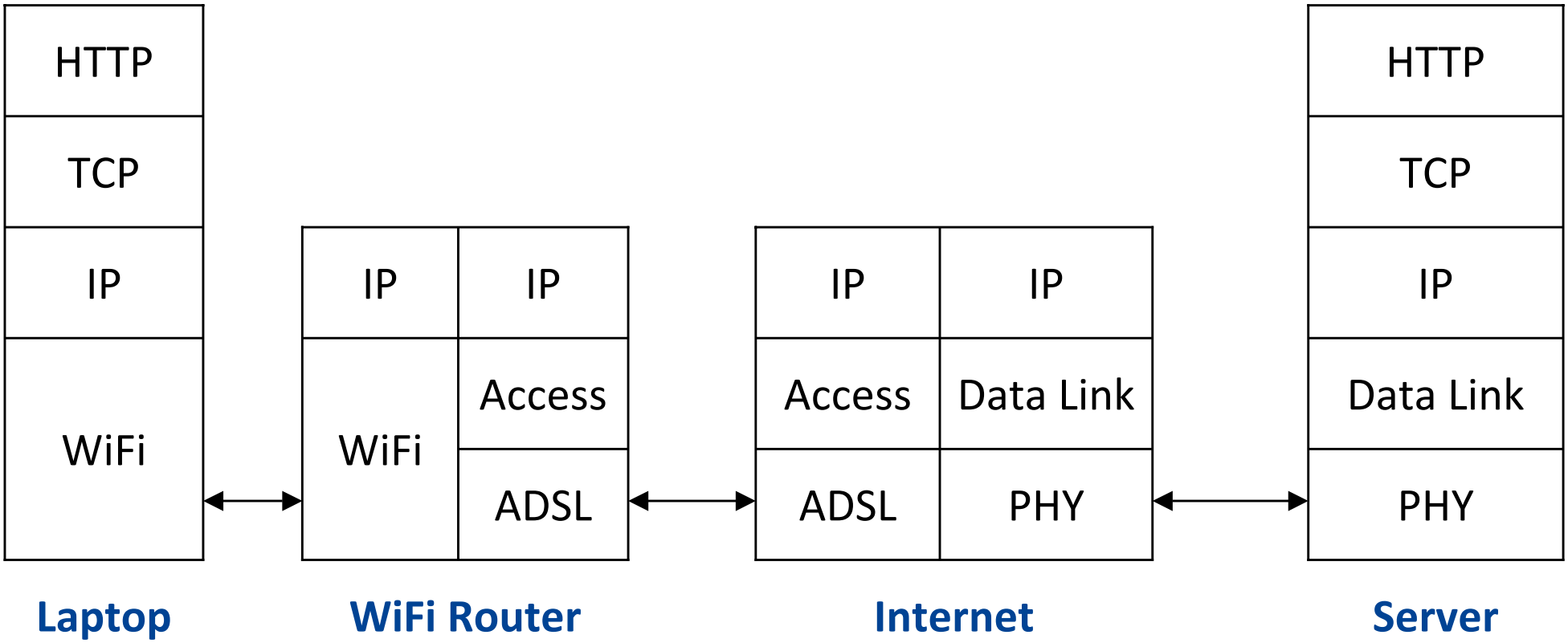
Small Office / Home Office (SOHO)

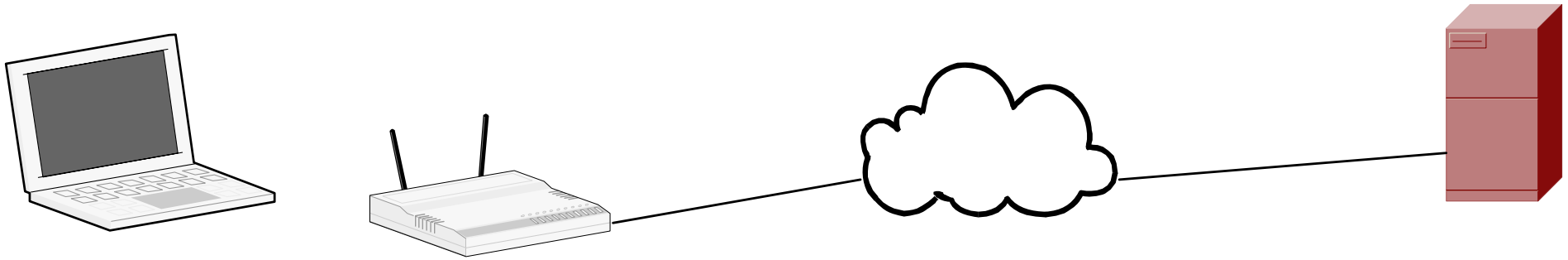
LAN (Local Area Network) to WAN (Wide Area Network)



Ethernet	802.3	Cable-based **LAN** protocol defined at **DATA LINK** and **PHY** layers
WiFi	802.11	Wireless **LAN** protocol defined at **DATA LINK** and **PHY** layers
ADSL	G.992.5	Cable-based transmission protocol defined at **PHY** layer

Laptop Browser to Web Server — Simplified View

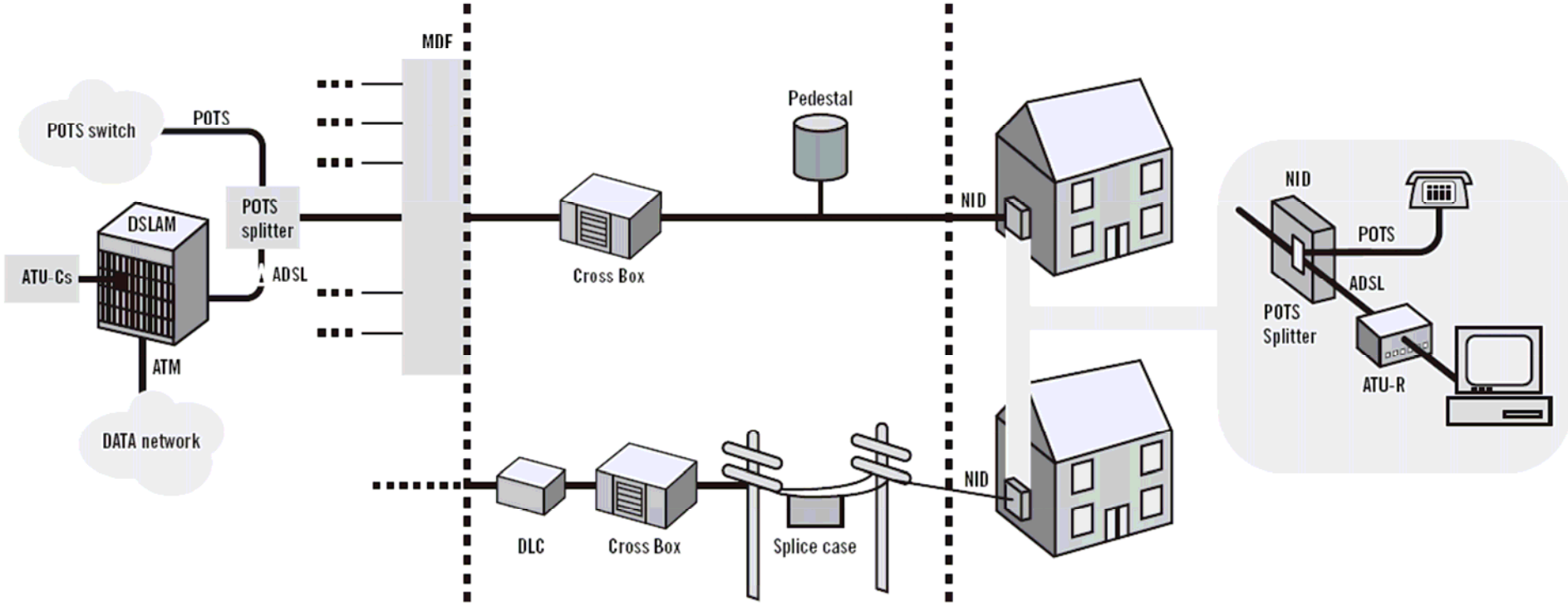




ADSL — Asymmetric Digital Subscriber Line

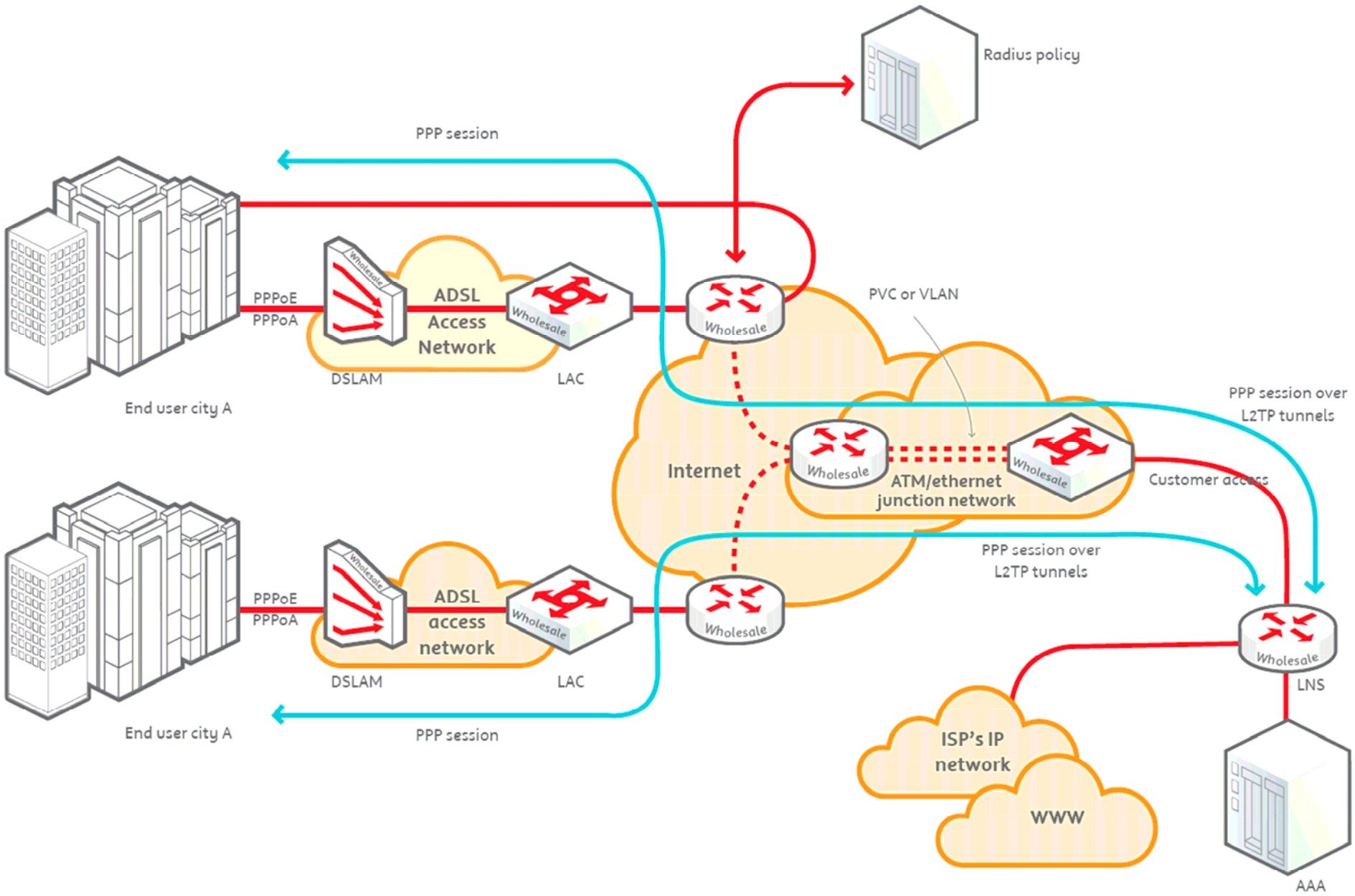
High speed transmission on standard voice line

- POTS — plain old telephone service
- 24 Mbps downstream
- 3.3 Mbps upstream



Ref: JDSU, **ADSL Technology**, JDS Uniphase Corporation, 2005

ADSL Access Network



Ref: Vodaphone, **Wholesale Layer2 DSL (W-DSL-L2I)**, VTCW011 - I 03/13

Typical Bezeq ATU-R

בזק. הכי טוב בבית

מדריך התקנה
מודם נתב אלחוטי
D-Link N 2760U



802.11
WiFi

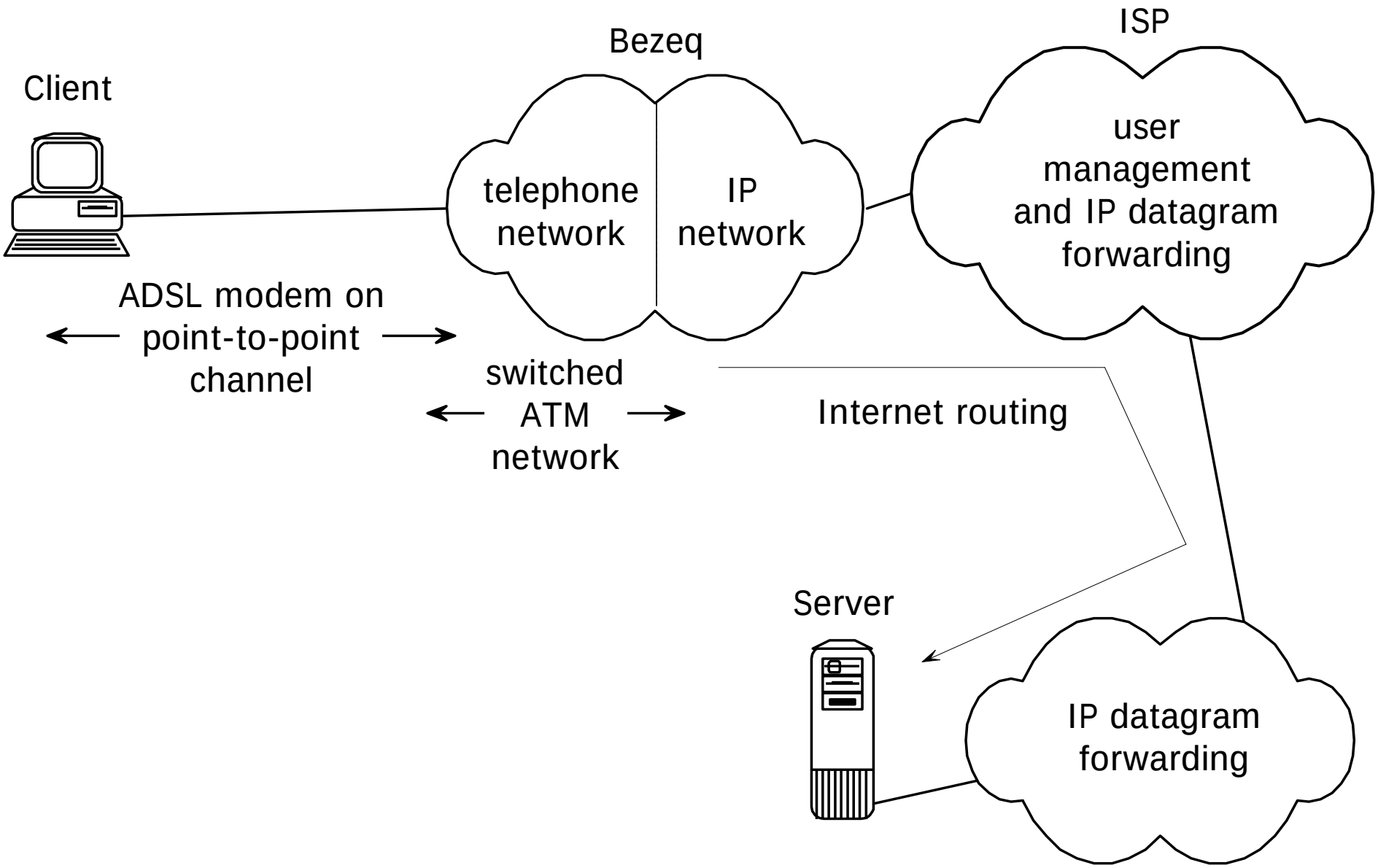
802.3
Ethernet

IP
Routing

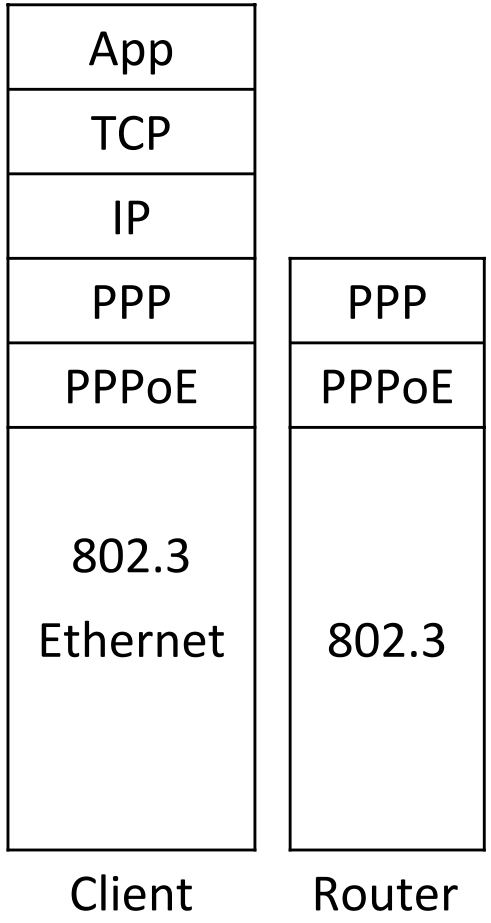
ADSL
33 Mbps



Bezeq Fast Internet Access



Fast Internet Protocols — Envisioned Campus Case



PPP

Point to Point Protocol

Logon + connection management

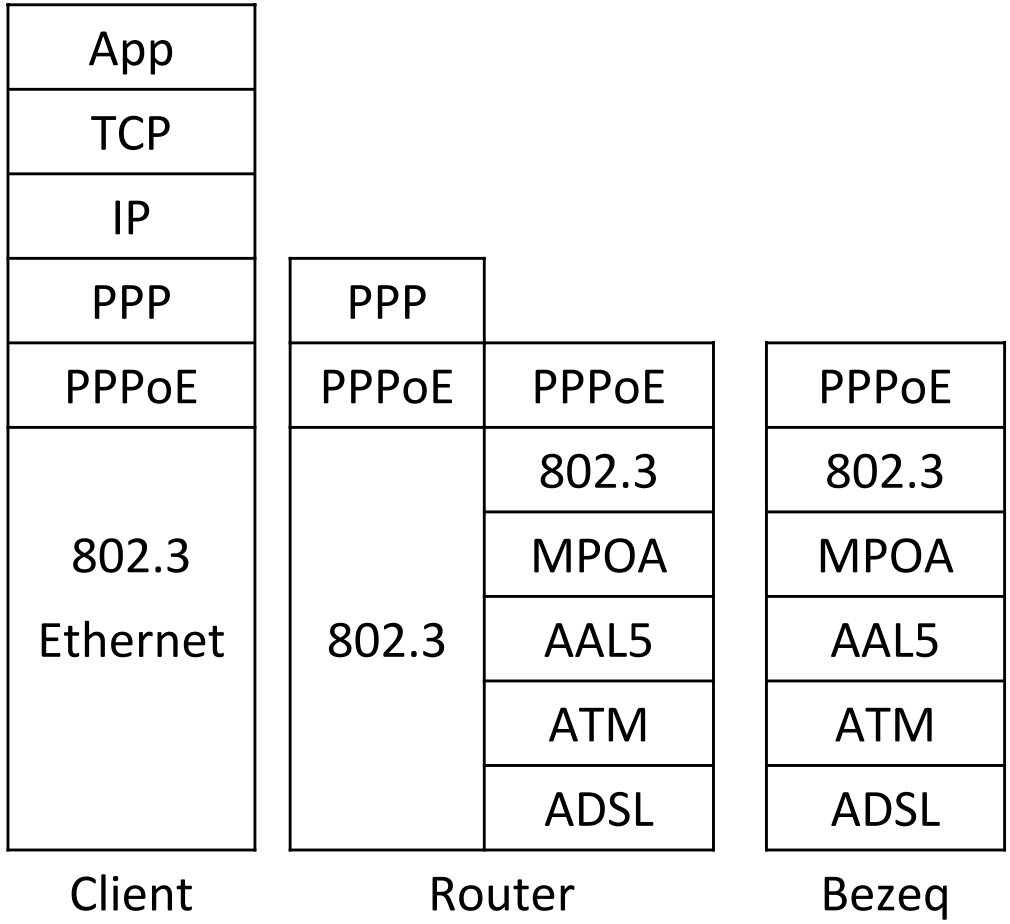
PPPoE

PPP over Ethernet

Virtual point-to-point connection over shared LAN

Client opens private session with ISP

Fast Internet Protocols — Envisioned Campus Case



ATM

Asynchronous Transfer Mode

Data Link protocol for broadband telephone services

Permits real time QoS

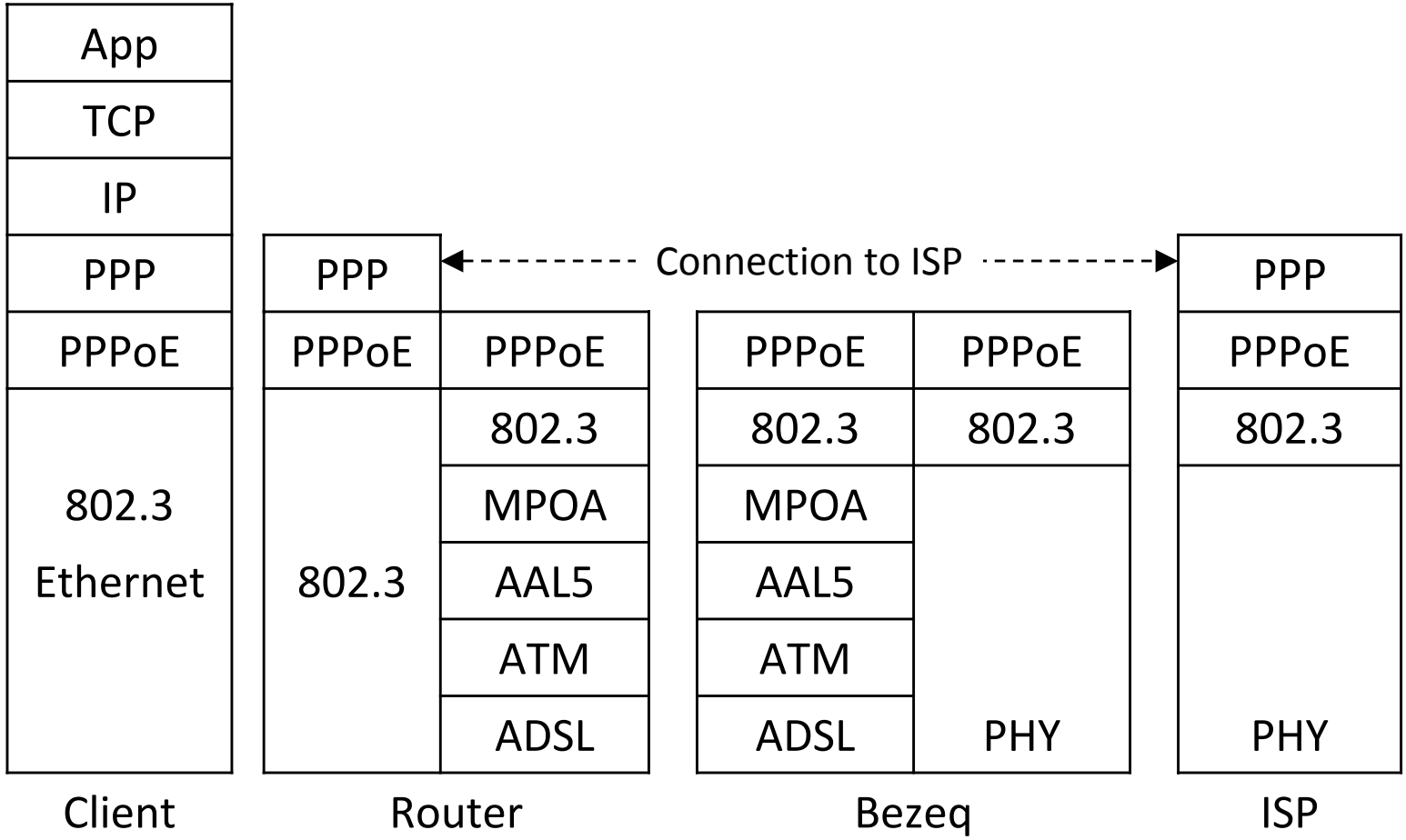
MPOA + AAL5

Adaptation protocols for ATM

ADSL

Physical bit transmission

Fast Internet Protocols — Envisioned Campus Case



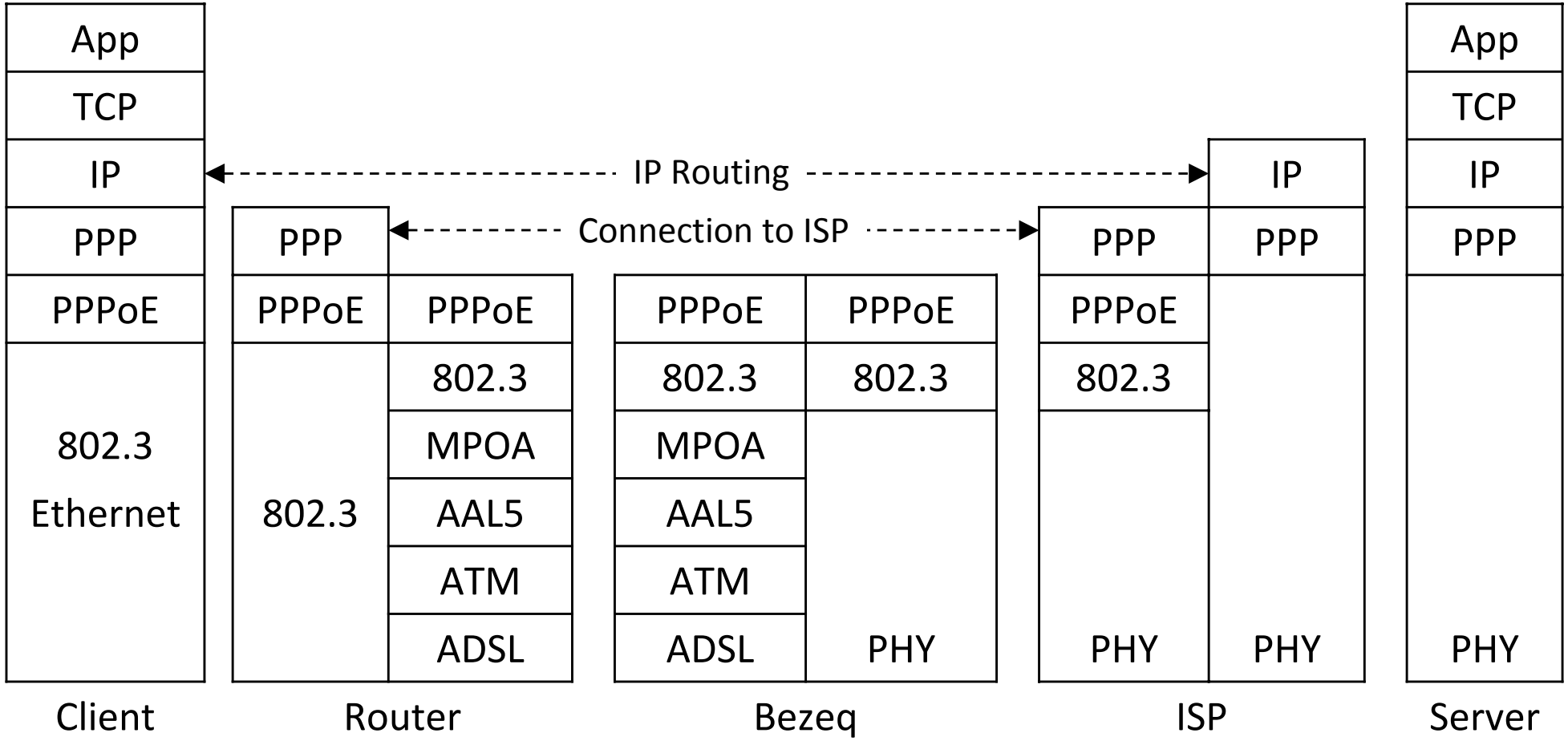
Connection to ISP

Client runs Network Control Protocol (NCP) over PPP

CHAP (challenge handshake authentication protocol) — User Name + Password

ISP authorizes user and engages IP forwarding

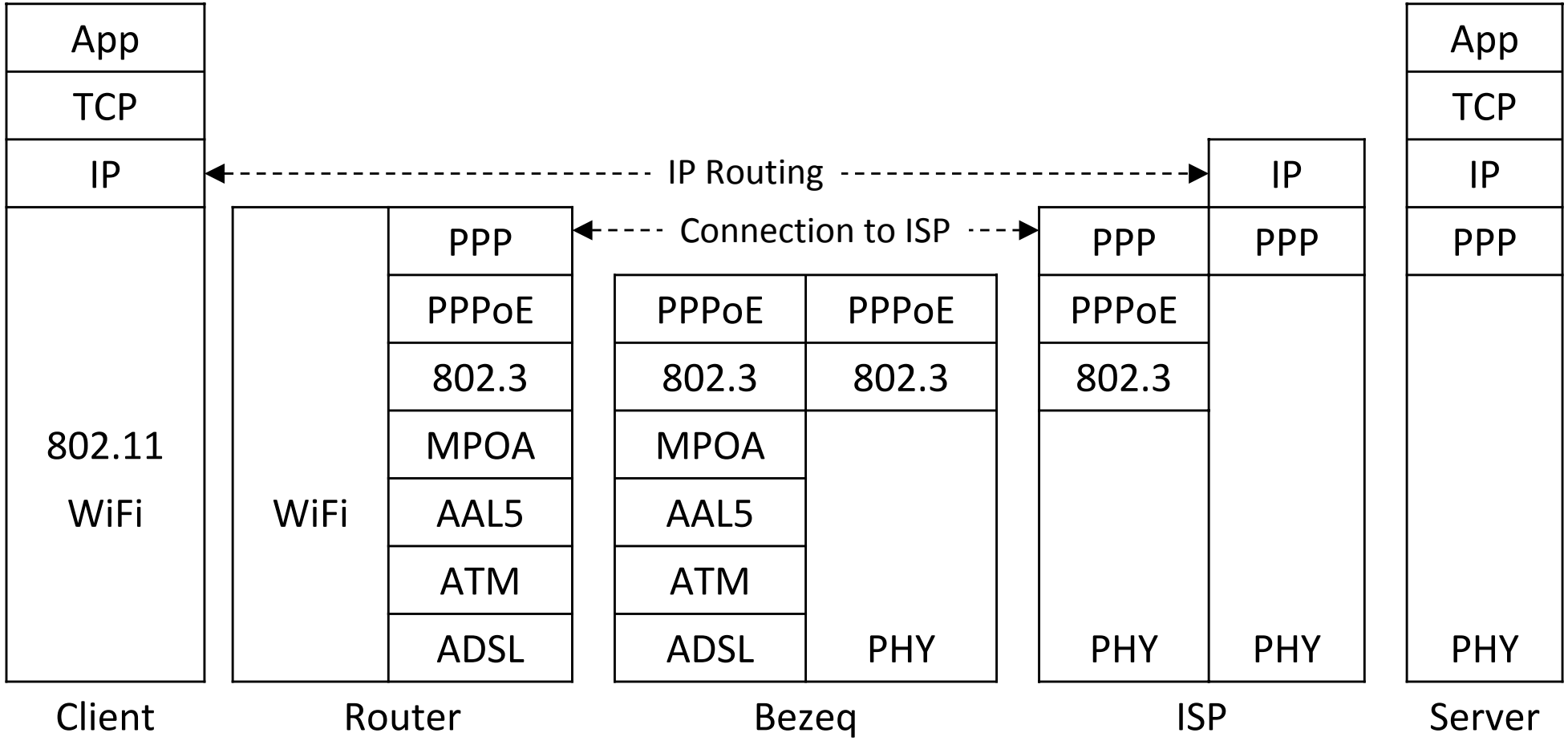
Fast Internet Protocols — Envisioned Campus Case



IP forwarding

ISP forwards IP datagrams to server via Internet backbone

Fast Internet Protocols — Typical SOHO Case



Router/modem initiates connection to ISP

Runs NCP over PPP over PPPoE over Ethernet

Router provides always-on Internet access over WiFi + Ethernet

Public Switched Telephone Network (PSTN)

Local loop (last mile) wired service

Analog voice + data

Voice → 64 kbps (DS0)

PDH / SDH digital hierarchy

DS0 streams combined to hierarchy of data rates

1.544 Mbps (T1) to 40 Gbps (STM-256)

ESS7

Hierarchical tree of central office switches for DS0 streams

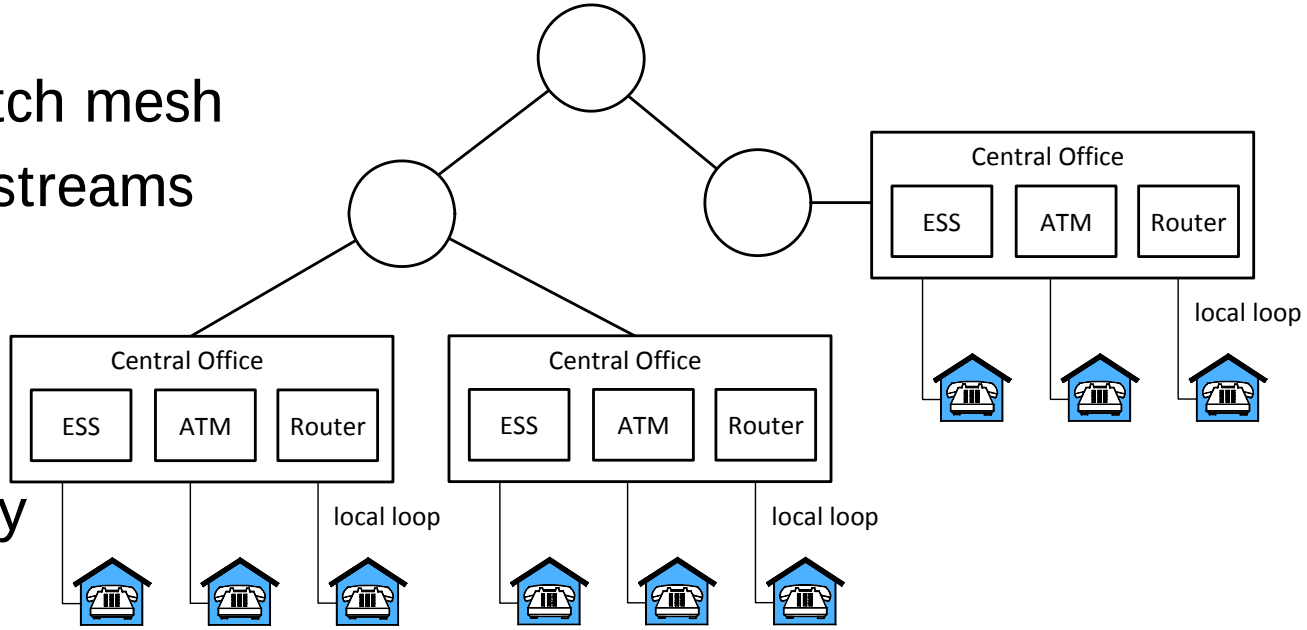
ATM

General packet switch mesh

Switches 2.5 Gbps streams (STM-16)

Telco

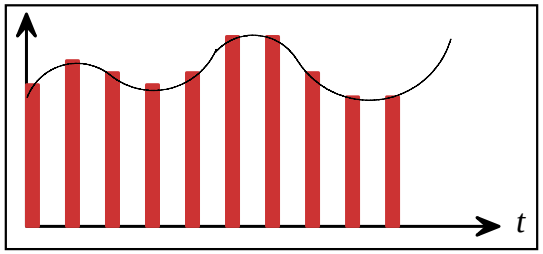
Telephone Company

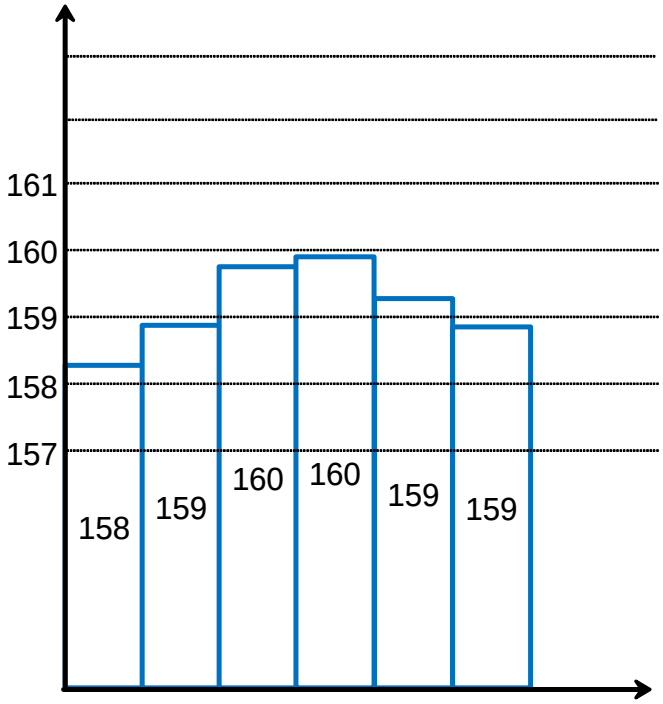


Digital Voice on Telco Telephone

Sample analog voice signal every 0.125 ms

0.125 ms per voice sample ⇒ 8000 voice samples / second





Round-off sample to 8-bit data

Data ∈ {0, 1, 2, … , 255}

Sample = {158.276, 158.879, 159.724, 159.821, 159.312, 158.791}

Data = {158, 159, 160, 160, 159, 159}

DS-0 stream

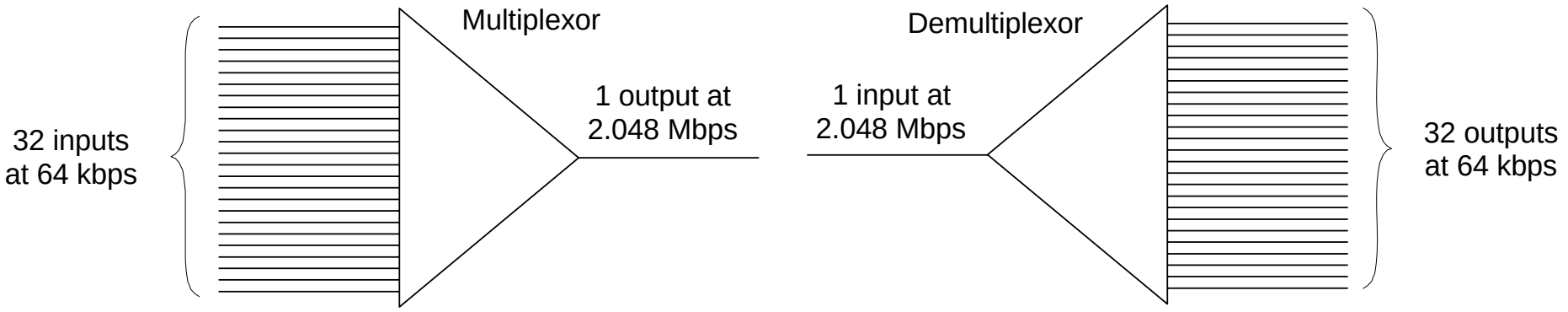
(8000 samples / second) × (8 bits / sample) = 64 kbps

64 kbps digitized voice (no compression)

Data Concentration

Multiplexing

Combine multiple nodes onto one link

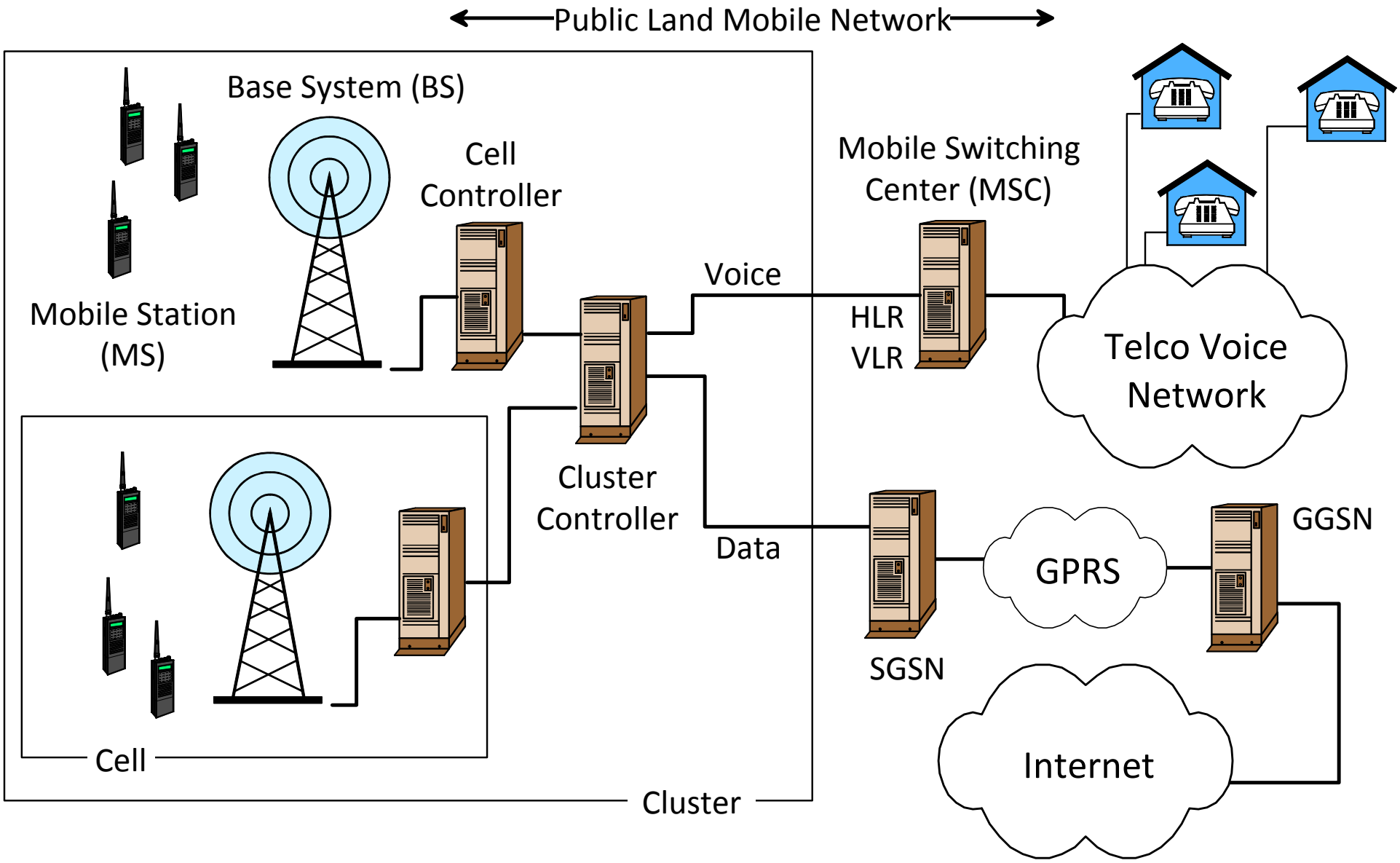


PDH		
E1	32 x DS0	2.048 Mbps
E2	4 x E1	8.448 Mbps
E3	4 x E2	34.368 Mbps
E4	4 x E3	139.264 Mbps

SDH		
STM-0	21 x E1	51.84 Mbps
STM-1	E4	155.52 Mbps
STM-4	4 x E4	622.08 Mbps
STM-16	16 x E4	2.48832 Gbps
STM-64	64 x E4	9.95328 Gbps
STM-256	256 x E4	39.81312 Gbps

Cellular Network in 2G

Wireless to base station — uses Telco network for voice service

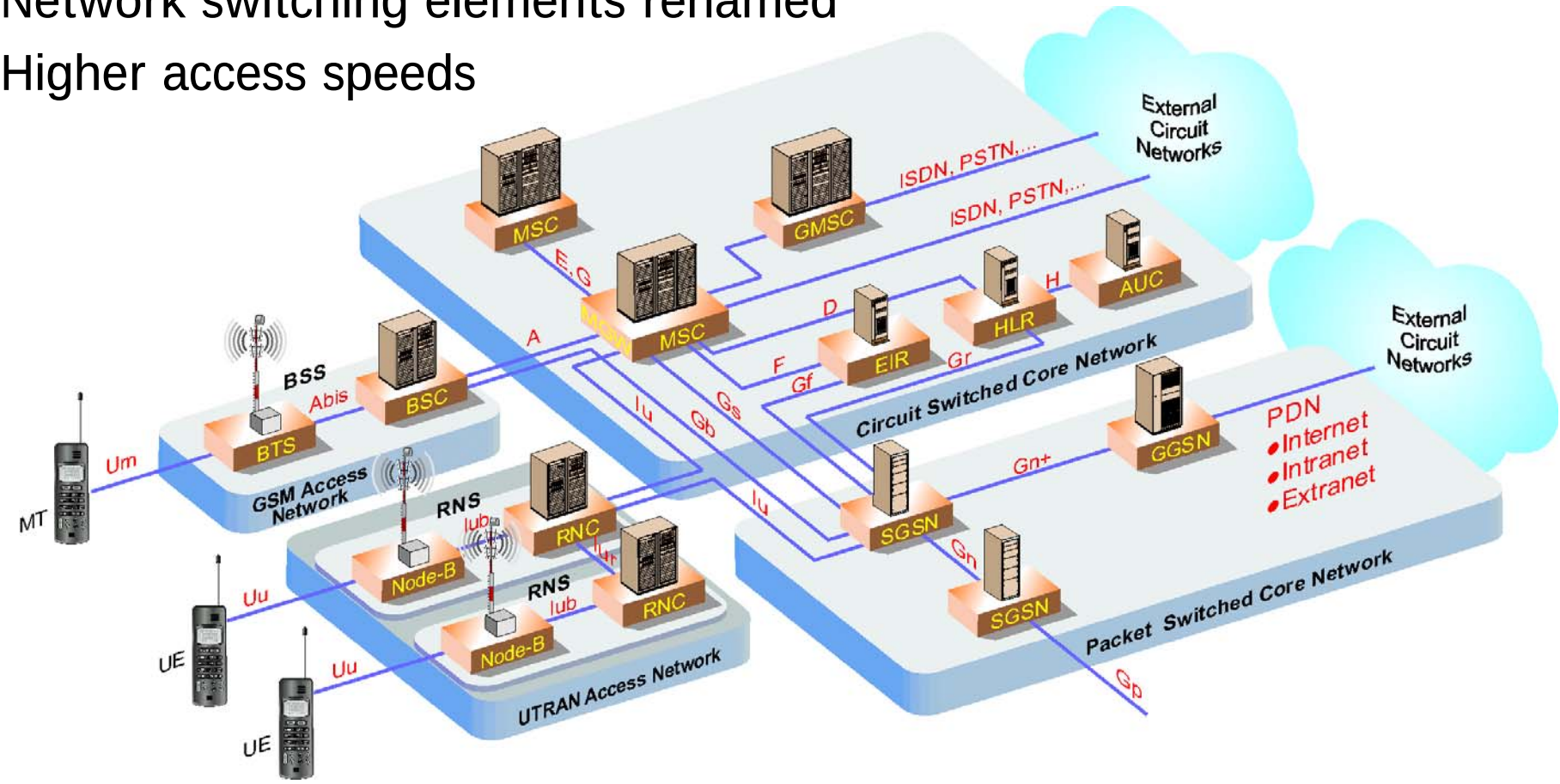


3G / 4G / 5G Cellular Networks

3G — evolution / improvement of 2G

Network switching elements renamed

Higher access speeds



4G / 5G — packet switched core network only

Eliminate circuit switched core network for voice

Voice calls carried as Voice over IP (VoIP)

Network Infrastructure

Economic perspective

Enormous investment in existing equipment

- Global network of hardware nodes + transmission lines
  - Developed to provide many services
  - Internet (IP-based unreliable connectionless) just one service
- Most developed before Internet
  - Telegraph — 1794
  - Telephone — 1876
  - Teletype modem — 1943
  - Digital telephone — 1962
  - Internet opened to public — 1992

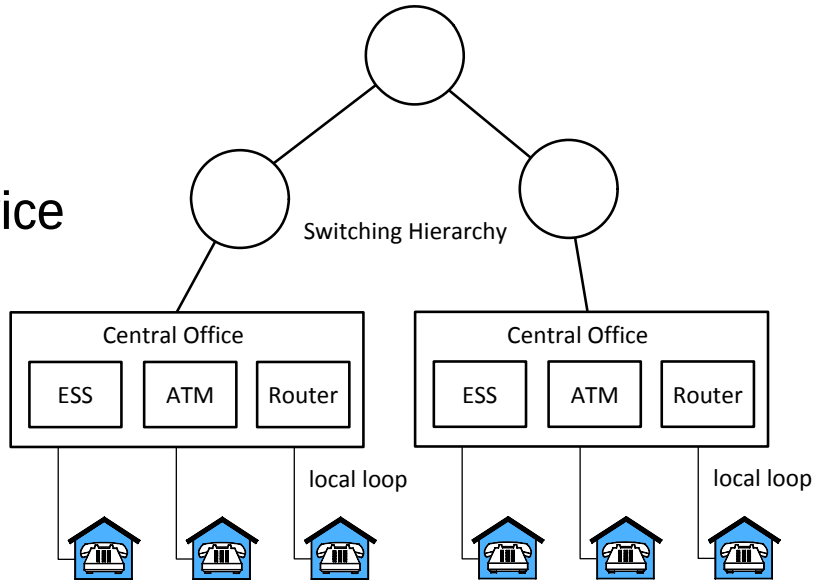
Hardware updates

- Replacement of manufactured hardware
- Slower than software updates
- More expensive than software updates

Changes in Telephone Business: 2000 — 2008

Enterprise networking needs

- Business revolves around data center
  - Access + storage + processing + service
- But employees still talk on phone
  - Mobility = standard
  - Voice calls cost money
  - Video calls growing



Network infrastructure providers (Telcos)

- Existing infrastructure designed for voice calls — not profitable
- Profit in leasing lines for data networks
  - Internet + private WAN + mobile backhaul (connections between cells)
- Strategy
  - Eliminate PSTN
  - Sell central office real estate to raise cash
  - Build data-oriented mesh network
  - Support voice as media streaming

Next Generation Networks (NGN)

ITU model for network planning in 21st century

ITU-T Recommendation Y.2001 (12/2004)

All-IP network

Every service operates over IP

IP operates over every infrastructure

Universal gateway point

Eliminate traditional service boundaries

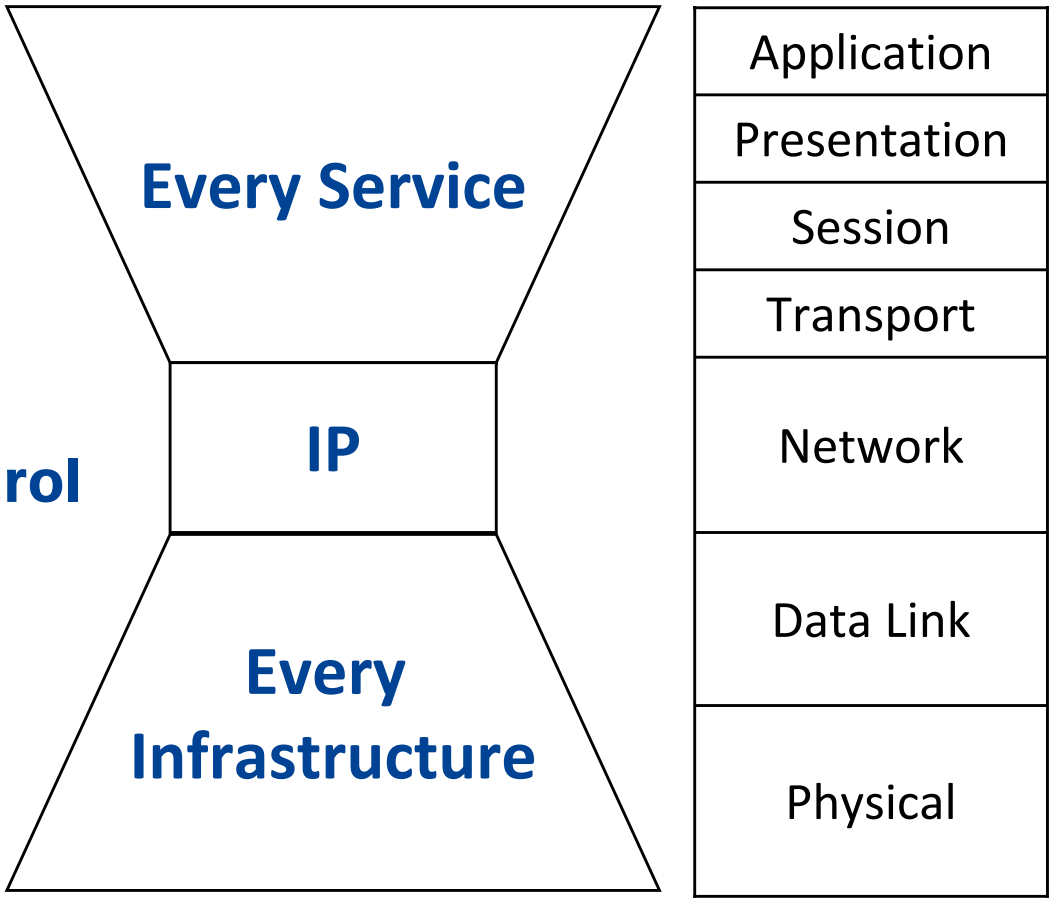
Universal mobility

Every service to every user

Quality of Service (QoS) control

Protocols with QoS fields

- IPv6
- MPLS
- SIP



Virtualization in Networking

Dedicated system

- Hardware constructed to perform one specific task
- Examples
  - WiFi access point
  - Internet router

Virtual system

- Implemented in software on programmable generic hardware
- Examples
  - Router implemented software
  - Java code running on Java VM over Linux

Transparency

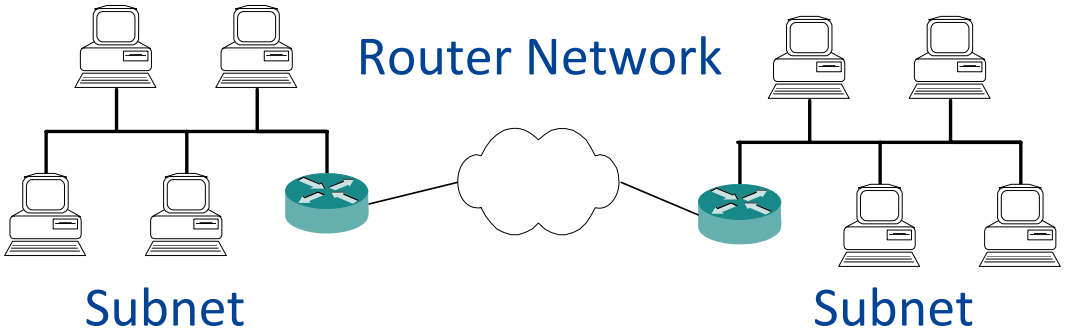
- Virtual system must look like dedicated system
  - To service requesters
  - To service providers

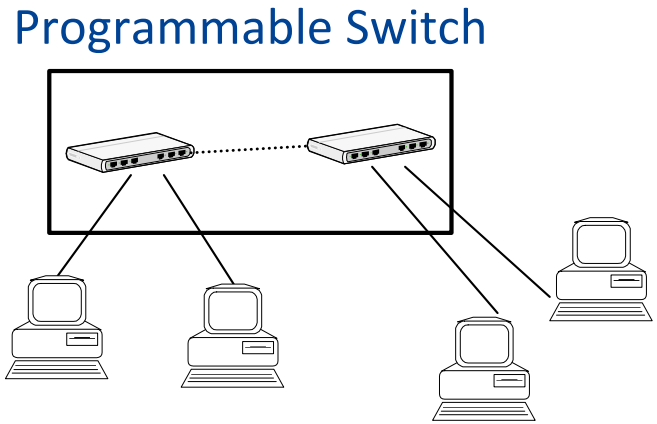
Standard Virtual Networks

Virtual LAN (vLAN)

LAN switch configured to partition nodes into subnets

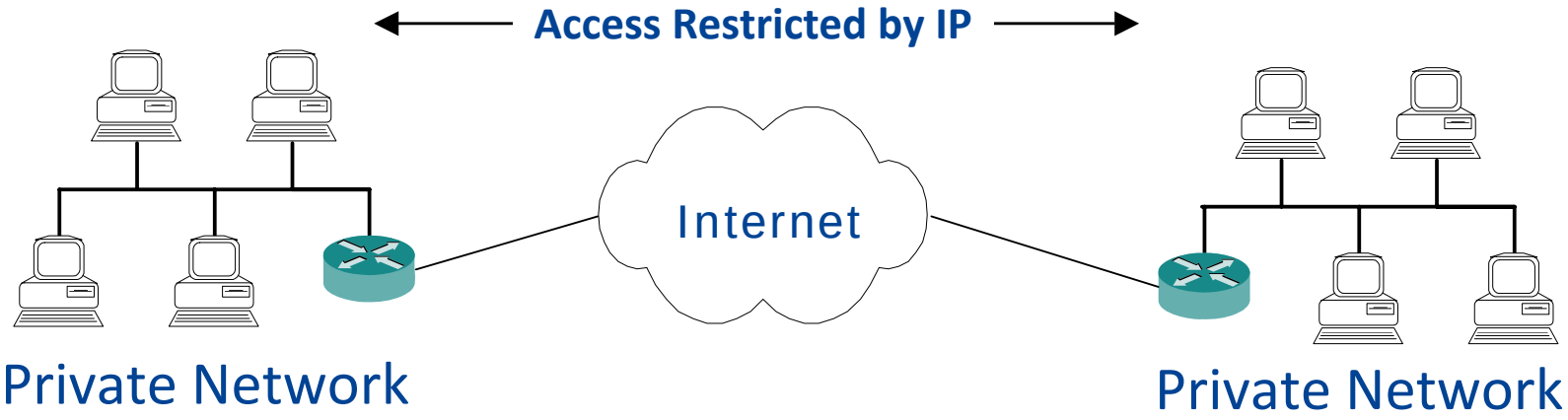
No router needed for subnet partitions





Virtual Private Network (VPN)

Private network implemented on public infrastructure



Traditional Networking Functions

Forwarding function

- Data transfer
- Implement network protocols
- High performance in specially designed dedicated hardware
  - Switches
  - Cable connections
  - Programming at console interface

Control function

- Configuration of network topology and policies
- Management of forwarding function
  - Supervision
  - Measurement
  - Maintenance

Software Defined Networking (SDN)

Virtual network

- Generic programmable hardware
  - Fast implementation of networking protocols
  - High capacity I/O
  - Standardized control interface for configuration
- Configured in software to implement networking functions
  - Modes, connections, protocols, policies

Data plane (user plane)

- Forwarding of user data

Control plane

- Network configuration data
- Security, maintenance, management

Fabric

- Hardware configured to implement SDN

Logical device

- Networking function implemented in SDN

Software Defined Networking (SDN)

SDN Application

Communicates network requirements to SDN Controller

Receive abstracted view of network

SDN Controller

Translates requirements from SDN application into SDN Datapaths

Provides SDN applications with view of network

SDN Datapath

Logical network device controls data forwarding hardware

Provides configured data forwarding capabilities

Programmable network

Regard network definition as problem in software engineering

Cloud Computing Business Model

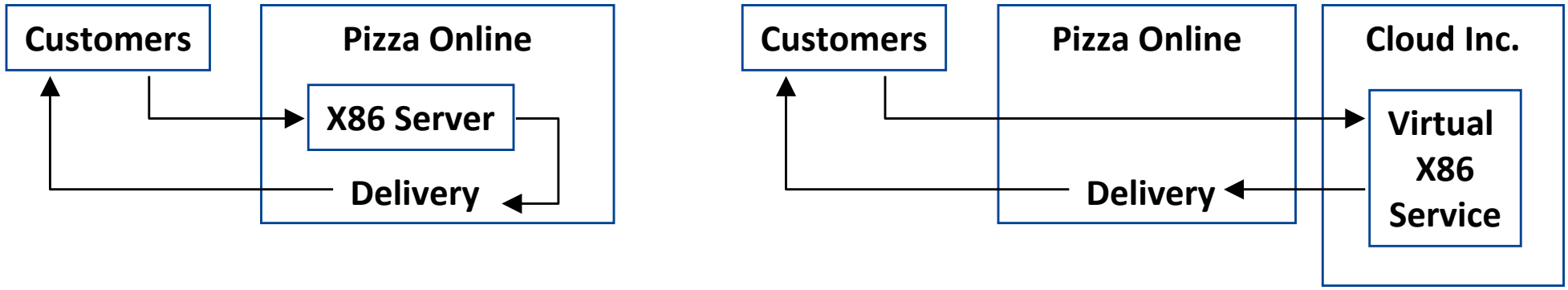
Outsourcing service model

Replace user hardware/software with "computing as service“

Implement virtual networks

Service Level Agreement (SLA)

- Defines service provided to user
- Guarantees performance and quality of service
- Provider handles operations+administration+maintenance (OAM)



Business advantages

Economies of scale to large provider → lower cost to user

User cuts labor/capital costs from balance sheet → happy investors

Cloud Computing Architecture

Cloud service organized from conventional resources

- Provider installs computing / networking hardware
  - Usually large servers or mainframes
- Provider offers menu
  - Mix of hardware + software + network types
- User chooses Service Level Agreement (SLA) from menu
- SLA implemented with dedicated or virtual system

Unique technological issues

- Service reliability
  - Provider financially committed to SLA
- Provider-side
  - Seeks minimum configuration cost for SLA
- User-side
  - Seeks minimum contract cost for requirements

Service Hierarchy in Cloud Computing

Infrastructure as a service (IaaS)

User sees virtual hardware environment

Real hardware or hypervisor / system virtual machine

User installs OS → installs software → runs jobs

Platform as a service (PaaS)

User sees virtual OS environment

OS on single hardware platform or virtual OS

User installs software → runs jobs

Software as a service (SaaS)

User sees virtual application software environment

Applications running on private OS or "sandboxed" on shared OS

Sandbox — private execution environment per application instance

User runs jobs

Storage as a service (STaaS)

User sees virtual mounted storage device

Considerations in Cloud Computing

Cost

- Provider issues
  - Economies of scale $\Rightarrow$ lower cost per compute job
- User issues
  - Capital + OAM costs $\rightarrow$ operating costs
  - Lower start-up costs $\Rightarrow$ operating debt

Reliability

- Provider issues
  - Redundant infrastructure $\rightarrow$ continuity + disaster recovery
  - Centralized management of OAM, security, performance
  - Virtualization $\rightarrow$ serve multiple users on physical server
  - Multitenancy $\rightarrow$ provide multiple sandboxed application instances on OS
- User sees guaranteed service

Agility

- User / provider reconfigure service / infrastructure as needed
- Growth, load balancing, time-zone serving

Cloud Ownership

Public cloud

- Service provider as public utility — sells / rents computing service
  - Initial providers leverage large existing infrastructure
  - Amazon, Microsoft, Google, IBM
- Menu of services at fixed prices

Private cloud

- Cloud infrastructure for private organization
- Managed internally or outsourced
- Isolates service developers from implementation issues
  - Standard development platform
- Requirements for economic justification
  - Large organization
  - Technology-based services
  - Frequent new service
  - Example — internet content provider

What's Different in Cloud Networking

End user

Not much

Accesses service "somewhere" on network

Business service provider

Define business service

Outsource implementation

OAM limited to SLA-level virtual environment



Cloud service provider

Manage vast real environment mapped to virtual environments

OAM requires effective picture of real system from SLA POV

Centralize → Decentralize → Centralize → ?

1950s — 60s

Centralized mainframe computer + multiple OS instances over hypervisor

Timesharing OS serves multiple users

User sees OS environment via dumb terminal (thin client)

1970s

User applications offloaded to minicomputers + timesharing services

User sees timeshared OS environment via dumb terminal

1980s

User applications offloaded to personal workstations (PC)

User sees single-user OS environment running locally

1990s

Network single user workstations

User sees single-user OS environment running locally

2000s

Centralized control of local OS environment by IT departments

2010s

Cloud + netbook / tablet / smart phone — dumb terminal with high-res GUI

Enterprise Networks

Enterprise

The word (ref: http://www.etymonline.com)

- Past participle of entreprendre — "undertake, take in hand"
- From Old French — entre "between" + prendre "to take"

The pose



Enterprise network

Larger + more complex + more expensive + more awesome than SOHO

Integrated LAN + WAN technologies

- Traditionally implemented in specifically-designed hardware systems
- Increasingly implemented as Software Defined Networks (SDN)

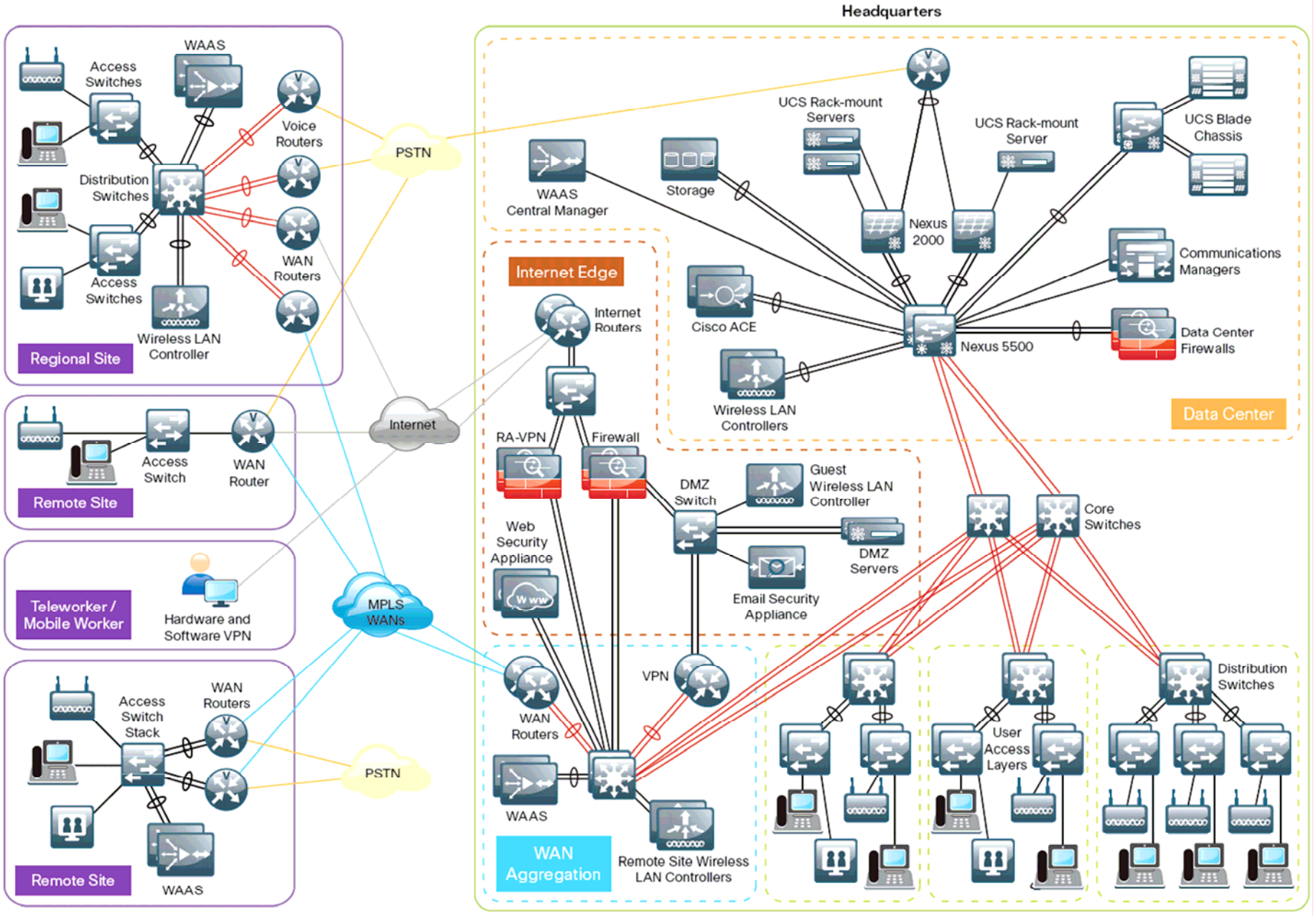
Security

High overall traffic volume

Traditional Network Infrastructure

Cisco Smart Business Architecture (SBA)

Ref: WAN Design Overview, Cisco 2013



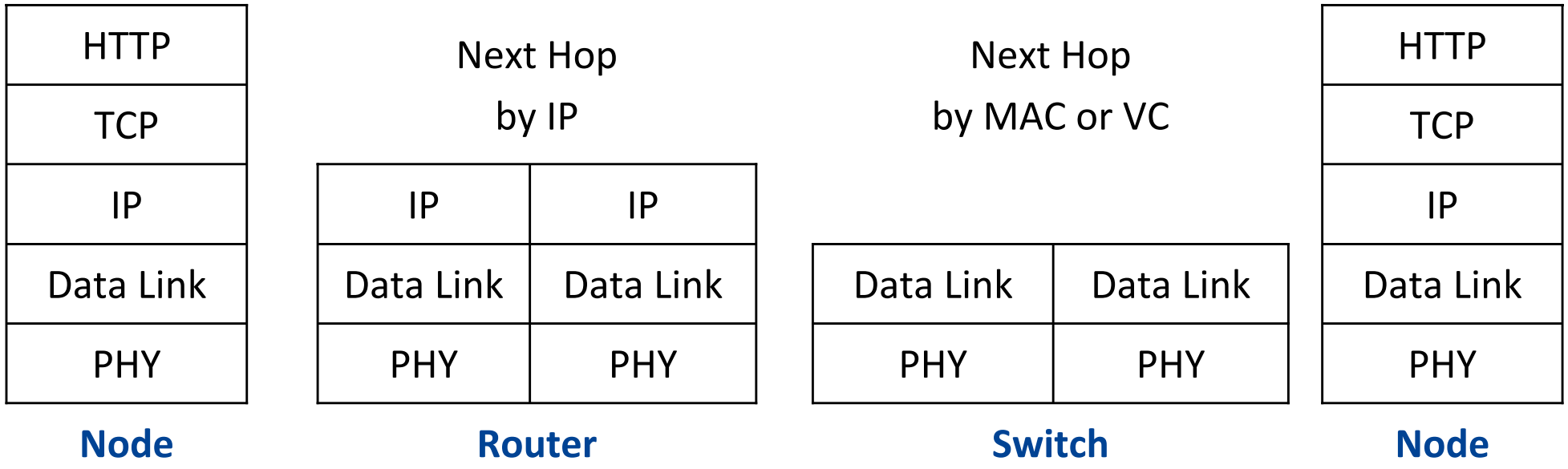
Technologies in the Cisco SBA

Conventional Layer 3 routing

- Permits universal access to any host / node
- Next-hop IP datagram forwarding

Conventional Layer 2 switching — N x N non-blocking switching

- Permits access to nodes on same physical network
- Ethernet MAC physical port-to-port switching
- ESS7 64 kbps voice telephone switch
- ATM cell switching for switched and permanent virtual circuits
- Frame Relay switching for permanent virtual circuits



Technologies in the Cisco SBA

Multiprotocol Label Switching (MPLS)

Label header added to IP datagram

Label identifies end-to-end route

Implements end-to-end virtual circuit

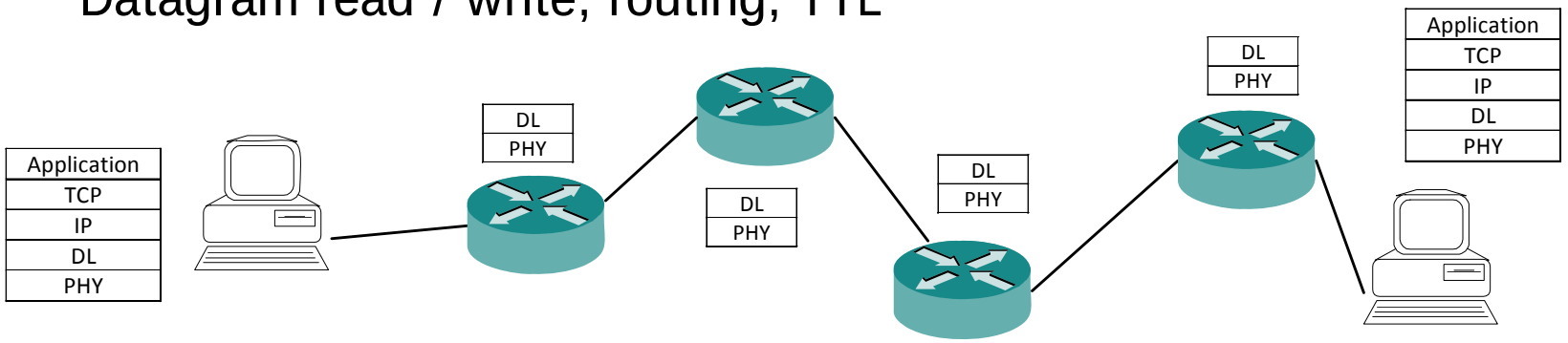
MPLS enabled router

Next hop by MPLS label

Faster than next hop IP routing

Saves time of layer 3 processing

Datagram read / write, routing, TTL



Switch / Router Interaction

Permits vLAN and VPN definition

Optimizes multimedia streaming

Technologies in the Cisco SBA

Nexus 7000 Switch



- 100 Gbps Ethernet switch
- Copper or fiber access ports
- QoS control
  - Cut-through architecture (forward data without buffering)
  - Low latency + jitter
- Extensible through fabric extenders (FEX)
  - Scalable to 15.76 Tbps (15,760 Gbps)
- Supports virtual networking through MPLS

Nexus 2000

- Fabric extender (FEX)
- Add remote ports to Nexus 7000



Ref: http://www.cisco.com

Technologies in the Cisco SBA

Cisco XR 12000 Router

For large enterprises and service providers

1280 Gbps capacity

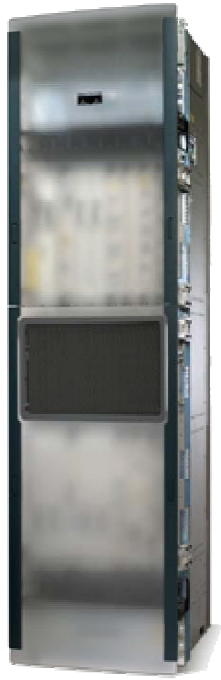
Internet protocols

IPv4/v6, MPLS

BGPv4/v6, IS-IS, OSPFv2.0, RIPv2, IGMP, DVMRP, PIM DX/SX

Infrastructure protocols

SONET/SDH, Ethernet, ATM, copper (DS-3/E3)



Cisco 7600 Router

WAN router

240 Gbps capacity

Infrastructure — Telco leased lines DS0 to OC-192



Cisco 3900 Router

For branch office

4 Gigabit Ethernet ports



Cisco Software-Defined Access Design

Underlay network

Physical dedicated hardware

Overlay network

Virtual network implemented in SDN

Management

Identity Services Engine (ISE)

- Account database

DNA Center

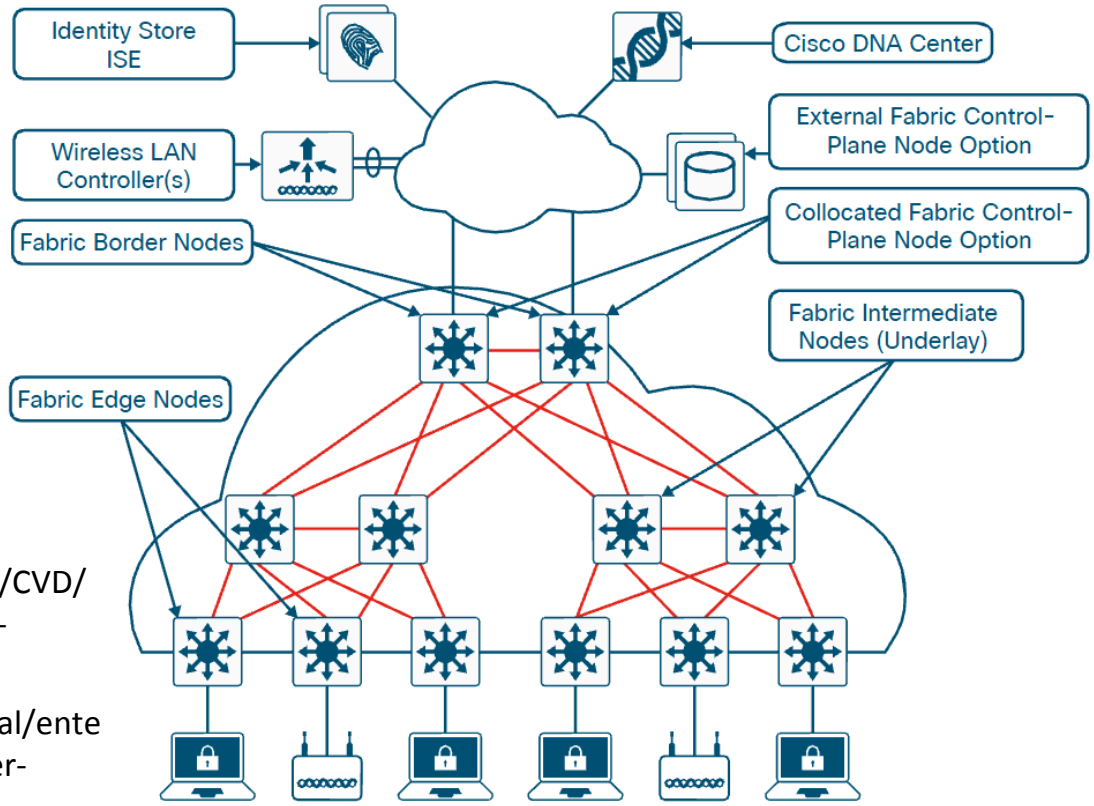
- Hardware database

Border node

- Connects fabric to WAN

Edge node

- Connects fabric to user



References:
https://www.cisco.com/c/dam/en/us/td/docs/solutions/CVD/Campus/CVD-Software-Defined-Access-Design-Sol1dot2-2018DEC.pdf

https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise-networks/enterprise-network-security/data-center-design-playbook.pdf

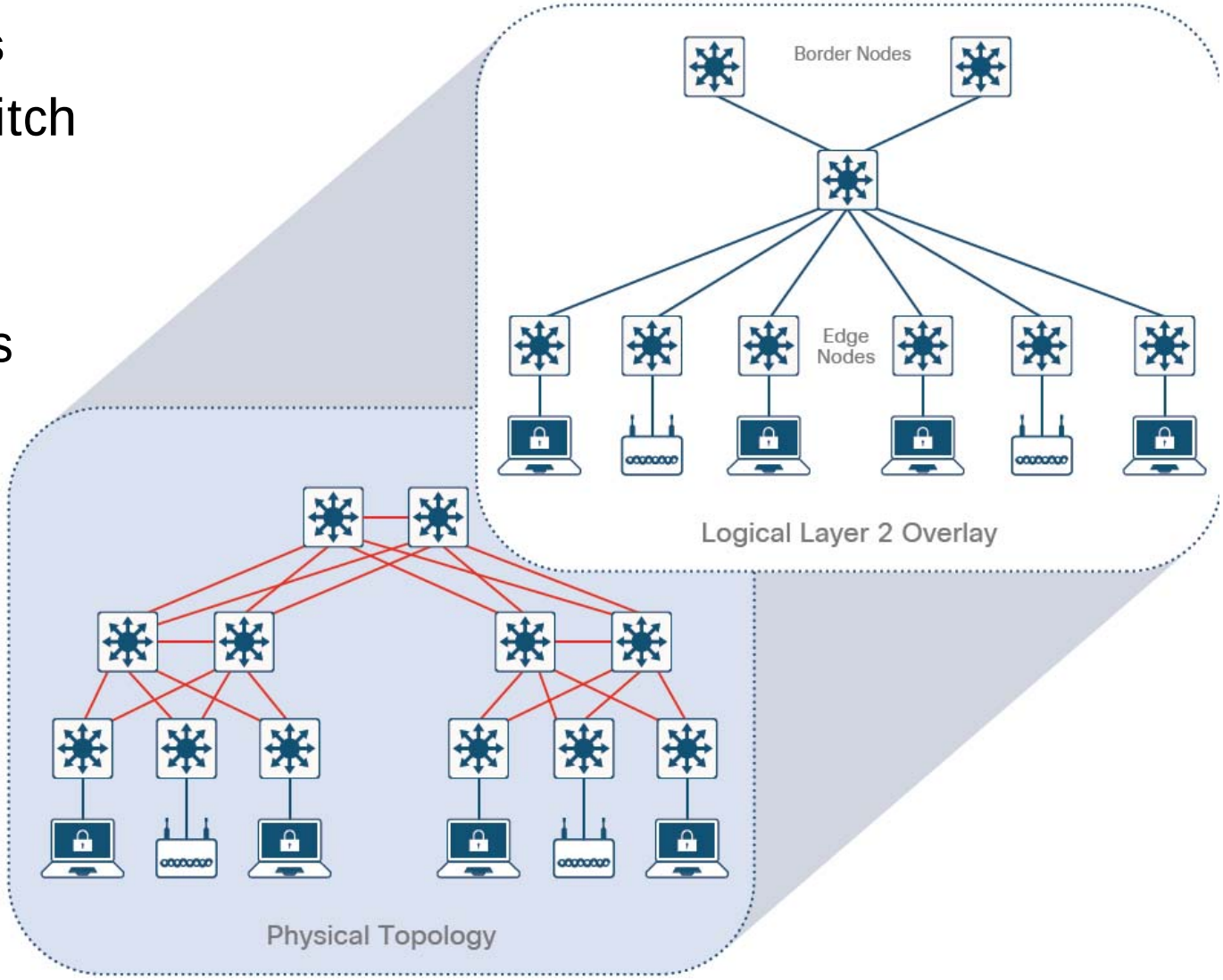
Layer 2 Overlay

Layer 2 (Data Link) Logical network

- Edge switches
  - Ethernet, etc
  - Connect users
- Intermediate switch
- Border switches

Physical network

- Physical switches
- Cables



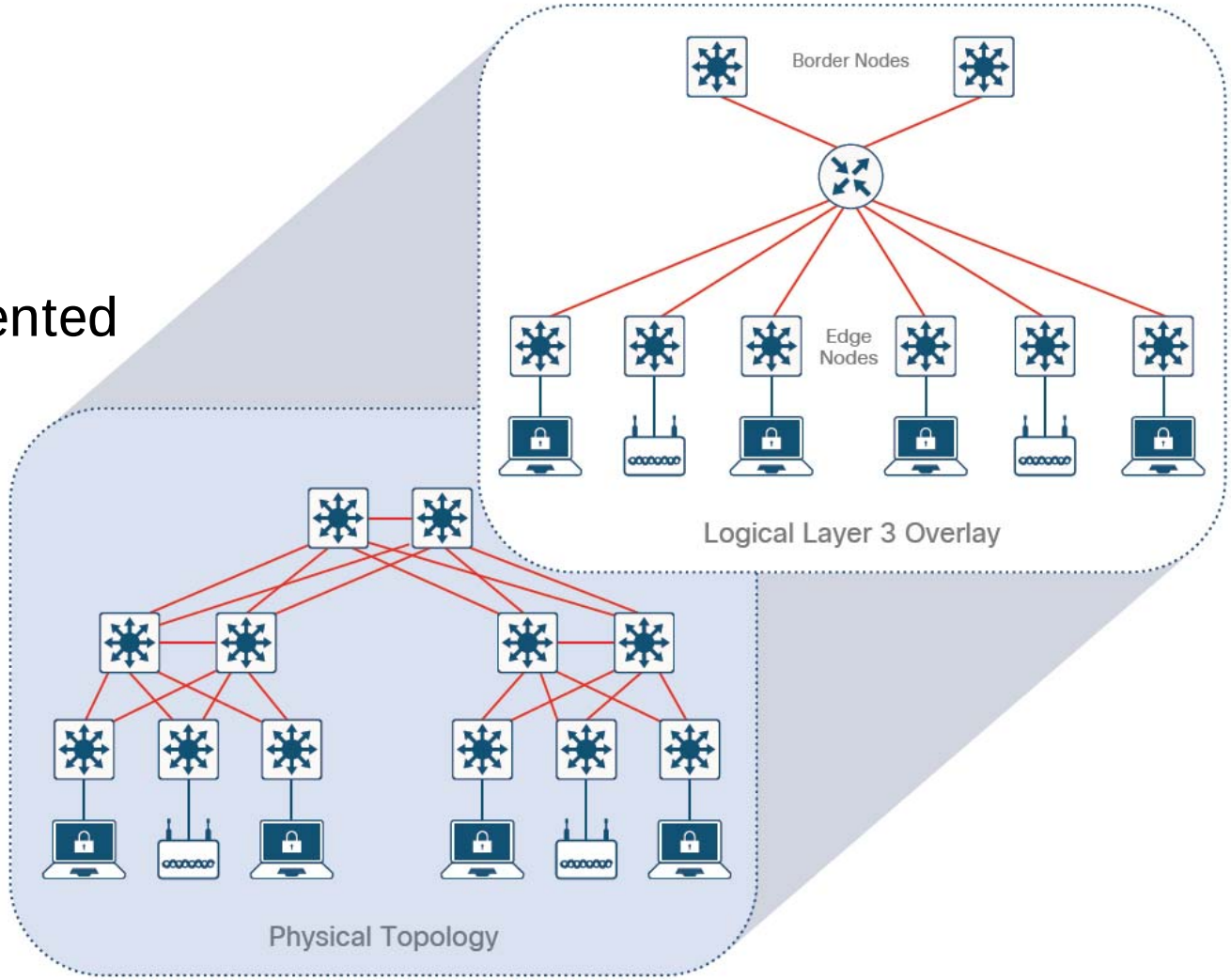
Layer 3 Overlay

Logical IP + switched network

- Edge and border switches
- Intermediate IP router

Physical network

- Same as Layer 2
- Physical switches
- Cables
- IP routing implemented in software



What Network Access Providers Do



RAD

Your Network's Edge®

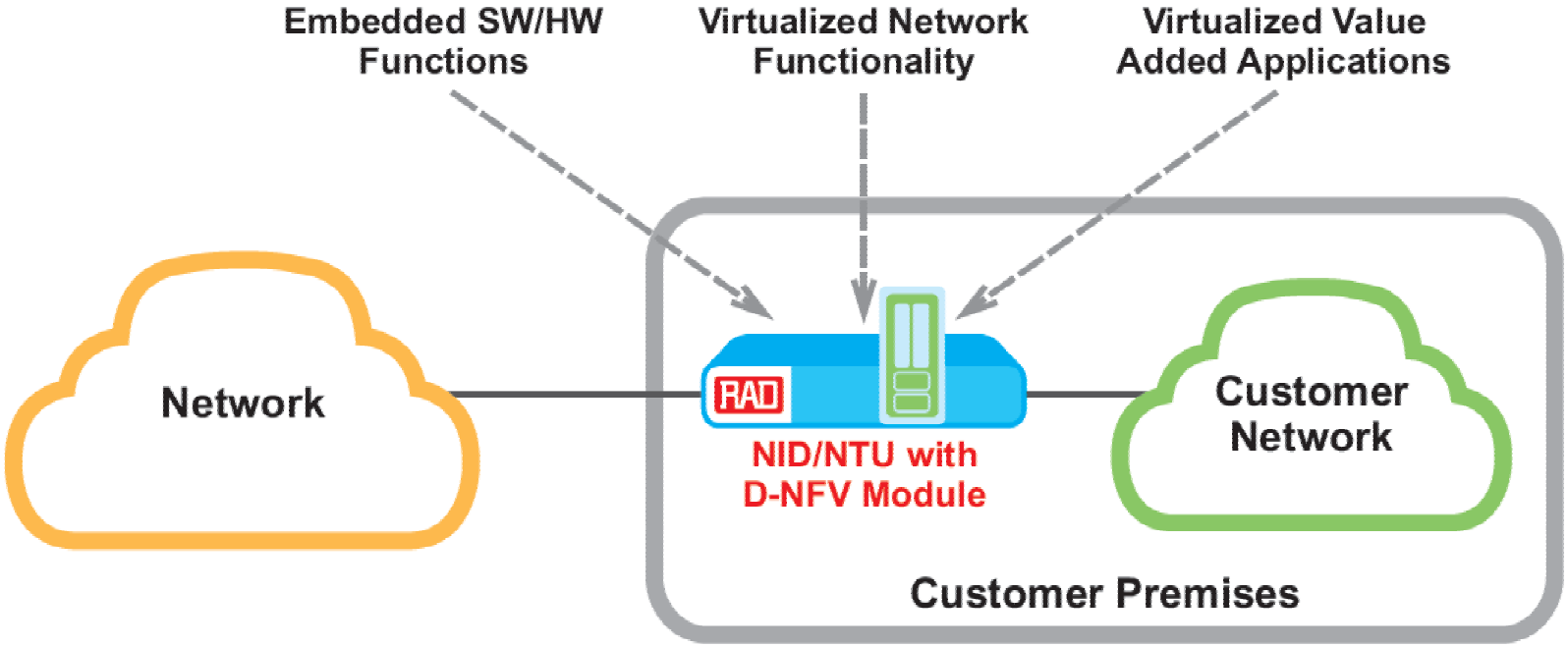
Service Assured Solutions

2019 Catalog

https://www.rad.com/system/files/Media/rad-catalog-2019.pdf
https://www.rad.com/system/files/Media/catalog_2023_0.pdf

International Headquarters **RAD Data Communications Ltd.**
24 Raoul Wallenberg Street, Tel Aviv 69719, Israel
Tel: 972-3-6458181 Fax: 972-3-7604732
email: market@rad.com www.rad.com

Distributed Network Functions Virtualization (D-NFV)



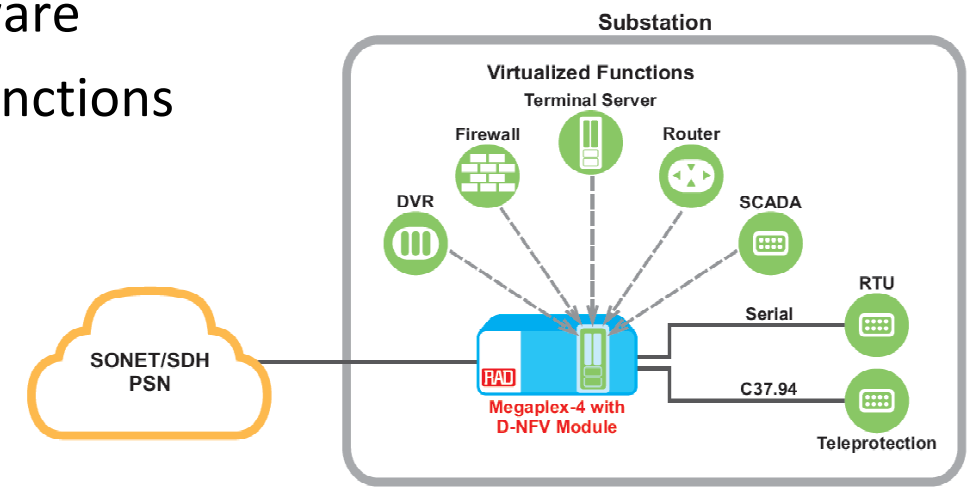
Virtualization

Highly configurable network hardware

Software defined network (SDN) functions

NID — Network Interface Device

NTU — Network Termination Unit for Carrier Ethernet



Carrier Ethernet

Ethernet

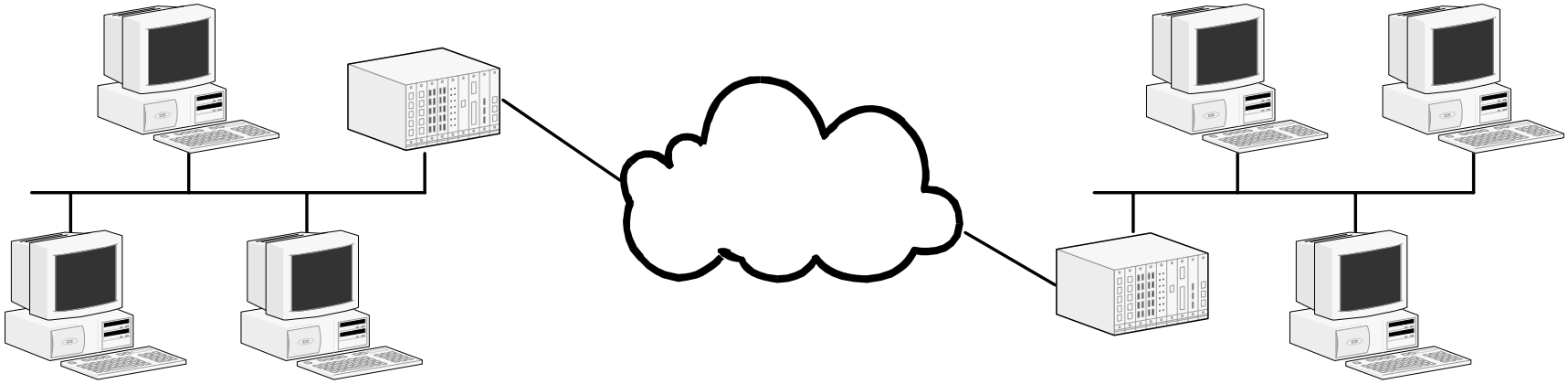
802.3 CSMA/CD shared medium local area (~ 100 m) network

Data rates — 10 Mbps to 100 Gbps on copper or optical fiber

Carrier Ethernet

Bridge Ethernet LAN segments over WAN

Operates as single Ethernet broadcast domain

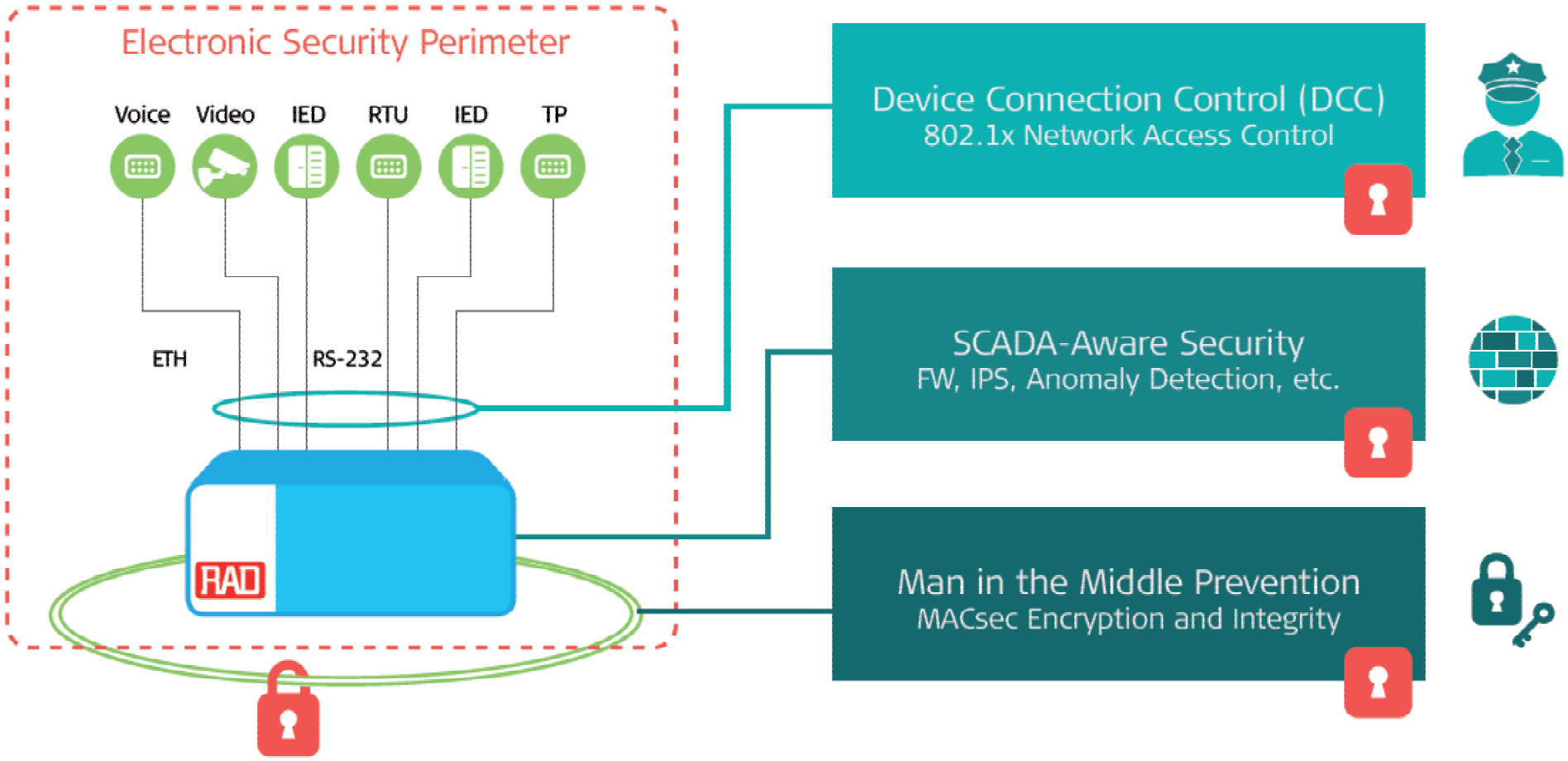


E-Line — point-to-point connection

E-LAN — general mesh

E-Tree — hierarchical tree

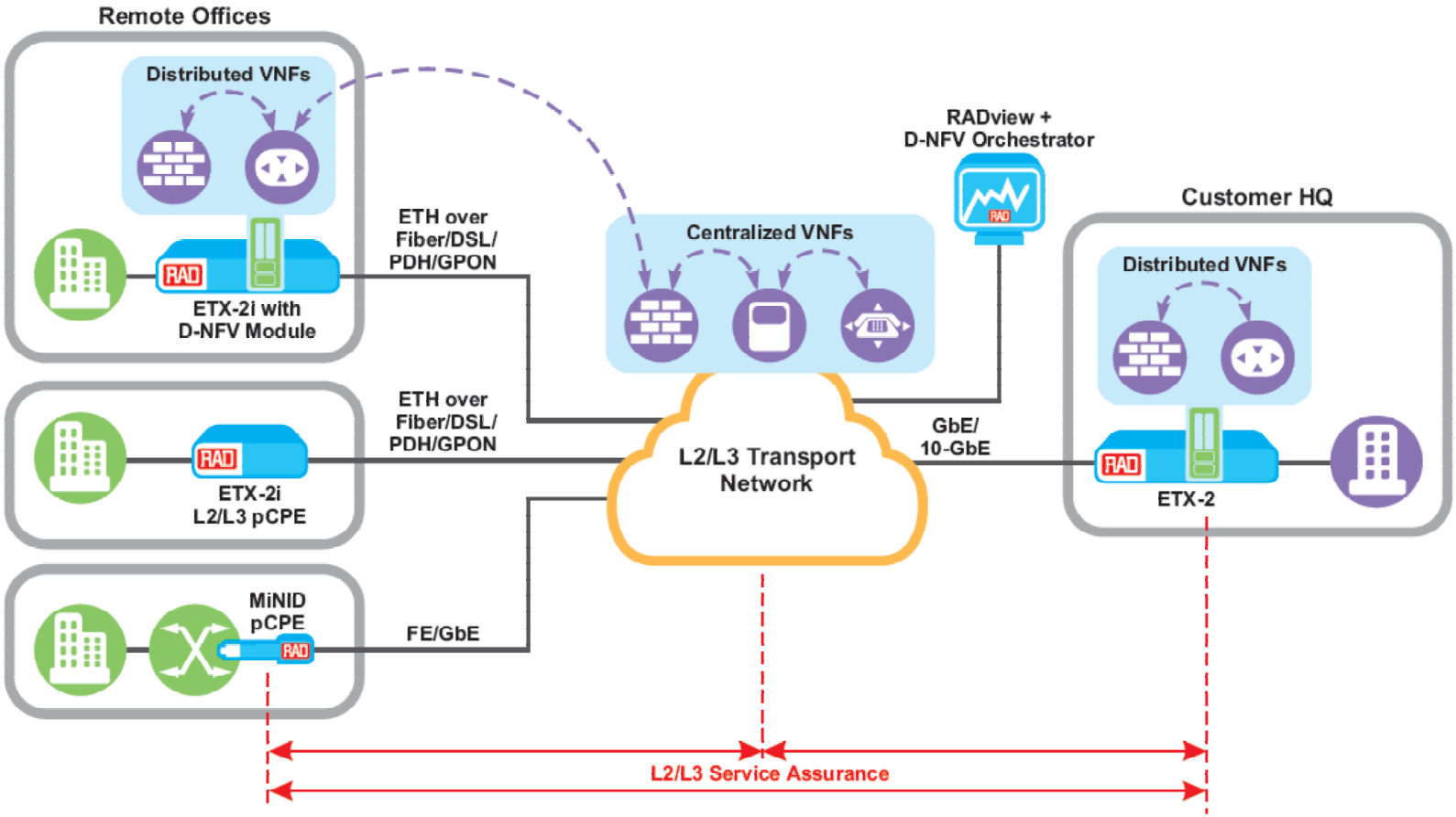
Cyber Security



DCC	Data Communications Channel Port Based Network Access Control (802.1X)
SCADA	Supervisory Control and Data Acquisition Remote Monitoring and Control
MACsec	MAC layer security standard (802.1AE)

IED	Intelligent Electronic Devices
RTU	Remote Terminal Unit
TP	Terminal Protocol (?)
ETH	Ethernet

vCPE (virtual customer premises equipment)



VNF — Virtualized Network Function

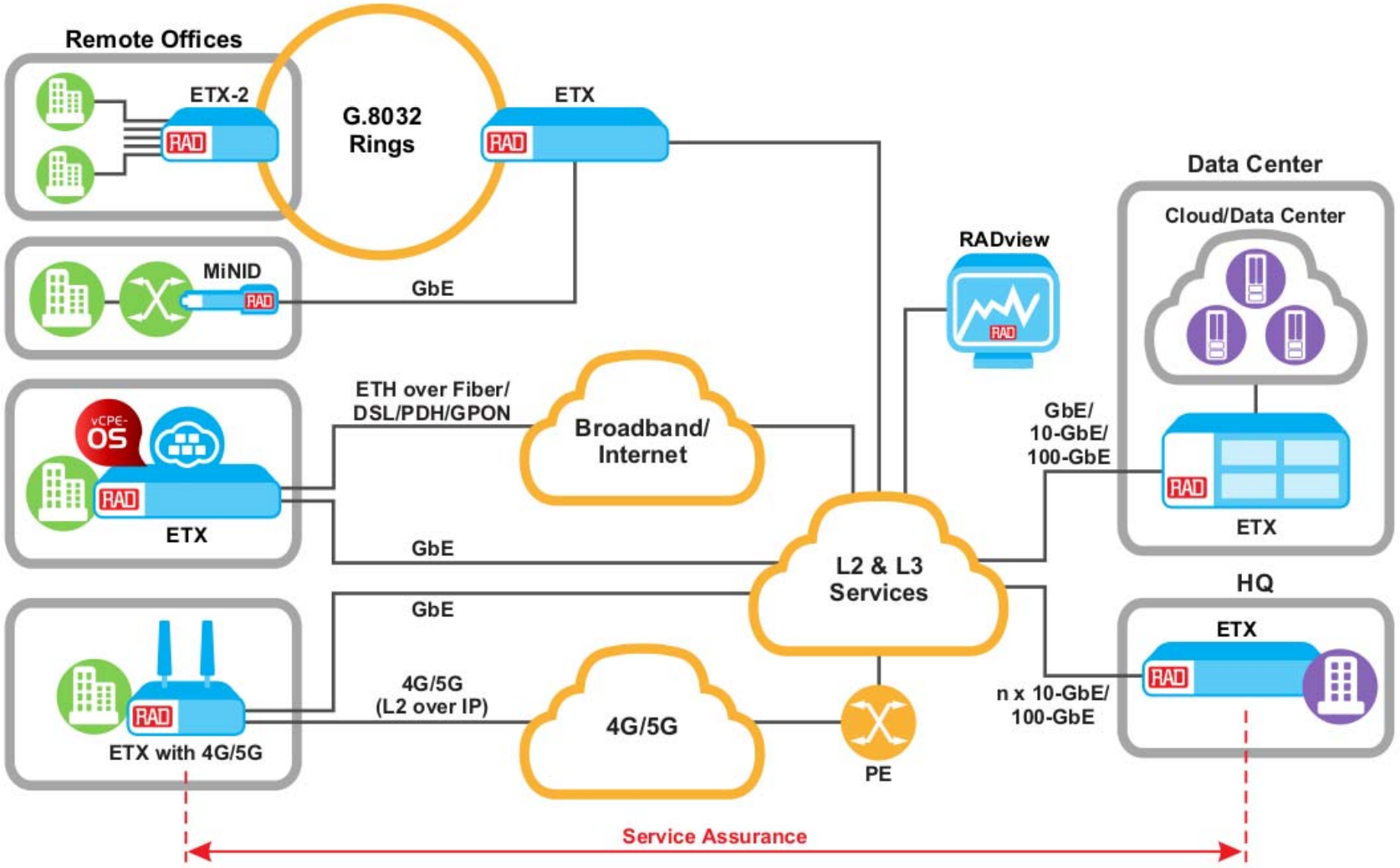
GPON — Gigabit Passive Optical Network

PDH — Plesiochronous Digital Hierarchy (standard telco multiplex hierarchy)

GbE — Gigabit Ethernet

L2/L3 — layer 2 (data link) switching + layer 3 (IP) routing

Carrier Ethernet



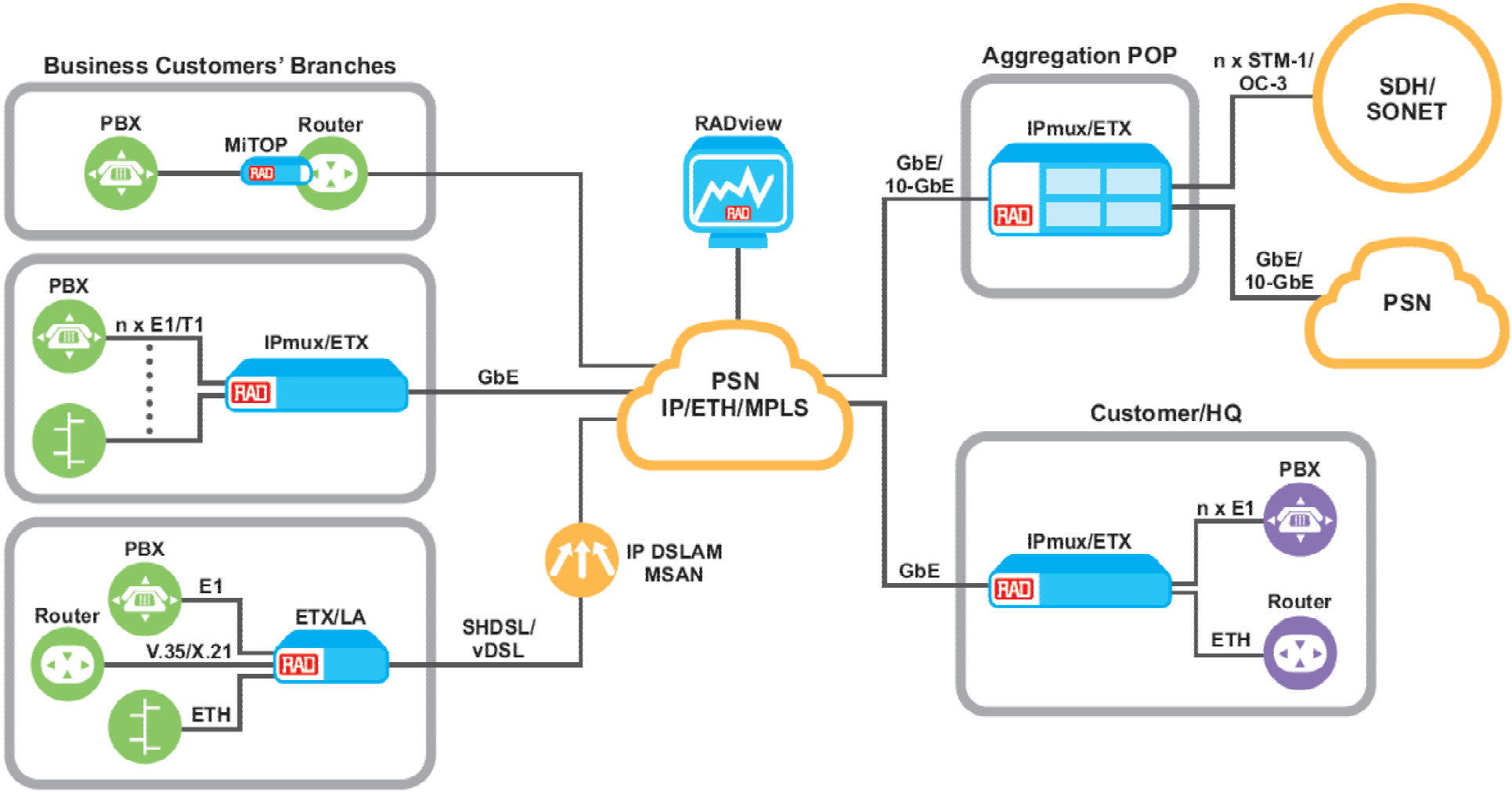
ETX

Carrier Ethernet demarcation and aggregation

MinID

login system

TDM Over Packet Mode Networks



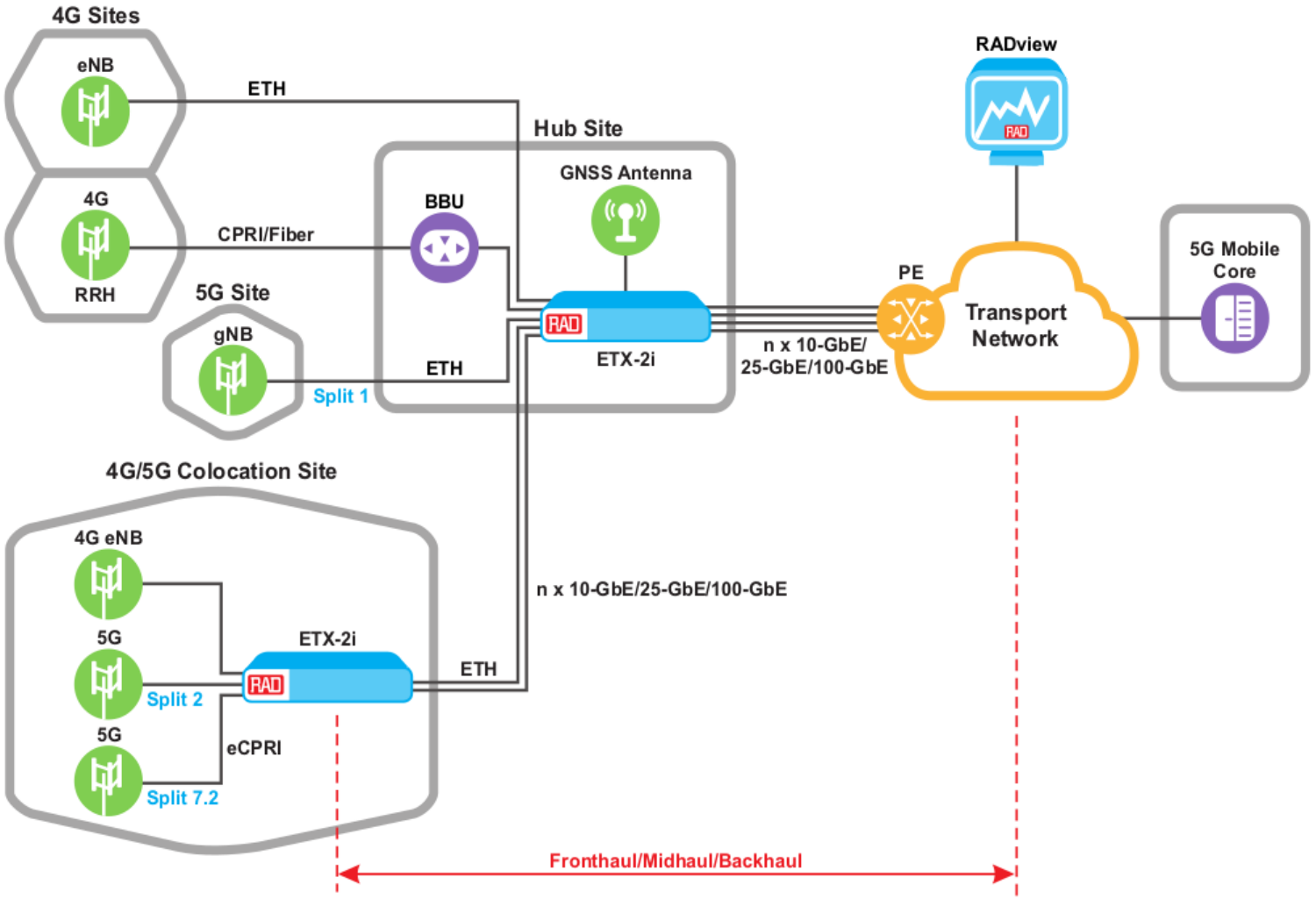
PSN

Packet Switched Network

PBX

Private Branch Exchange — telephone switchboard

Cellular Backhaul



Cell site

2G BTS or 3G/4G Node B (NB/eNB) connects mobile device to cellular network

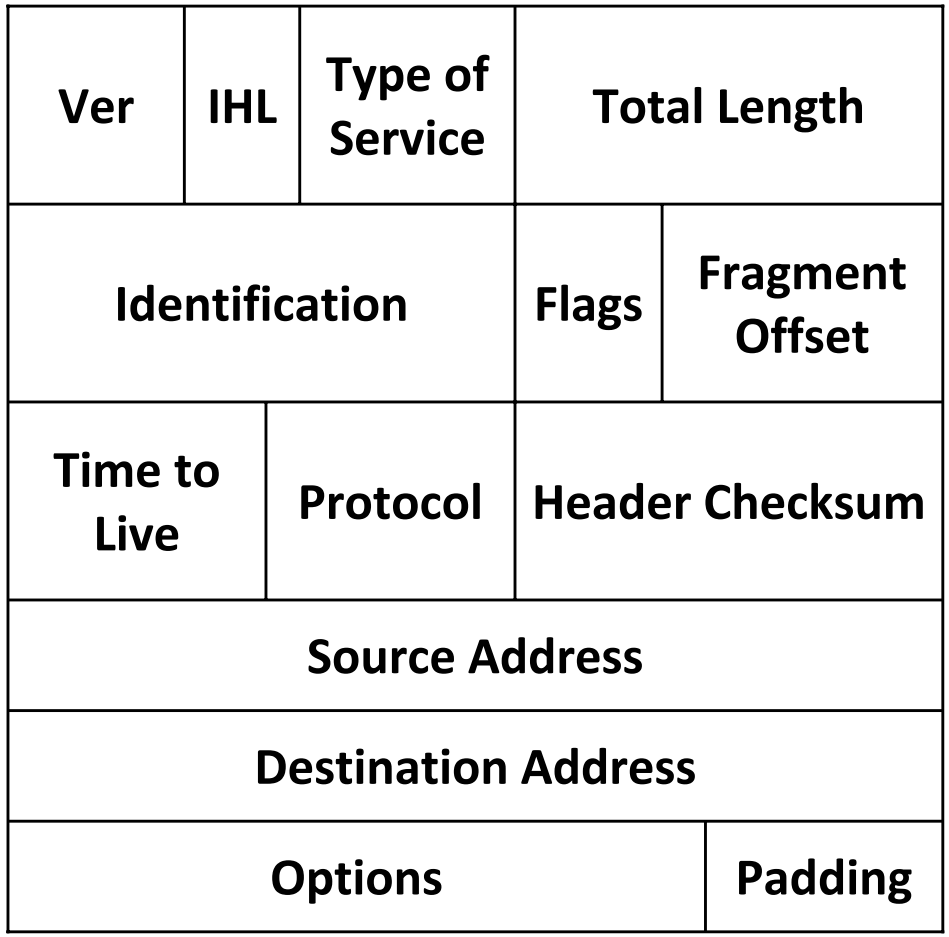
Backhaul

Carry digital voice from mobile cell site to telephone central office

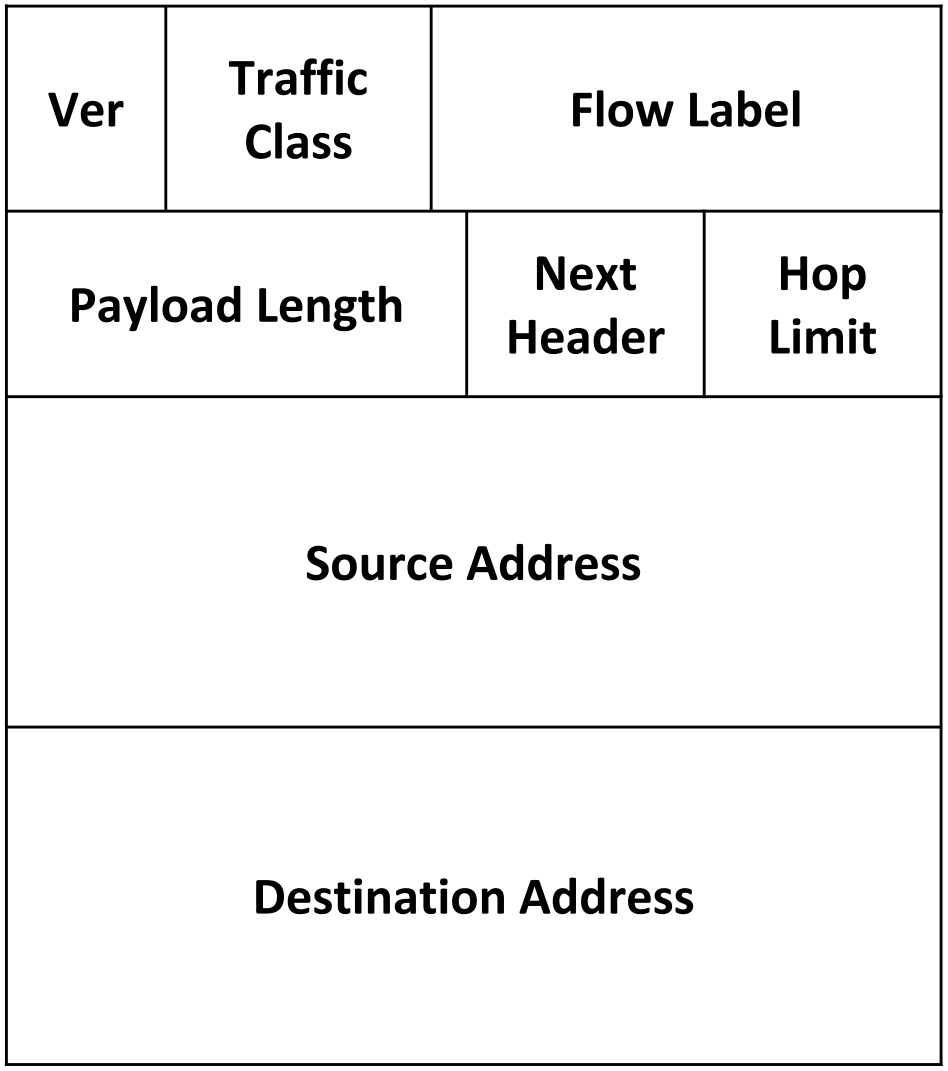
IPv6

IPv4 & IPv6 Header Comparison

IPv4 Header



IPv6 Header



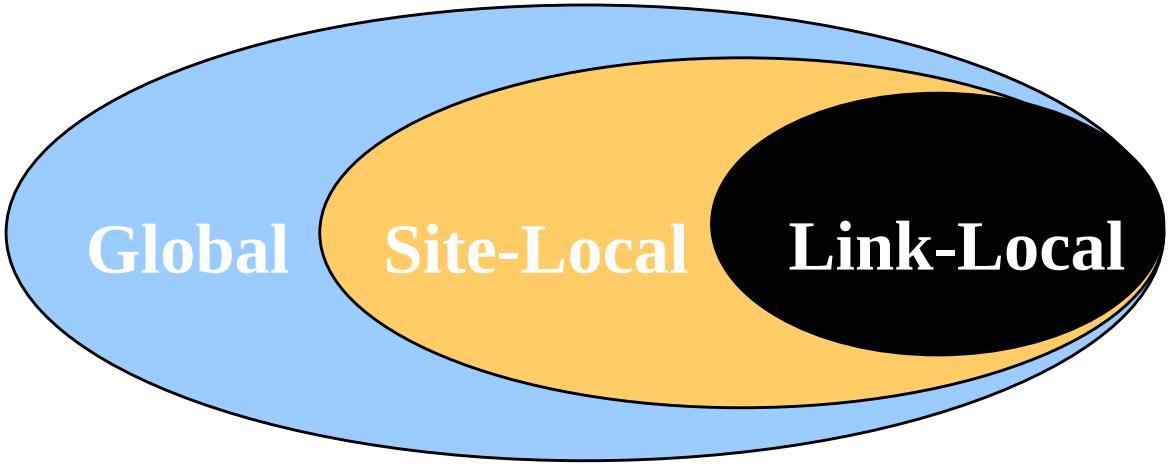
IPv6 Address Scope

Address assigned to interface

- Interface ID

Addresses depend on scope

- Link Local
  - Local hardware connection
  - Broadcast domain
- Site Local
  - Autonomous system (AS)
  - Network of one organization
- Global



Types of IPv6 Addresses

Unicast

One address on a single interface

Delivery to single interface

Multicast

Address of a set of interfaces

Delivery to all interfaces in set

Anycast

Address of a set of interfaces

Delivery to closest single interface in set

No broadcast addresses

IPv6 Address

128-bit address

340,282,366,920,938,463,463,374,607,431,768,211,456 addresses

50,000 addresses per square meter of land on Earth

Colon-separated 16-bit hex

2031:0000:130F:0000:0000:09C0:876A:130B

Leading zeros optional

2031:0:130F:0:0:9C0:876A:130B

Successive 16-bit 0 fields written :: (once)

legal **2031:0:130F::9C0:876A:130B**

illegal **2031::130F::9C0:876A:130B**

IPv4 compatible — used in tunneling IPv6 through IPv4

0:0:0:0:0:0:1.2.3.4 = ::0102:0304

IPv4 mapped — used by IPv6 source sending to IPv4 dest

0:0:0:0:0:FFFF:1.2.3.4 = ::FFFF:0102:0304

IPv6 Prefix

Prefix Format (PF)		Type	Application
Binary	**Hex**		
0000 0000	0::/8	Reserved address	Not assigned
0000 001	0200::/7	Reserved for ATM	Used in telephony
001	2000::/3	Aggregatable Global Unicast Address	Aggregation of host addresses into networks and subnets
1111 1110 10	FE80::/10	Link Local Address	Address hosts on LAN segment
1111 1110 11	FEC0::/10	Site Local Address	Addresses used with an AS (like 10.0.0.0 in IPv4)
1111 1111	FF00::/8	Multicast Address	Assigned to a group

Unicast Address Formats

Link Local

FP (10 bits)	Reserved (54 bits)	Interface ID (64 bits)
1111111010 FE80::/10	MUST be 0	MAC derived

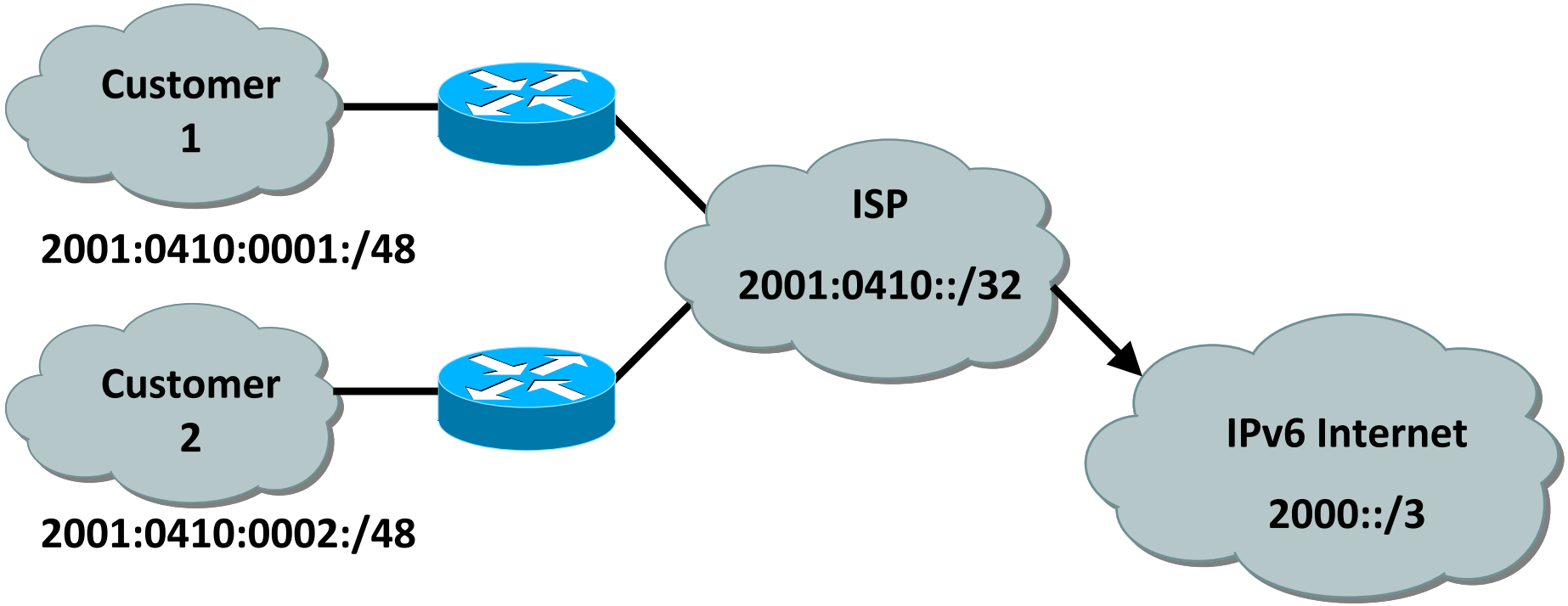
Site Local

FP (10 bits)	Subnet (38 bits)	Subnet (16 bits)	Interface ID (64 bits)
1111111011 FEC0::/10	Locally Administered		MAC derived

Global

FP (3 bits)	ISP assigned (45 bits)	Subnet (16 bits)	Interface ID (64 bits)
001 2000::/3	Provider Administered	Locally Administered	MAC derived or Locally Administered or Random

Hierarchical Addressing & Aggregation



Authority

Assigns **`2001:0410::/32`** to ISP

ISP

Assigns **`2001:0410:1:/48`** to customer 1

Assigns **`2001:0410:2:/48`** to customer 2

Customer 1

Assigns subnets **`2001:0410:1:1/64 , 2001:0410:1:2/64`**

Extension Headers

Next header field

Points to header following IPv6 header

Upper Layer Protocol	
6	TCP Header
17	UDP Header
41	IPv6 Header
45	Interdomain Routing Protocol
46	Resource Reservation Protocol
58	IPv6 ICMP Packet

IP Option Header	
0	Hop-by-Hop Options Header
43	IPv6 Routing Header
44	IPv6 Fragment Header
50	Encapsulating Security Payload
51	IPv6 Authentication Header
59	No Next Header
60	Destination Options Header

Extension headers

Multiple of 8 bytes

Syntax depends on option

Headers (except 60) appear only once

IPv6 Option Headers

Tunneling

Option 41

IPv6 datagram in data field of IPv6 datagram

Example — used with fragmentation when router adds option field

Hop-by-hop header

Options checked by every router and destination node

Destination options header

Options checked by destination node

Routing header

Source sets route through network

Fragment header

Handles fragmentation

Authentication header / Encapsulating Security Payload

Implement IPsec

Option Header Examples

IPv6 Header Next = TCP	TCP Header + Data

IPv6 Header Next = Routing	Routing Header Next = TCP	TCP Header + Data

IPv6 Header Next = Routing	Routing Header Next = Fragment	Fragment Header Next = TCP	TCP Header + Data Fragment

IPv6 Header Next = Routing	Hop-by-Hop Header Next = Fragment	Fragment Header Next = TCP	TCP Header + Data Fragment

Traffic Class + Flow

DSCP — 6 bits

- Differentiated Services Code Point
- Sets per-hop behavior according to service

ECN — 2 bits

- Explicit Congestion Notification
- Router can set congestion indication

Flow

- Stream of related packets from 1 source to 1 destination
- Require particular handling by routers
  - Requirements cached in routers
  - Example — real time priority
- 24 bit flow label
  - Chosen randomly from 1 to FFFFF
  - 0 — not part of a flow
- Flow identified by label + src IP + dest IP

ICMP — Internet Control Message Protocol

Offset	0 – 7	8 – 15	16 – 31
0	Type	Code (subtype)	Checksum
32	ICMP Message		

1	Destination Unreachable
2	Packet Too Big
3	Time (Hop Count) Exceeded
4	Parameter Problem

128	Echo Request
129	Echo Reply
130	Group Membership Query
131	Group Membership Report
132	Group Membership Reduction
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement
137	Redirect Message

Neighbor Discovery

Identify hosts and routers on physical LAN segment

Replaces ARP

Address resolution

Obtain MAC address for neighbors

Neighbor Solicitation

Node sends ICMP neighbor solicitation message to host

64-bit local link multicast prefix	64-bit host address
FF02::	Interface ID

Neighbor advertisement

Neighbor sends ICMP neighbor advertisement with MAC address

Router and Prefix Discovery

Router advertisements

- Routers send ICMP messages to hosts on some schedule
- Addressed to FF02::1
  - Multicast address — all systems on local link
- Provides configuration parameters
  - MTU
  - Global IP prefix
  - IP address configuration method (ICMP / stateless)

Router solicitation message

- Host sends ICMP message to request router service
- Addressed to FF02::2
  - Multicast address — all routers on local link
- Routers respond with router advertisement

Stateful Auto-Configuration

DHCP service

- Similar to IPv4

DHCP messages

- Solicit
- Advertise
- Request
- Reply
- Release
- Reconfigure

DHCP client

- Waits for DHCP advertisement
- Sends DHCP request

DHCP server

- Sends DHCP reply
- Provides IP address + configuration parameters

Stateless Auto-Configuration

Node reads 48-bit MAC address from hardware

Converts 6-byte MAC address to 8-byte node address

48-bit MAC address ⟶ 24 bits FFEE 24 bits

Example: 11 22 33 44 55 66 ⟶ 11 22 33 FF EE 44 55 66

Extends link-local prefix to 64 bits

FE80::/10 ⟶ FE80::/64

Attaches link-local prefix to node address

Creates temporary link-local unicast address

Example: FE80::1122:33FF:EE44:5566

Verifies unique address with neighbor solicitation

No response to link-level address ⟶ assigns address to interface

Sends router solicitation

Routers respond with router advertisement

No response ⟶ node attempts DHCP

Response message provides parameters — MTU, global prefix

Replaces FE80::/64 ⟶ global prefix to form global unicast address

Mobile IP (RFC 2002)

Internet protocol supporting host mobility

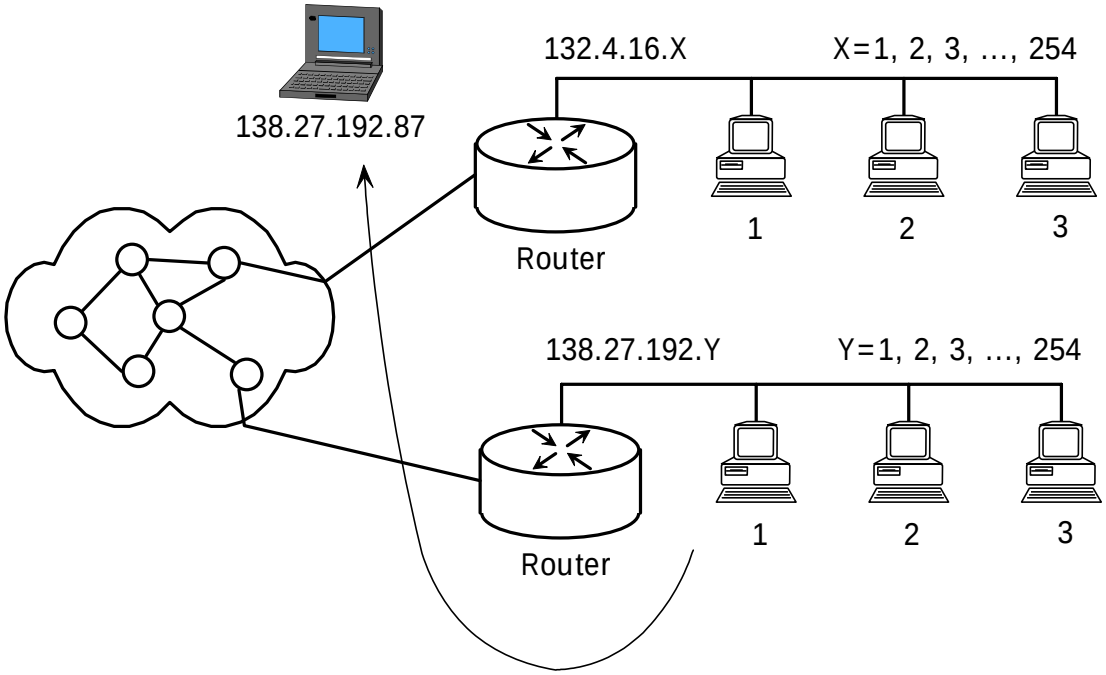
Maintains TCP connections as host changes location

Supports authentication

Mobile host maintains single long-term IP address

Version 4 IP address and address format are unaffected

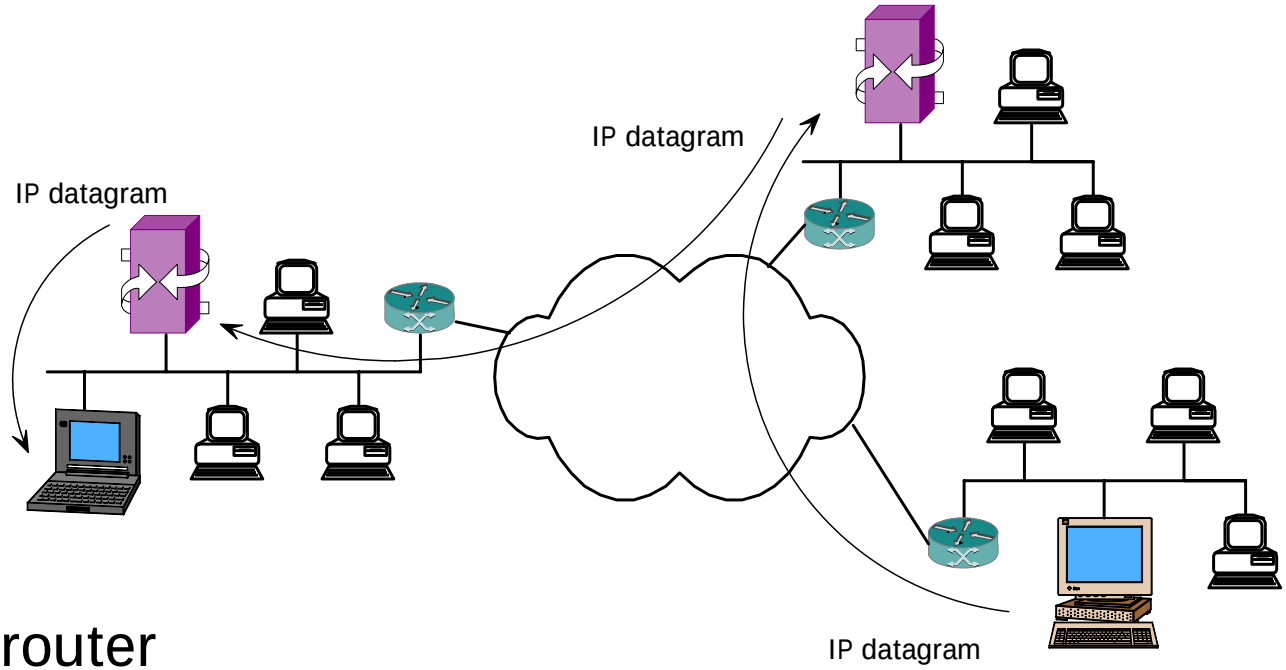
Routing tunnel replaces standard IP routing



Mobile IP — Basic Idea

Similar to

- Call forwarding
- Mail forwarding at post office



Mobile Node

- Permanent IP address
- Usual home service router
- Finds mobile service routers when roaming

Home router and mobile service router coordinate

IP datagrams

- Sent to usual home router
- Forwarded by home router to mobile service router
- Forwarded to Mobile Node

Mobile IP Entities

Mobile Node (MN)

Host or router that can change its point of attachment

Home Address

Permanent IP address assigned to MN

Correspondent Node (CN)

Node that sends datagrams to MN home address

Home Agent (HA)

Maintains table of registered mobile nodes

Forwards datagrams addressed to mobile node

Foreign Agent (FA)

Delivers datagrams between MN and HA

Mobility Agent

HA or FA supporting mobility

Care-of-Address (COA)

FA address used to identify current location of MN

Agent Discovery

Mobility Agents broadcast availability

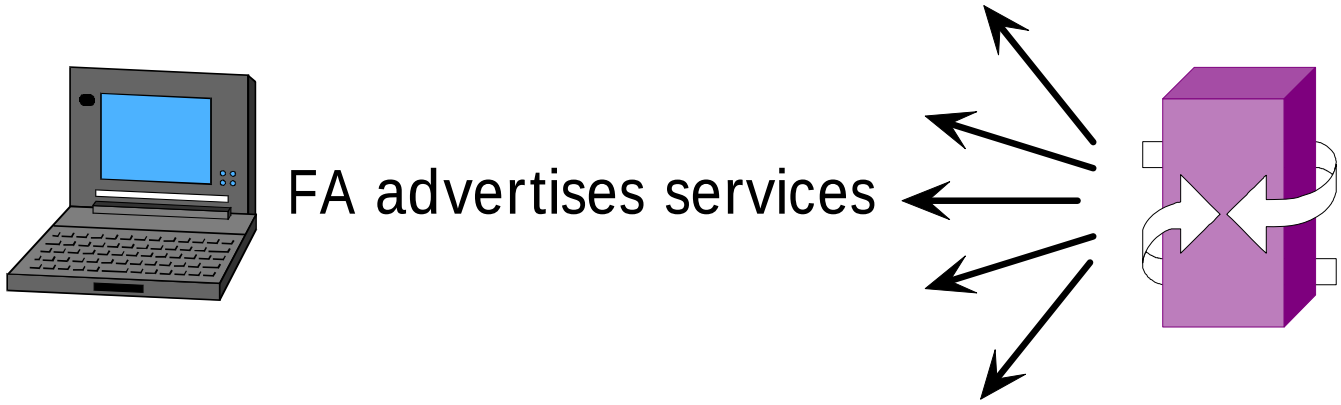
Home Agents (HA)

Foreign Agents (FA)

Mobile Node (MN) looks for

Local router services when connected to home network

Foreign Agent (FA) services when connected to foreign network



Care-of-Address Discovery

Mobile Node (MN)

Requests service from Foreign Agent (FA)

Foreign Agent

Assigns Care-of-Address (COA) to MN

FA can have 1 or more available COAs

Usually FA assigns same COA to all MNs



Registration

Foreign Agent

Requests Mobile IP support from MN's Home Agent

Home Agent

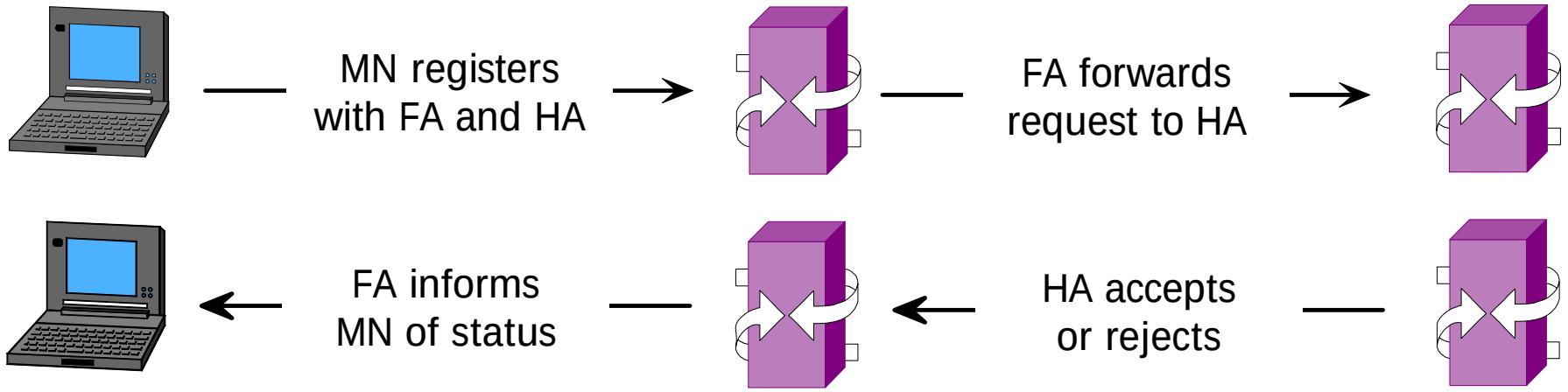
Agrees to provide Mobile IP support

MN

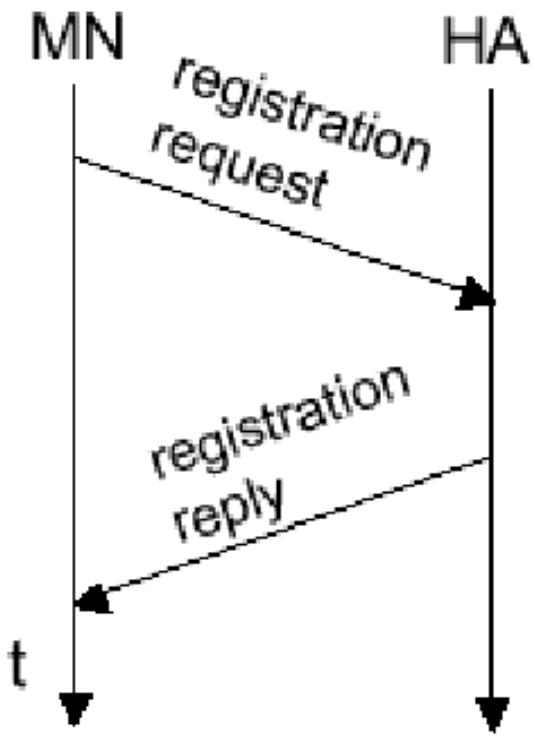
Registers COA with its HA

HA forwards datagrams to FA

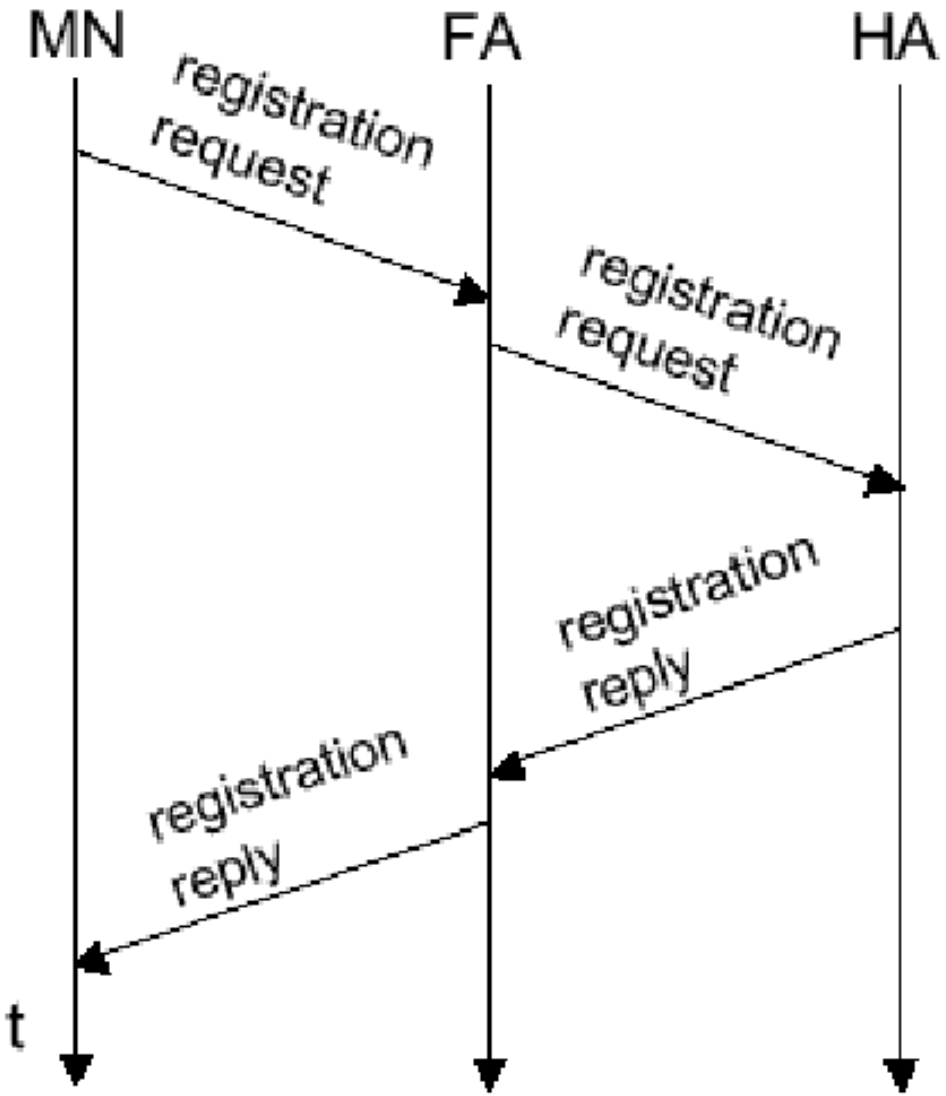
Datagrams for MN arrive at HA



Establishment of Service in Mobile IP

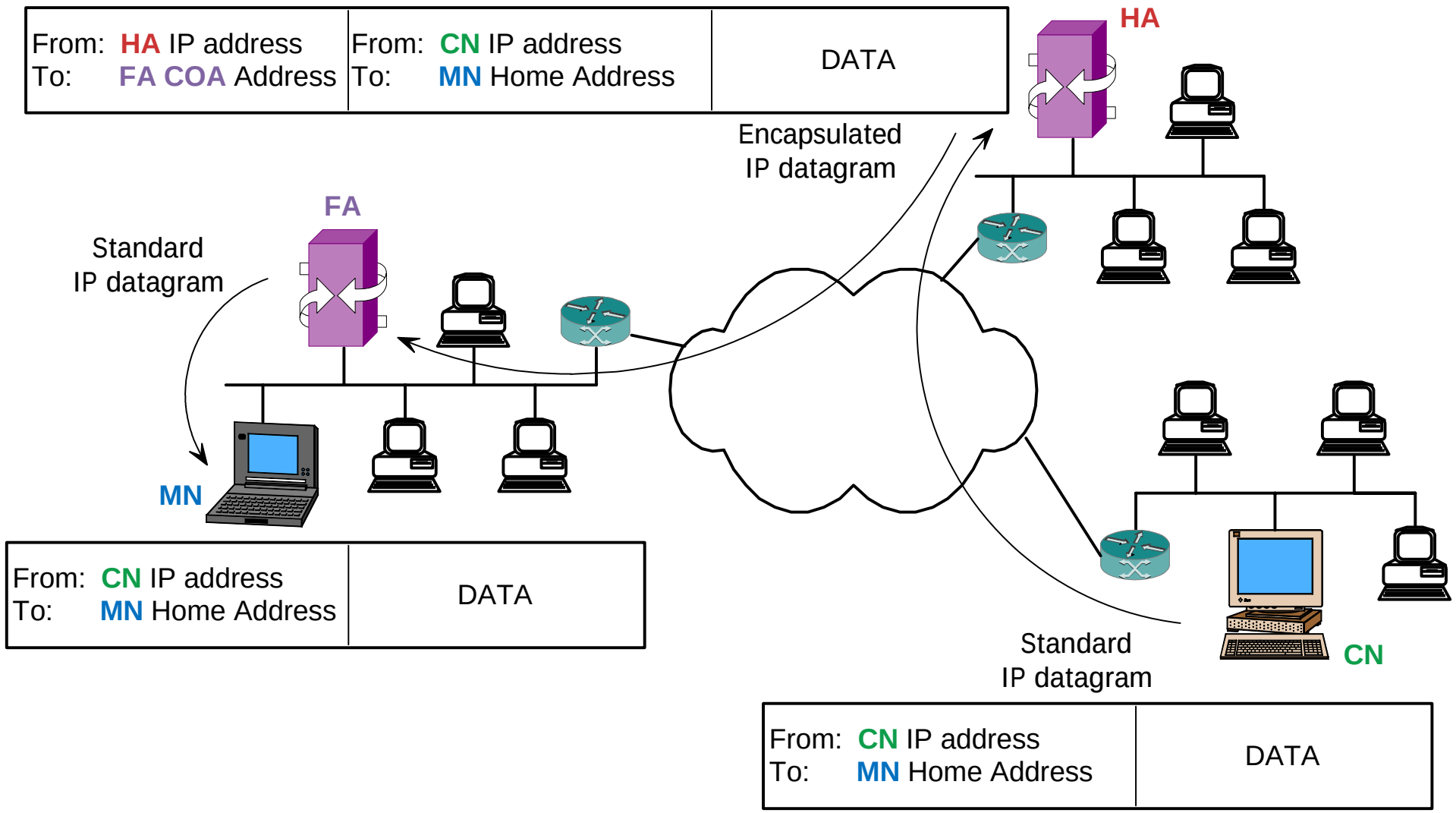


MN in home network



MN in foreign network

Mobile IP End-to-End Delivery



Roaming and Hand-Off

Roaming MN

Moves to new attachment point (network)

Requires change of FA

Multiple bindings

Multiple COAs — old + new

Avoid datagram loss

Avoid too frequent registrations

HA

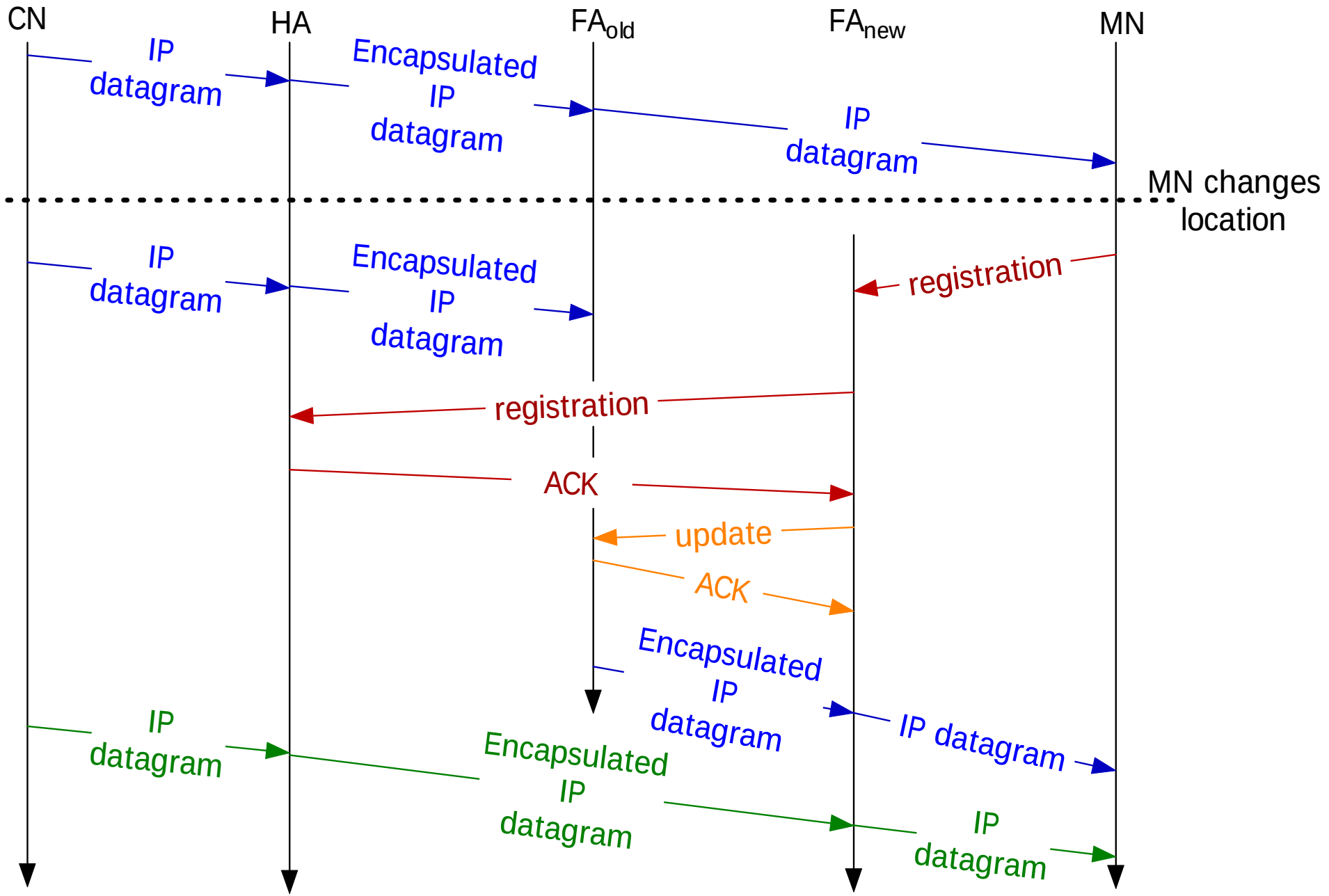
Forwards each packet to multiple COAs

MN

Receives packet at one COA

Route optimization after reconnection

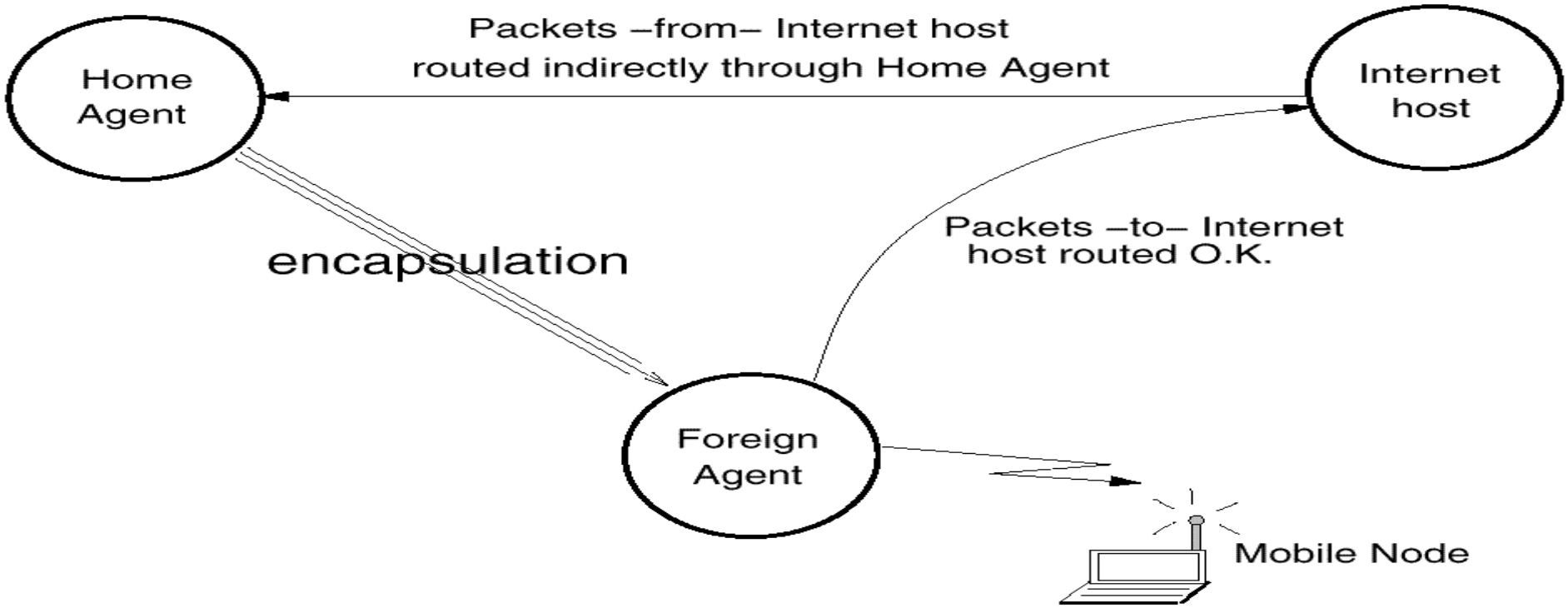
Change of Foreign Agent



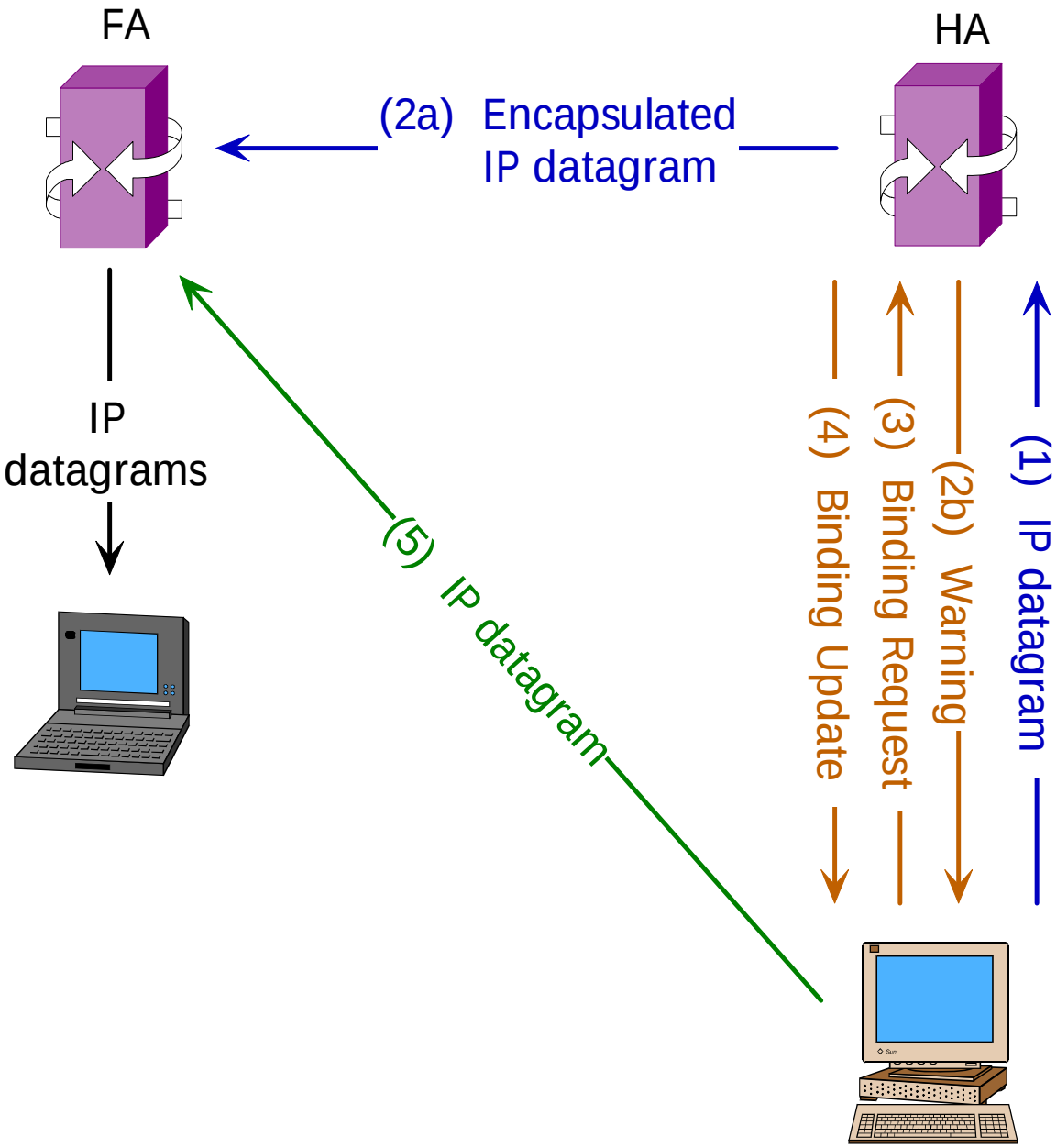
Triangle Routing

Home agent is bottleneck

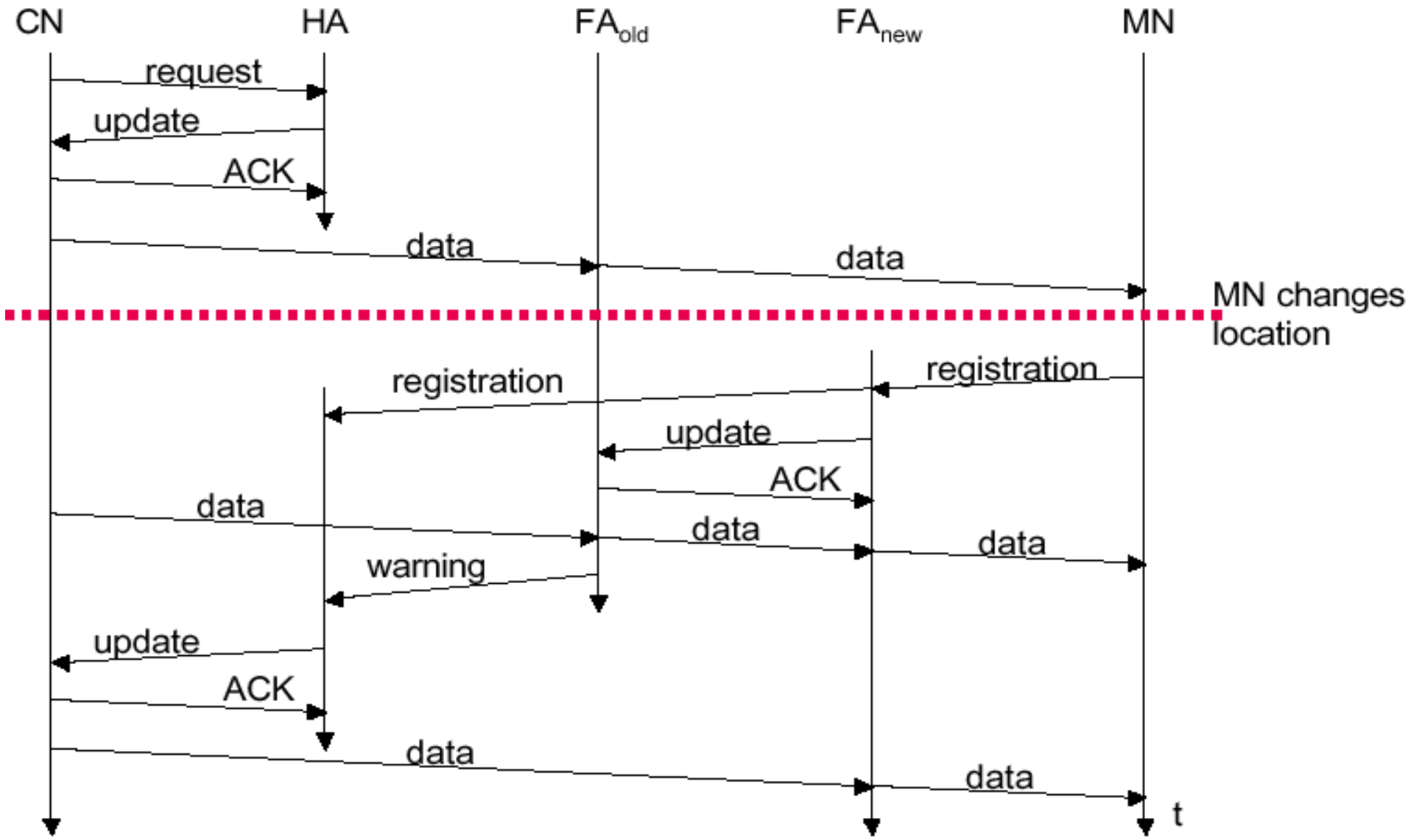
Increases network load



Route Optimization



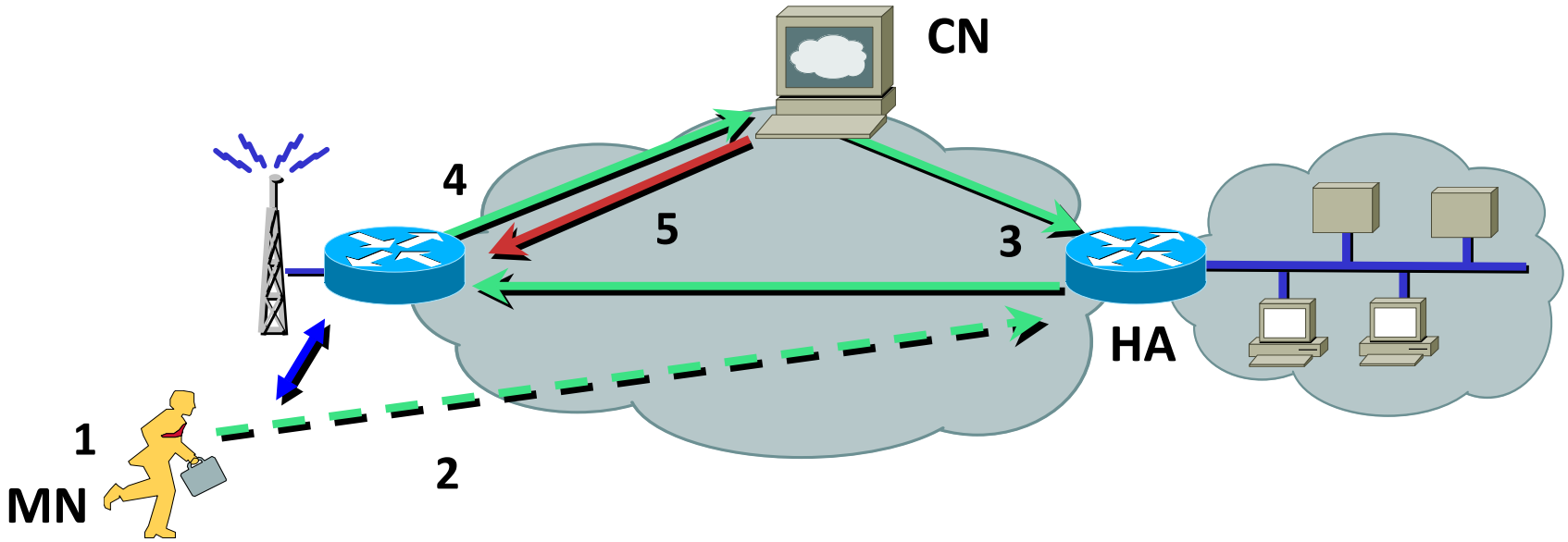
Roaming Under Route Optimization



IPv6 Mobile IP

Implements Mobile IP

1. Mobile node (MN) obtains local address using autoconfiguration
   Roaming address = care-of-address (CoA)
   No special Foreign Agent
2. MN registers with Home Agent by sending Binding Update
3. HA forwards traffic for registered MN
   Tunnels packets from CN to MN
4. MN sends packets to CN directly
5. Route optimization — HA provides CN with CoA



IPv6 Mobility Support

Node writes home address in destination option header

Destination node can identify datagram by home address

Tunneling

Using IPv6 routing extension headers instead of encapsulation

Reduces processing cost of delivering packets

Handover

Node moves from ESS to ESS

Layer 2 handover — change AP and ESS ID

Node detects change in on-link subnet prefix

Updates CoA

IPv6 Mobility header messages

Home Test Init, Home Test, Care-of Test Init, and Care-of Test

Binding Update / Acknowledgement

MN to notifies node or HA of current binding

Basics of Wireless Networking

Energy and Power

Energy

The ability to do work

Energy can be kinetic (movement) or potential (stored)

Power

Energy transfer per second

Transfer can be kinetic (motion) or potential (moving stored energy)

Units

Power is measured in Watts

Energy is measured in Joules = Watts $\times$ seconds

1 kW-hour = 1000 Watts $\times$ 3600 seconds/hour

= 3.6×10^6 Joules

Electricity and Magnetism

A charged object may create

Electric field $\mathbf{E}$

Magnetic field $\mathbf{B}$

Radiation (transfer of power) from a charged object

Power $\propto \mathbf{E} \times \mathbf{B}$

Motionless charge does not radiate

$$\mathbf{E} = -\frac{\text{charge}}{R^2}, \qquad \text{at distance } R$$

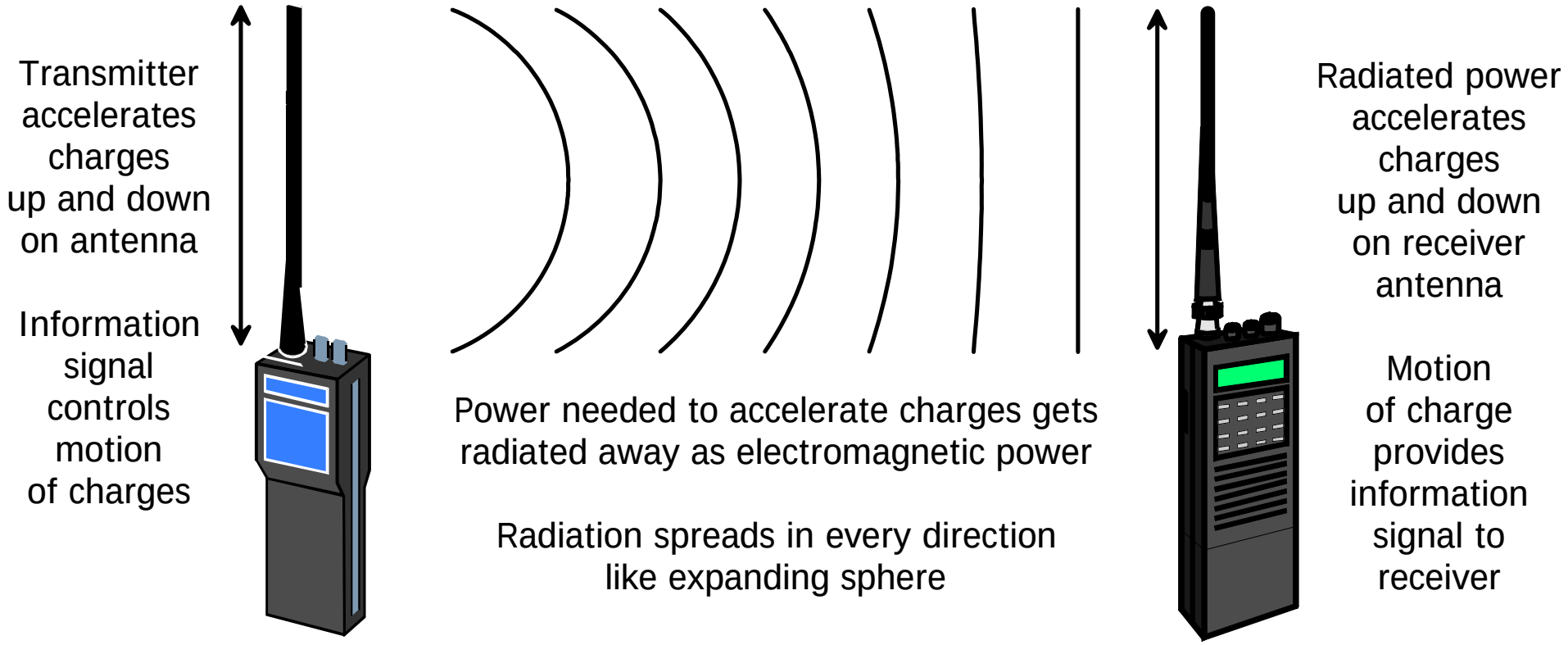
$$\mathbf{B} = 0$$

Antenna accelerates charges = electric current

Accelerated charges induce fields $\mathbf{E} \neq 0$ and $\mathbf{B} \neq 0$

Antenna radiates power as electromagnetic waves

Radio Communication



Moving electric charge is called electric current

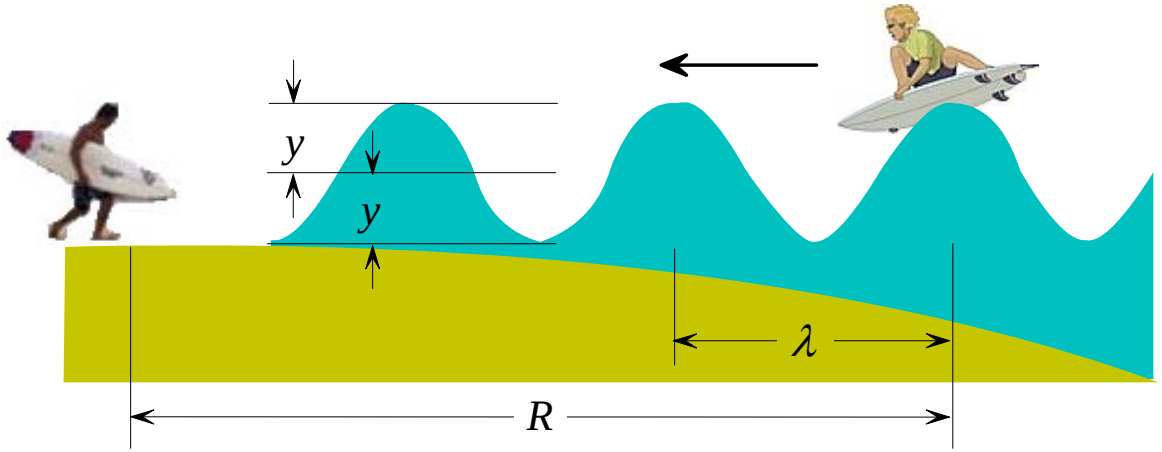
Current depends on time $\Rightarrow$ charges must accelerate

Electromagnetic radiation satisfies wave equation

Radiated power depends on time t and distance R from antenna

Wave Motion

Ocean waves rolling onto a beach



Wave height has peaks and troughs

y = height of peak above center = depth of trough below center

At fixed distance from shore, wave rises and falls over time

T = time between two wave peaks (period)

$f = 1/T$ = number of wave peaks per second (frequency)

At fixed time, multiple wave peaks at various distances

λ = distance between two wave peaks (wavelength)

Surfer rides peak of wave

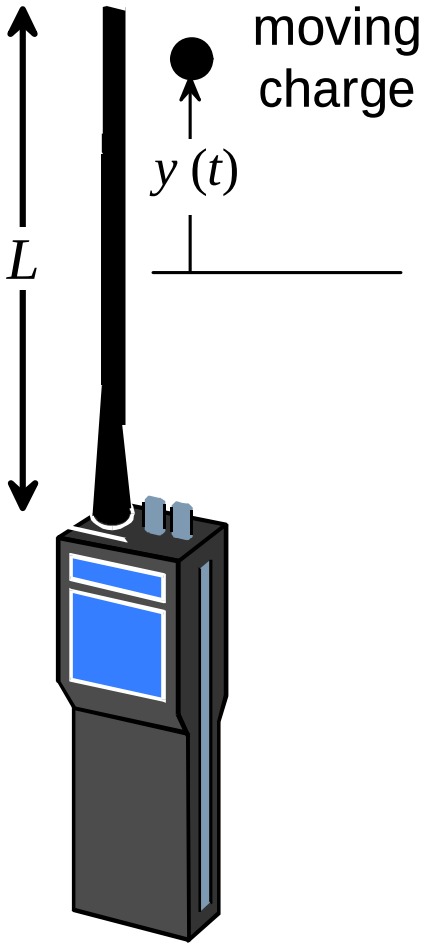
Peak depends on distance and time $\Rightarrow$ peak moves over time

Speed of moving peak = $f \times \lambda$

Charge Moving on Antenna

Charge on antenna accelerated up and down

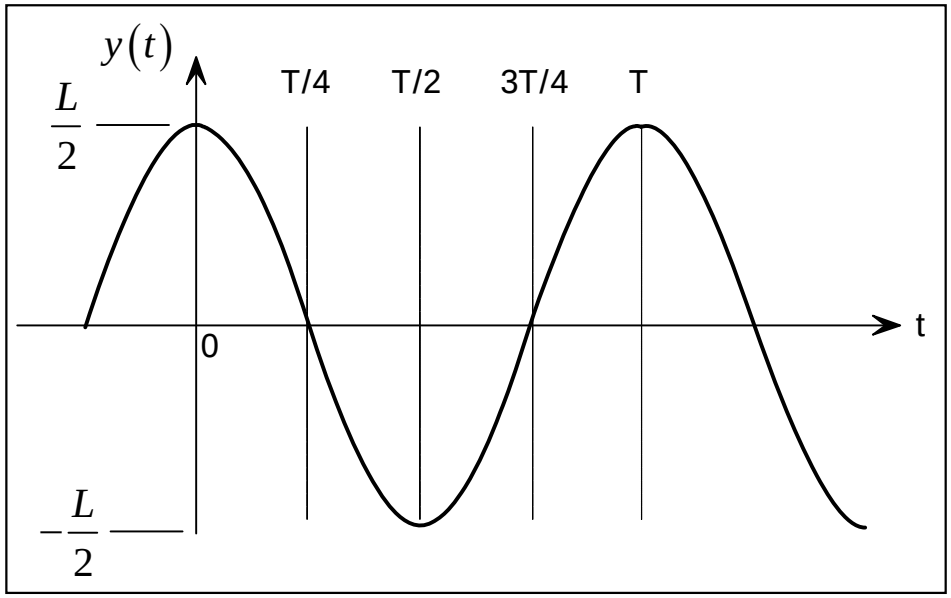
Oscillates top to bottom (distance L) every T seconds



Frequency $f = 1/T$ oscillation cycles per second

$y(t)$ = position of charge on antenna at time t

$$y(t) = \frac{L}{2}\cos\left(2\pi\frac{t}{T}\right) = \frac{L}{2}\cos(2\pi ft)$$



Field is Solution to Maxwell Equations

Radiation fields

$$\mathbf{E}(R,t) = \frac{\mathbf{E}_0 \cos\left[2\pi f\left(R/c - t\right)\right]}{R} \qquad \mathbf{B}(R,t) = \frac{\mathbf{B}_0 \cos\left[2\pi f\left(R/c - t\right)\right]}{R}$$

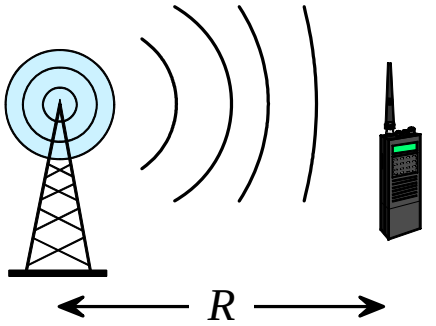
R = distance from antenna to point of measurement

t = time (measured on some clock)

f = frequency $= 1/T$

$\mathbf{E}_0, \mathbf{B}_0$ are physical constants

c = speed of light

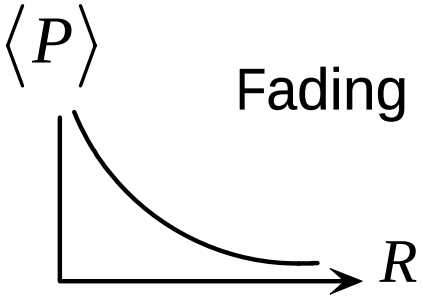


Radiated power

$$\mathbf{P} = \mathbf{E} \times \mathbf{B} = \frac{\mathbf{E}_0 \times \mathbf{B}_0}{R^2} \cos^2\left[2\pi f\left(R/c - t\right)\right]$$

Average power

$$\langle P \rangle = \frac{1}{T}\int_0^T \left|P(t)\right| dt = \frac{\left|\mathbf{E}_0 \times \mathbf{B}_0\right|}{2}\frac{1}{R^2}$$



Wavelength

Define electromagnetic wavelength

$$\lambda = cT = {}^{c}\!/_{f} \qquad \Rightarrow \qquad \lambda f = c$$

Radiation field

$$\mathbf{E}(R,t) = \frac{\mathbf{E}_0 \cos\left[2\pi f\left({}^{R}\!/_{c} - t\right)\right]}{R} = \frac{\mathbf{E}_0 \cos\left[2\pi\left(\frac{f\,R}{c} - ft\right)\right]}{R} = \frac{\mathbf{E}_0 \cos\left[2\pi\left(\frac{R}{\lambda} - \frac{t}{T}\right)\right]}{R}$$

Wave peaks

$$\cos\left[2\pi\left(\frac{R}{\lambda} - \frac{t}{T}\right)\right] = 1 \Rightarrow \left(\frac{R}{\lambda} - \frac{t}{T}\right) = 0,1,2,\ldots$$

$$\left(\frac{R}{\lambda} - \frac{t}{T}\right) = 0 \rightarrow \left(\frac{R+\Delta R}{\lambda} - \frac{t+\Delta t}{T}\right) = 0 \Rightarrow \left(\frac{\Delta R}{\lambda} - \frac{\Delta t}{T}\right) = 0$$

Speed

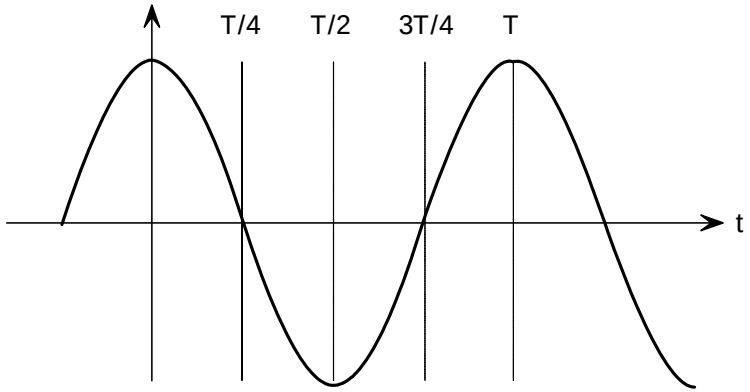
$$\mathrm{v} = \frac{\Delta R}{\Delta t} = \frac{\lambda}{T} = \lambda f = c$$

Wave peaks travel at speed of light

Spherical Waves in Space and Time

At fixed distance wave rises and falls over time

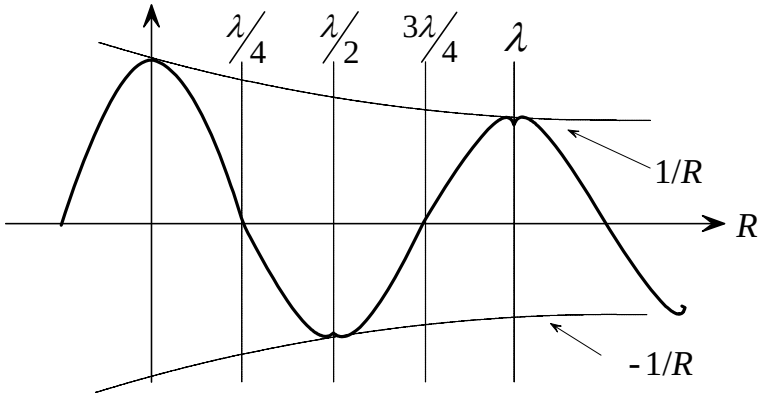
$$\mathbf{E}\left(R_{const}, t\right) = \frac{\mathbf{E}_0}{R_{const}} \cos\left[2\pi\left(\frac{R_{const}}{\lambda} - \frac{t}{T}\right)\right]$$
$$= C_1 \times \cos\left(2\pi ft - C_2\right)$$



At fixed time, multiple wave peaks at various distances

Wave peaks decrease at larger distances from source

$$\mathbf{E}\left(R, t_{const}\right) = \frac{\mathbf{E}_0}{R} \cos\left[2\pi\left(\frac{R}{\lambda} - \frac{t_{const}}{T}\right)\right]$$
$$= \frac{C_1}{R} \cos\left[2\pi\left(\frac{R}{\lambda} - C_2\right)\right]$$



Electromagnetic Spectrum

Radio antennas are effective in the frequency range

from ~ 30 kHz (λ = 10 km)

to ~ 300 GHz (λ = 1 mm)

Chemical reactions generate higher frequencies:

Infra-Red (IR)

Visible Light

Ultra-Violet (UV)

X-rays (Roentgen)

Nuclear reactions generate gamma rays (γ)

Electromagnetic Spectrum

Wavelength (m)	10^{4}	10^{2}	10^{0}	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}	10^{-14}	10^{-16}
Frequency (Hz)	10^{4}	10^{6}	10^{8}	10^{10}	10^{12}	10^{14}	10^{16}	10^{18}	10^{20}	10^{22}	10^{24}
	radio		microwave		IR	visible	UV	X-ray		gamma	

$\lambda f = c = 3 \times 10^{10}$ cm/s

1 MHz ~ 300 m
100 MHz ~ 3 m
10 GHz ~ 3 cm

VLF	< 30 kHz
LF	30 - 300 kHz
MF	300 kHz - 3 MHz
HF	3 - 30 MHz
VHF	30 - 300 MHz
UHF	300 MHz - 3 GHz
SHF	3 - 30 GHz
EHF	> 30 GHz

Example

Line antenna most efficient when $L = \lambda / 2$

GSM cellphones operate at f ~ 1 GHz

$\lambda = (3 \times 10^{10}$ cm/sec)/(10^{9} Hz) = 30 cm $\Rightarrow$ L ~ 15 cm = phone size

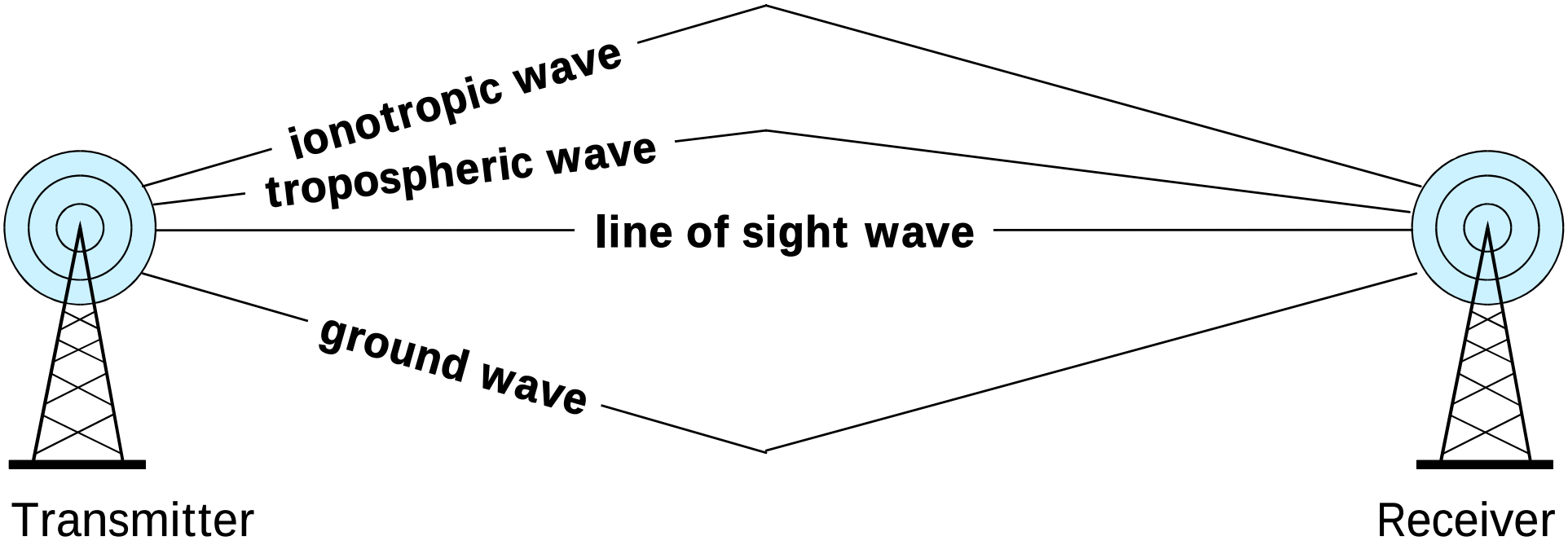
Radio Wave Propagation

Transmitter generates radio waves

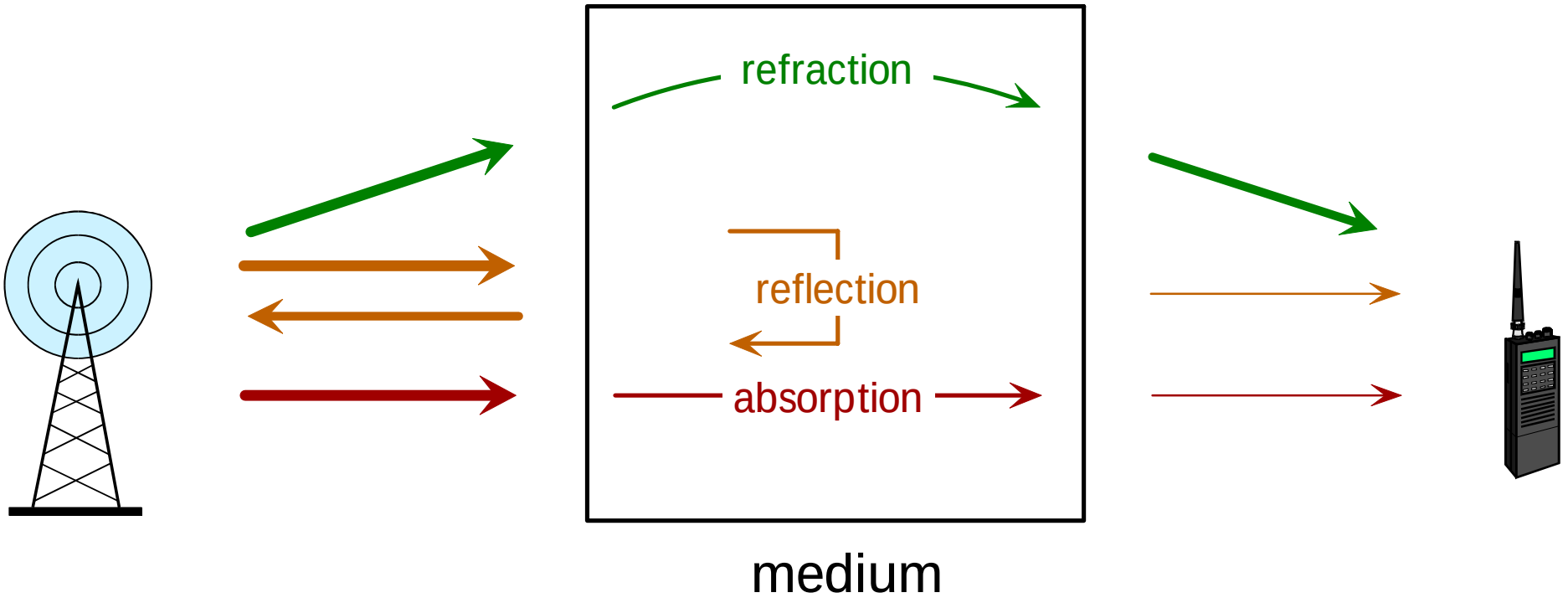
Waves propagate (spread out) through space

Part of radiated power may be obstructed

Part of radiated power is detected by receiver



Interference with Radio Signals



Multipath Fading

Obstacles reflect radio waves

Receiver gets signals from multiple paths

Time-to-arrive depends on path taken by signal

Receiver gets signals transmitted at different times

Example

Three signals sent at times $t_1 < t_2 < t_3$

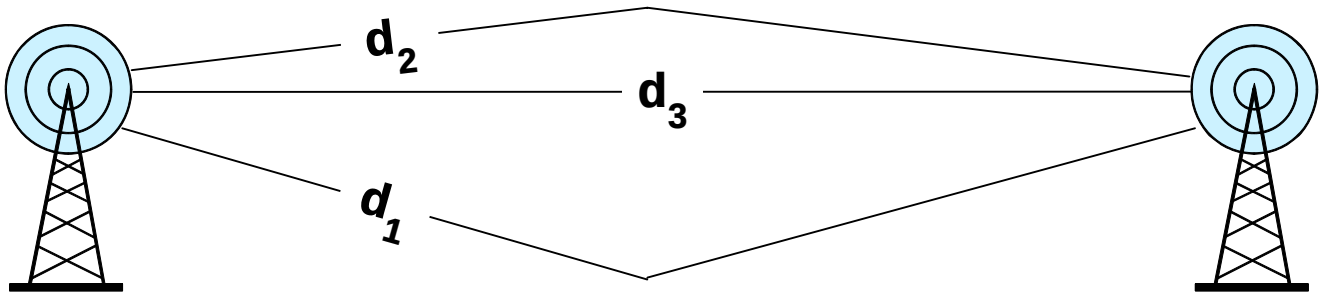
Antenna receives all three signals at time t

Signal 1 — sent first and followed longest path d_1

Signal 2 — sent second and followed second longest path $d_2 < d_1$

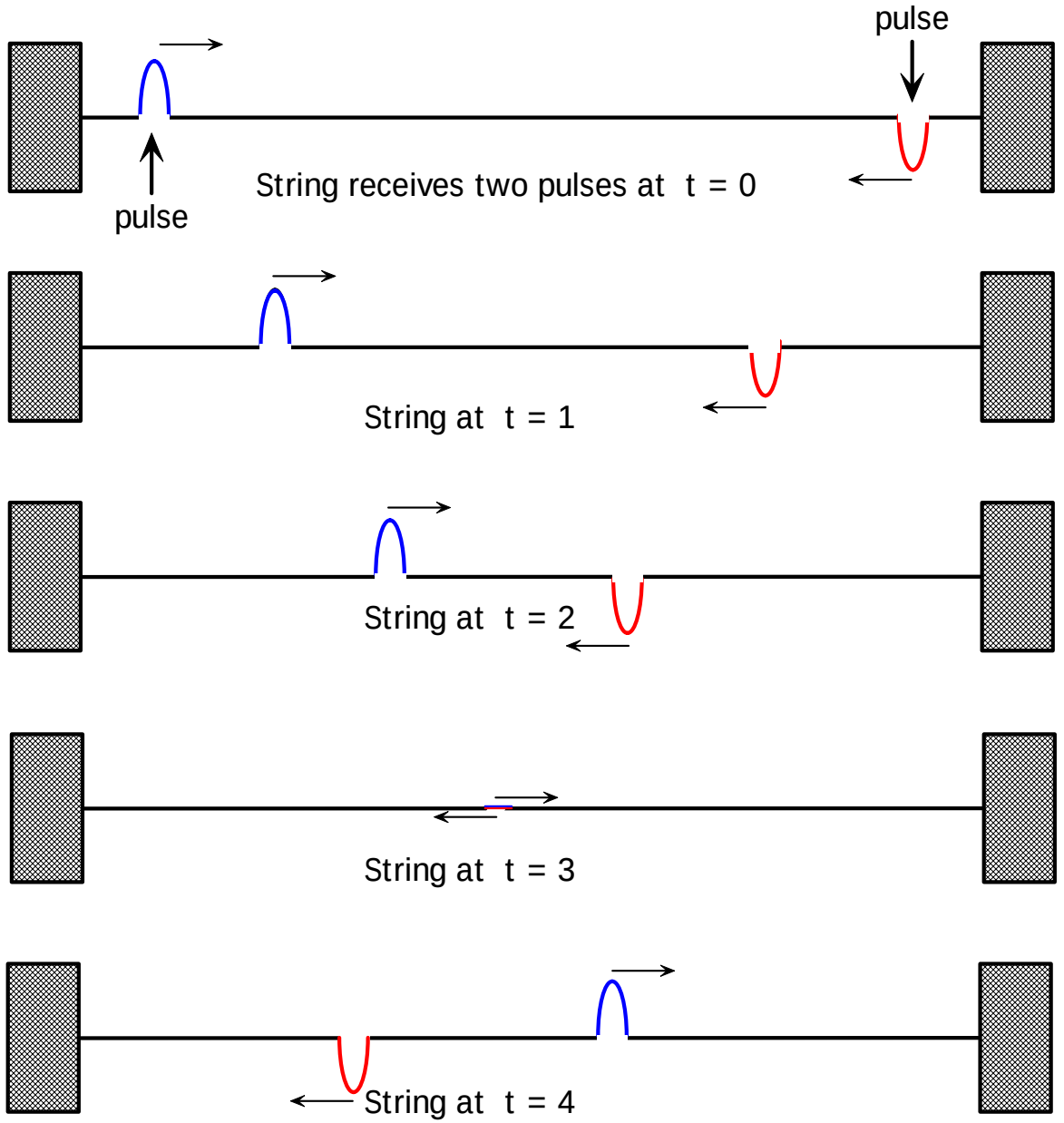
Signal 3 — sent last and followed shortest path $d_3 < d_2$

Sum of waves can cancel out signals



Cancellation of Signals in Wave Motion

Wave amplitudes combine by adding



Wave Interference

Two waves arrive at antenna by slightly different paths

$$\mathbf{E} = \mathbf{E}\left(R, t\right) + \mathbf{E}\left(R + \Delta R, t + \Delta t\right)$$

where $\Delta R << R$ and $\Delta t << t$

$$\mathbf{E} = \frac{\mathbf{E}_0}{R}\cos\left[2\pi\left(\frac{R}{\lambda} - ft\right)\right] + \frac{\mathbf{E}_0}{R + \Delta R}\cos\left[2\pi\left(\frac{R + \Delta R}{\lambda} - f\left(t + \Delta t\right)\right)\right]$$

$$\mathbf{E} = \frac{\mathbf{E}_0}{R}\cos\left[2\pi\left(\frac{R}{\lambda} - ft\right)\right] + \frac{\mathbf{E}_0}{R\left(1 + \frac{\Delta R}{R}\right)}\cos\left[2\pi\left(\frac{R}{\lambda} + \frac{\Delta R}{\lambda} - ft - f\Delta t\right)\right]$$

Ignoring $\frac{\Delta R}{R} << 1$

$$\mathbf{E} \approx \frac{\mathbf{E}_0}{R}\left\{\cos\left[2\pi\left(\frac{R}{\lambda} - ft\right)\right] + \cos\left[2\pi\left(\frac{R}{\lambda} - ft\right) + 2\pi\left(\frac{\Delta R}{\lambda} - f\Delta t\right)\right]\right\}$$

Wave Interference

$$\mathbf{E} \approx \frac{\mathbf{E}_0}{R}\left\{\cos\left[2\pi\left(\frac{R}{\lambda} - ft\right)\right] + \cos\left[2\pi\left(\frac{R}{\lambda} - ft\right) + 2\pi\left(\frac{\Delta R}{\lambda} - f\Delta t\right)\right]\right\}$$

Using identity $\cos(A) + \cos(B) = 2\cos\tfrac{1}{2}(A+B) \times \cos\tfrac{1}{2}(A-B)$

$$\mathbf{E} = \frac{\mathbf{E}_0}{R}\left\{2\cos\left[2\pi\left(\frac{R}{\lambda} - ft\right)\right] \times \cos\left[\pi\left(\frac{\Delta R}{\lambda} - f\Delta t\right)\right]\right\}$$

Transparent medium

$$\Delta R = c\Delta t \quad \Rightarrow \quad \frac{\Delta R}{\lambda} - f\Delta t = \Delta t\left(\frac{c}{\lambda} - f\right) = \frac{\Delta t}{\lambda}(c - \lambda f) = 0$$

$$\cos\left[\pi\left(\frac{\Delta R}{\lambda} - f\Delta t\right)\right] = 1$$

Total cancellation

$$\frac{\Delta R}{\lambda} - f\Delta t = \frac{1}{2} \quad \Rightarrow \quad \cos\left[\pi\left(\frac{\Delta R}{\lambda} - f\Delta t\right)\right] = \cos\left(\frac{\pi}{2}\right) = 0$$

Wireless Local Area Networks

IEEE 802.11 Protocol Layers

Physical Layer Convergence Sublayer

Specifies header for PHY Dependent Sublayer

- Direct Sequence Spread Spectrum (DSSS)
- Frequency Hopping Spread Spectrum (FHSS)

Transmission type

Modulation scheme

Data transmission rates

MAC layer

Medium access

Addressing

Procedures

Wi-Fi

Trademark of Wi-Fi Alliance trade association



Data Link Layer	LLC	802.2	LLC frame for SEQ/ACK/Control
	Bridging		Exchange of 802.2 PDUs
	MAC	802.11	CSMA/CA, MACA, CFP
Physical Layer	Convergence		PHY-Dependent Convergence Sublayer
	PHY		FHSS, DSSS, IR, Data rates

WiFi Ad Hoc Mode

Independent Basic Service Set (IBSS)

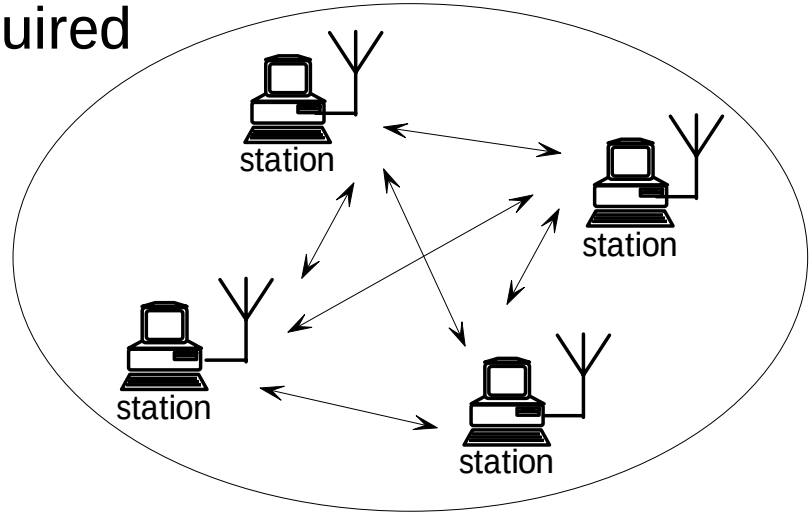
- Any set of 802.11 STAs (wireless stations)
- All STAs transmit / receive on same frequency
- Peer-to-peer service
- No connection to a wired network

Simple unmediated communication

- STAs communicate directly with one another
- Useful for quick set up
- Authentication or Registration not required

Multiple IBSSs are independent

- No bridging
- No hand-off



Independent Basic Service Set

WiFi Infrastructure Mode

Basic Service Set (BSS)

A set of wireless end stations (STA)

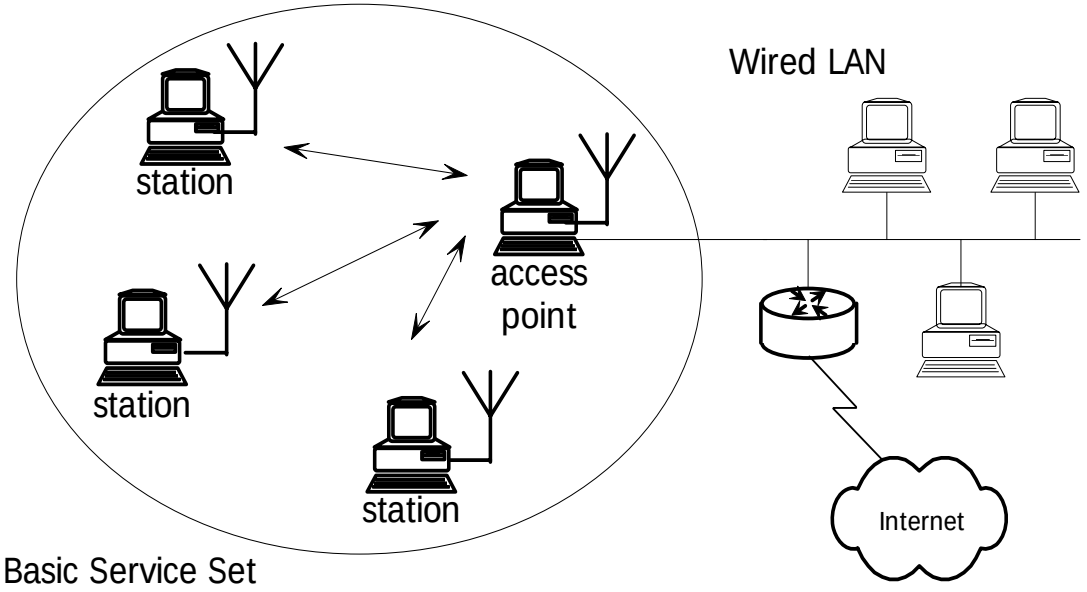
An Access Point (AP)

- Connected to the wired network infrastructure
- Acts as base station for the wireless network
- All traffic flows through AP by Contention or Polling (CFP)

Stations must Associate with AP

Authentication

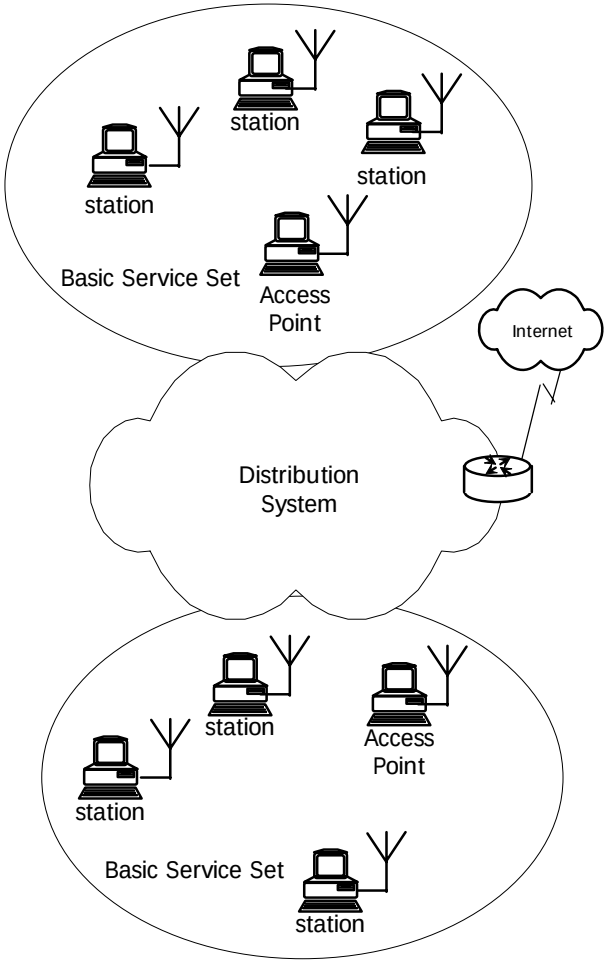
Registration



WiFi Extended Infrastructure Mode

Extended Service Set (ESS)

- Two or more BSSs
  - Form single subnetwork (broadcast domain)
  - Looks like one large BSS to LLC layer
  - One Access Point (AP) in each BSS
- BSSs connected via Distribution System (DS)
  - DS is backbone network
  - DS performs MAC-level transport of MAC SDUs
  - DS implementation not specified in 802.11



Portal

- Software gateway function in AP
- Bridges BSS to any non-802.11 DS protocol

DS services permit handoff

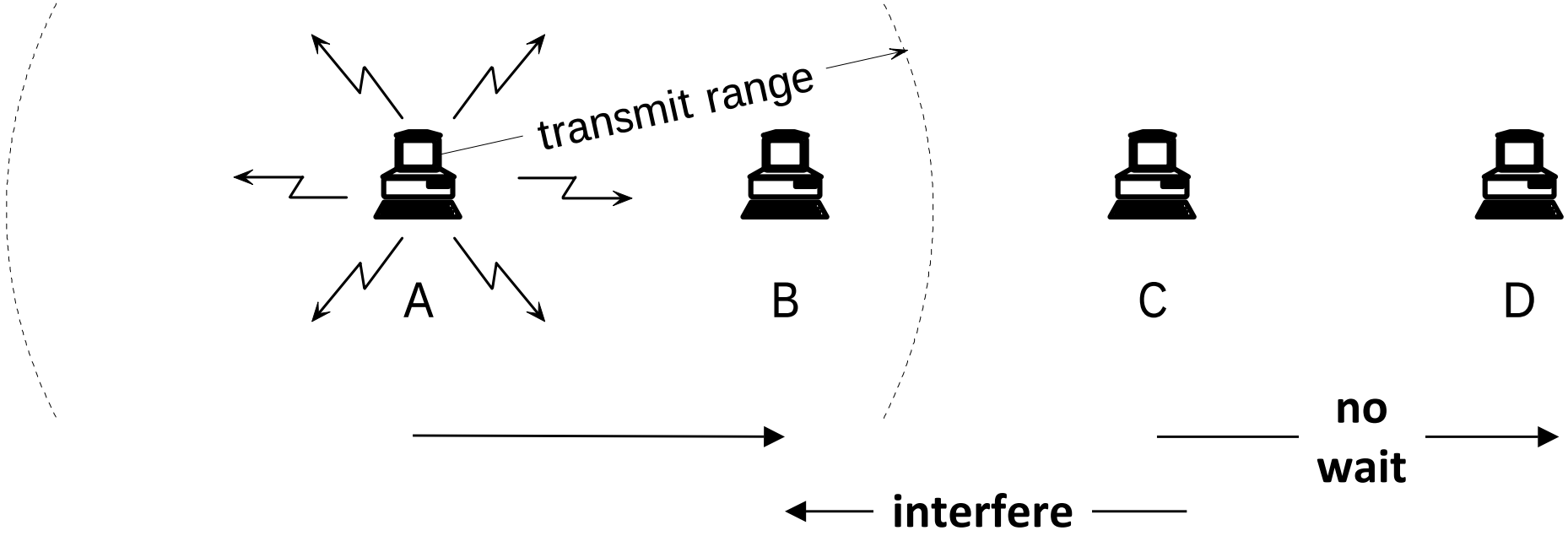
- Station moving from one BSS to another
- Requires coordination between APs

Hidden Node Problem

A transmits to B

C cannot receive from A — out of range

C transmits — corrupts transmission from A to B

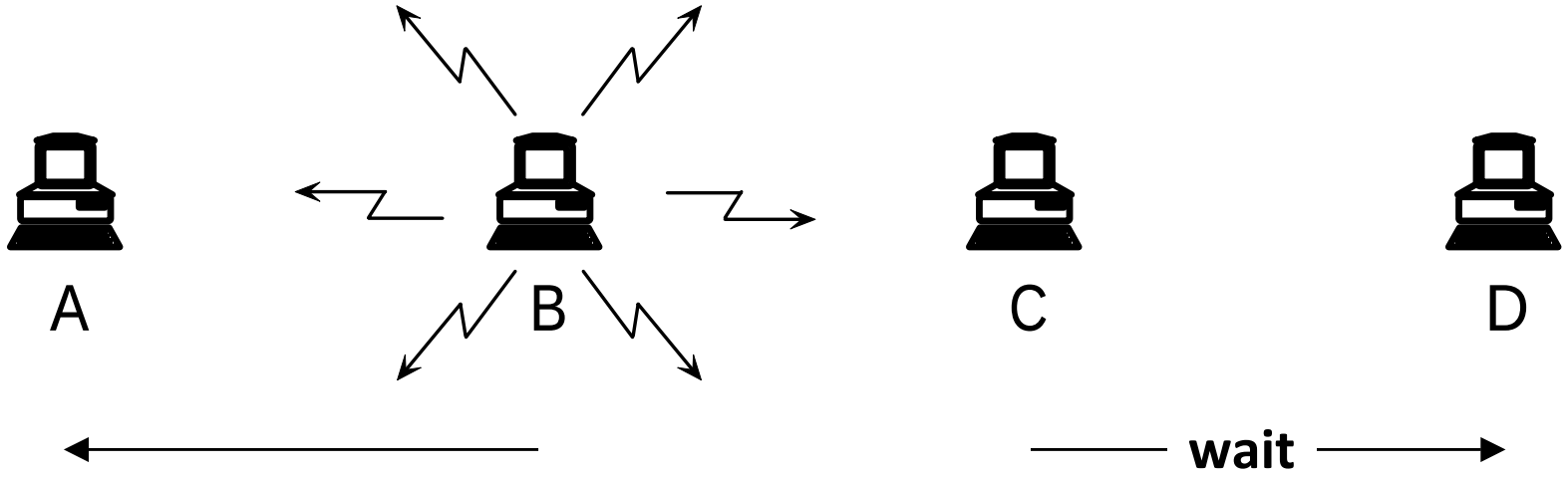


Exposed Node Problem

B transmits to A

C receives transmission from B — delays transmission to D

Inefficient — C transmission to D will not interfere with B to A



CSMA with Collision Avoidance (CSMA/CA)

Carrier Sense Multiple Access (CSMA)

Stations listen for transmissions

Do not transmit if carrier is detected

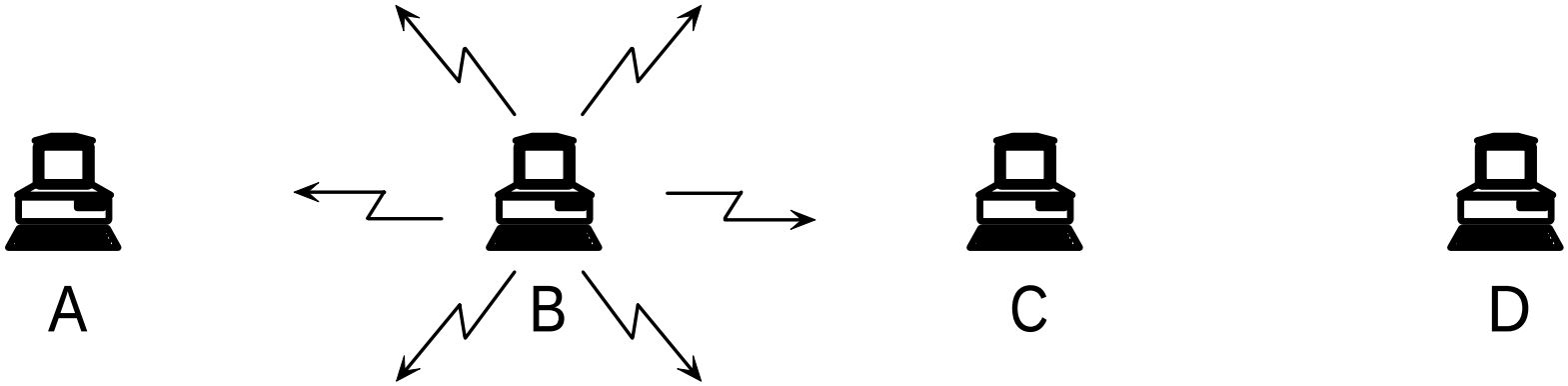
Collision detection not possible

- Hidden node problem
- Antenna cannot receive while transmitter active

Collision Avoidance (CA)

Non-persistent access

Random backoff



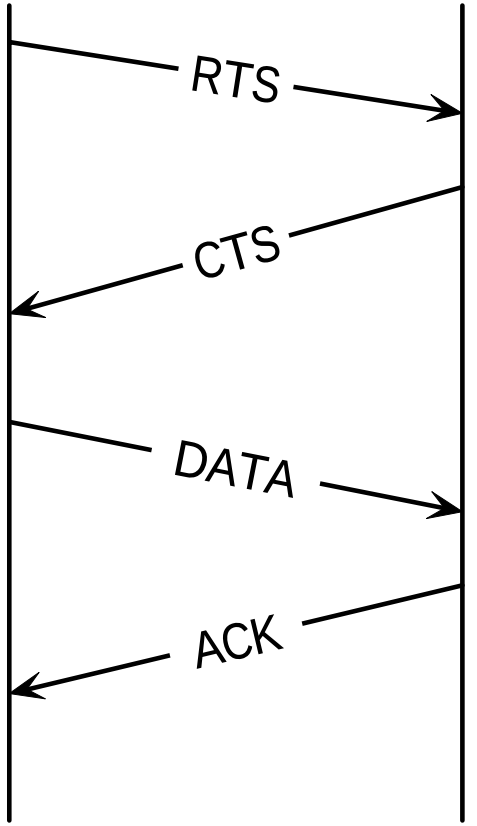
Multiple Access with Collision Avoidance (MACA)

Channel set-up before data transmission

RTS — Request To Send

CTS — Clear To Send

ACK — Acknowledge error-free transmission



Net Allocation Vector (NAV)

Transmitted in RTS

Echoed in CTS

Predicted data transmission time

Trade-off

Adds overhead

Better throughput in hidden and exposed Nodes

Multiple Access with Collision Avoidance (MACA)

C sends 30-byte RTS to D

Includes NAV for data transmission

B and D hear RTS

D responds with CTS to C

Echoes NAV

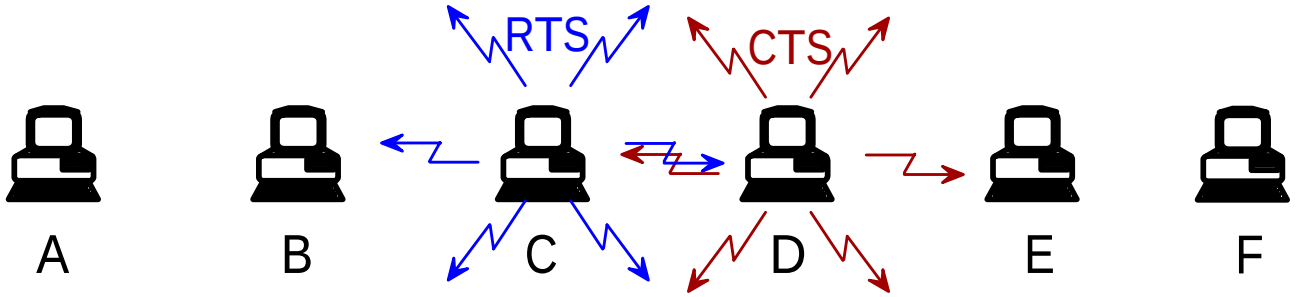
C and E hear CTS

B hears RTS but not CTS

B can transmit to A — no interference with C to D

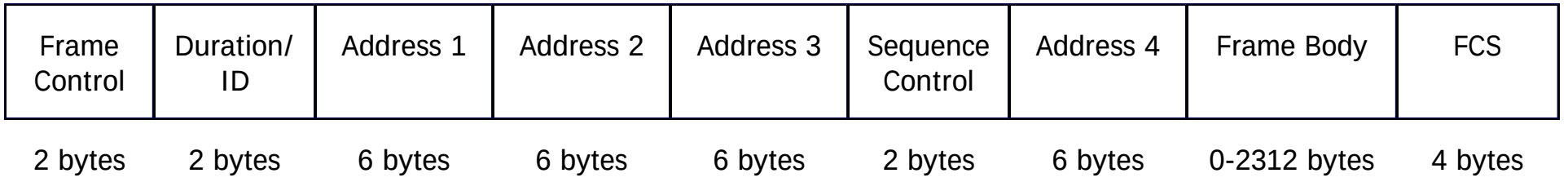
E hears CTS but not RTS

E waits data transmit time before transmission to F



MAC Sublayer Frame Structure

Frame Control	Control flags
Duration/ID	Timing control
Addresses	Various MAC entities
Sequence Control	Sequence/Fragment number for error/flow control
Frame Body	0 or more data bytes (SDU)



Frame Control

Type and Subtype	Data, Control, Management with subtypes	
To DS/From DS	Access Point (AP) is destination/source	
More Fragments	Part of fragmented LLC packet	
Retry	Indicates re-transmission of bad packet	
Power Management	STA alerts AP of its mode	
	Value of 1	STA will be in power-save mode
	Value of 0	STA will be in active mode
More Data	AP alerts STA (in power-save mode) of buffered frames	
WEP	Indicates WEP encrypted data	
Order	Indicates Strictly Ordered service class	

Protocol Version	Type	Subtype	To DS	From DS	More Fragments	Retry	Power Management	More data	WEP	Order
2 bits	2 bits	4 bits	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit

MAC Layer Address Fields

4 Address Fields

5 possible MAC entities:

BSS Identification Number (BSSID)

Source Address (SA)

Station that initiated the message

Destination Address (DA)

Final destination for the message

Transmitting Station Address (TA)

Source station for the message on this hop

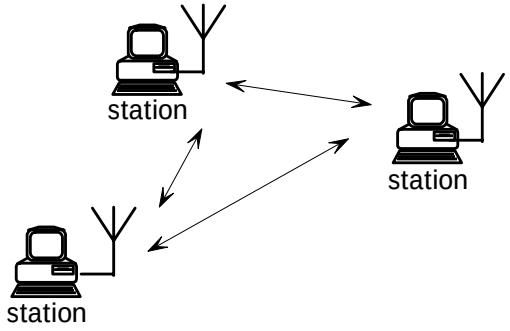
Receiving Station Address (RA)

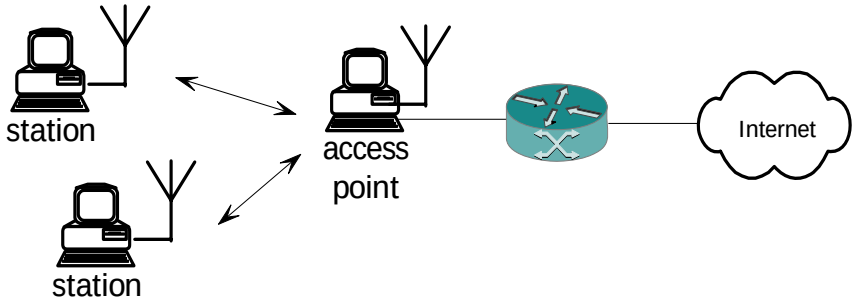
Destination station for the message on this hop

Address Field Definitions

To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	DA	SA	BSSID	—
0	1	DA	BSSID	SA	—
1	0	BSSID	SA	DA	—
1	1	RA	TA	DA	SA

Address 1	Immediate destination address
Address 2	Immediate source address
Address 3	Final destination or source when DS performs distribution
Address 4	Source address for DS to DS messages (802.11 is also DS)





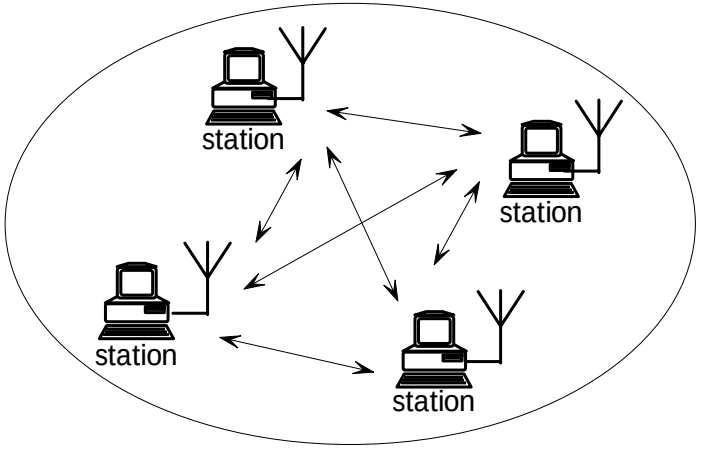
Addressing in an IBSS

To DS	From DS	Address 1	Address 2	Address 3
0	0	DA	SA	BSSID

Independent Basic Service Set (IBSS)

No Access Point (AP) and no DS

Fields To DS and From DS are 0



Independent Basic Service Set

Address 1	Immediate destination address (DA)
Address 2	Immediate source address (SA)
Address 3	BSSID Identifies Ad Hoc network Prevents message from reaching outside IBSS

Data Addressing in a BSS

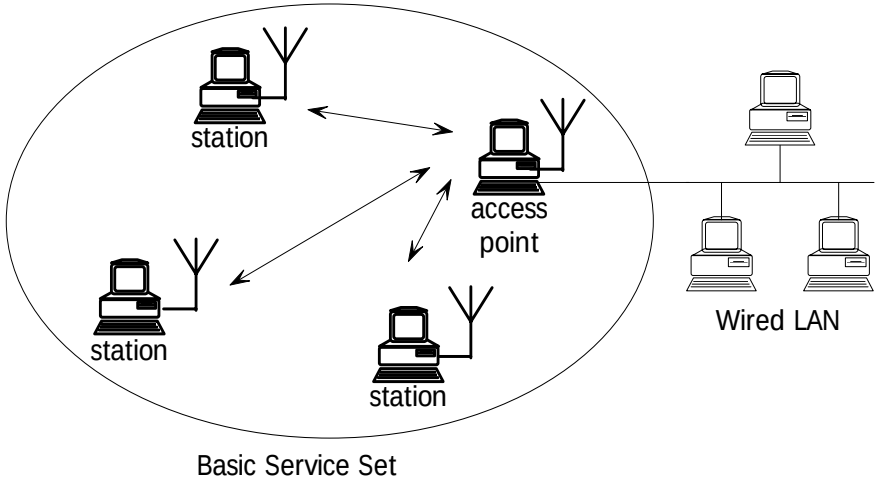
To DS	From DS	Address 1	Address 2	Address 3
0	1	DA	BSSID	SA
1	0	BSSID	SA	DA

Basic Service Set (BSS)

All transmissions are sent To/From Access Point

To/From DS actually means To/From AP

Address 1	Immediate destination address (DA)
Address 2	Immediate source address (SA)
Address 3	Final Destination or Source

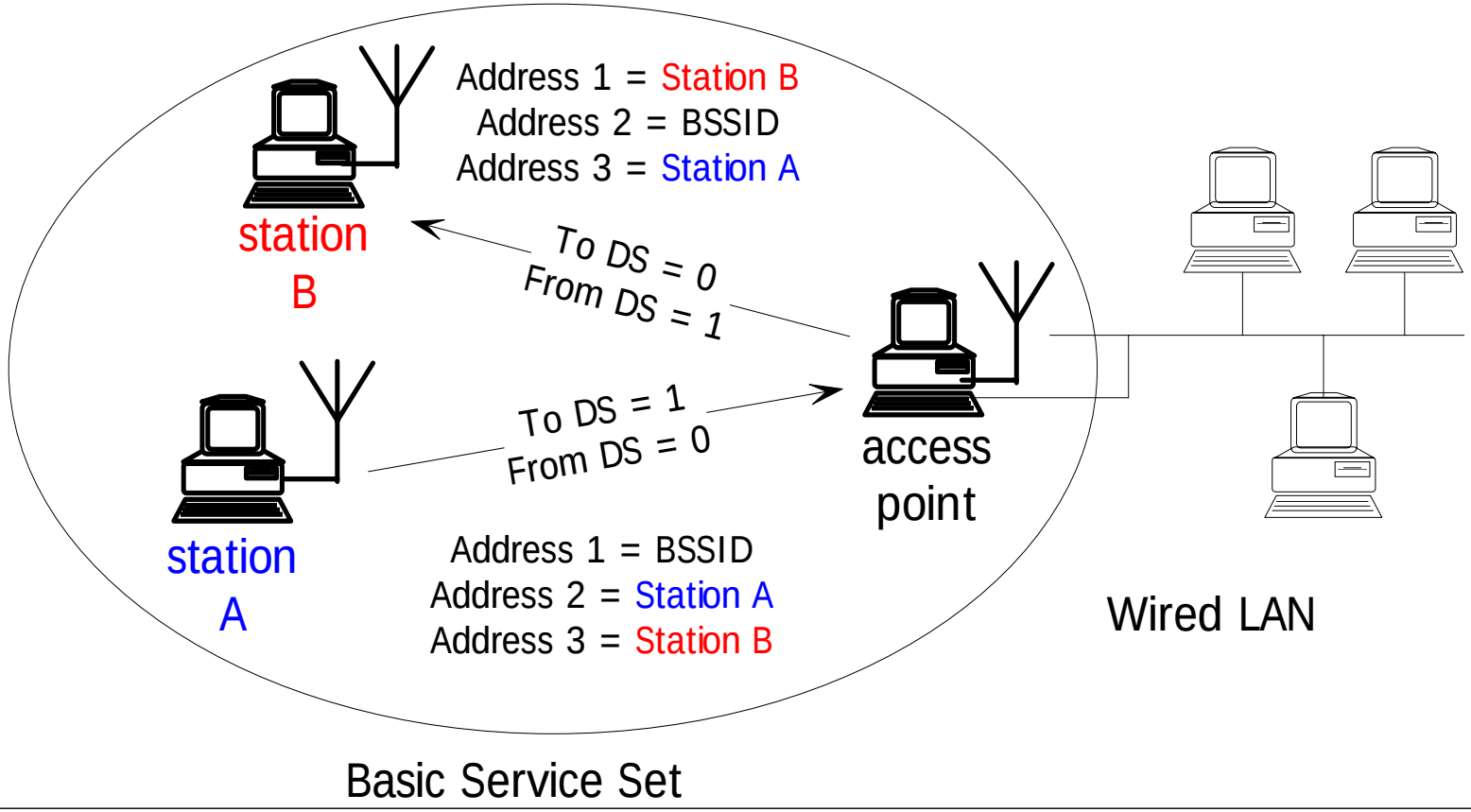


Basic Service Set

BSS Addressing Example

To DS	From DS	Address 1	Address 2	Address 3
0	1	DA	BSSID	SA
1	0	BSSID	SA	DA

Station A sends message to Station B via AP (BSSID)



Control and Management Addressing in a BSS

To DS	From DS	Address 1	Address 2	Address 3
0	0	DA	SA	BSSID

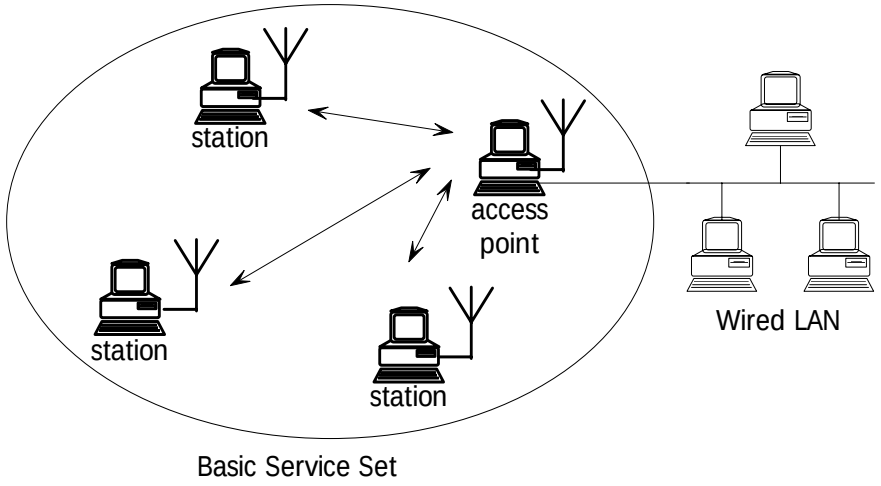
Control and Management messages in a BSS:

Only involve stations in the BSS and the AP

Are sent with To DS = From DS = 0

Either the Source or the Destination will be the AP (BSSID)

Address 3 in included as an error check



Addressing in an ESS

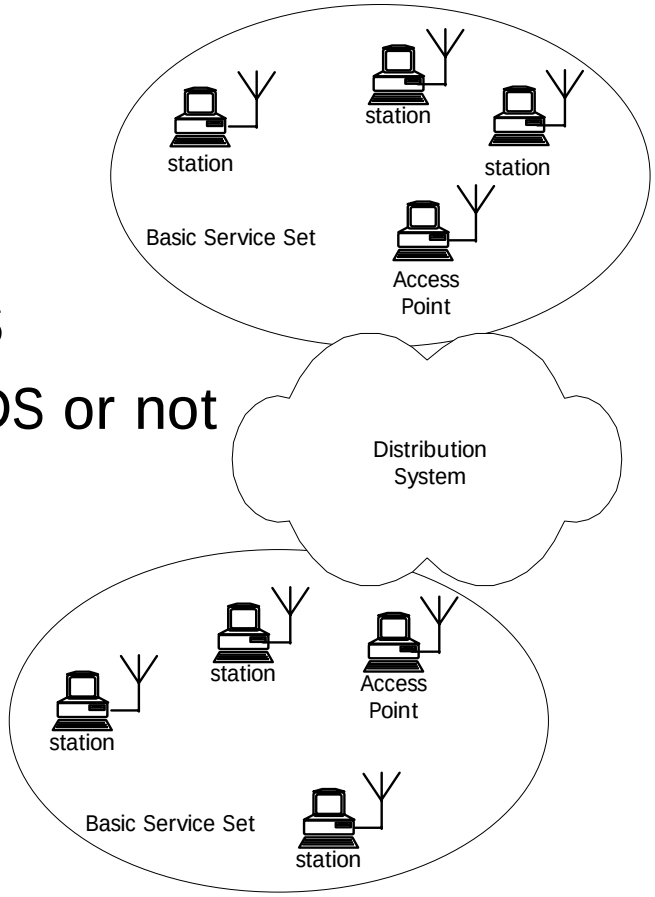
To DS	From DS	Address 1	Address 2	Address 3
0	1	DA	BSSID	SA
1	0	BSSID	SA	DA

Extended Service Set (ESS)

All transmissions are sent via an AP

To the stations, entire ESS looks like one BSS

Stations do not know if message passes via DS or not



Address 1	Immediate destination address (DA)
Address 2	Immediate source address (SA)
Address 3	Final Destination or Source

ESS Addressing Example

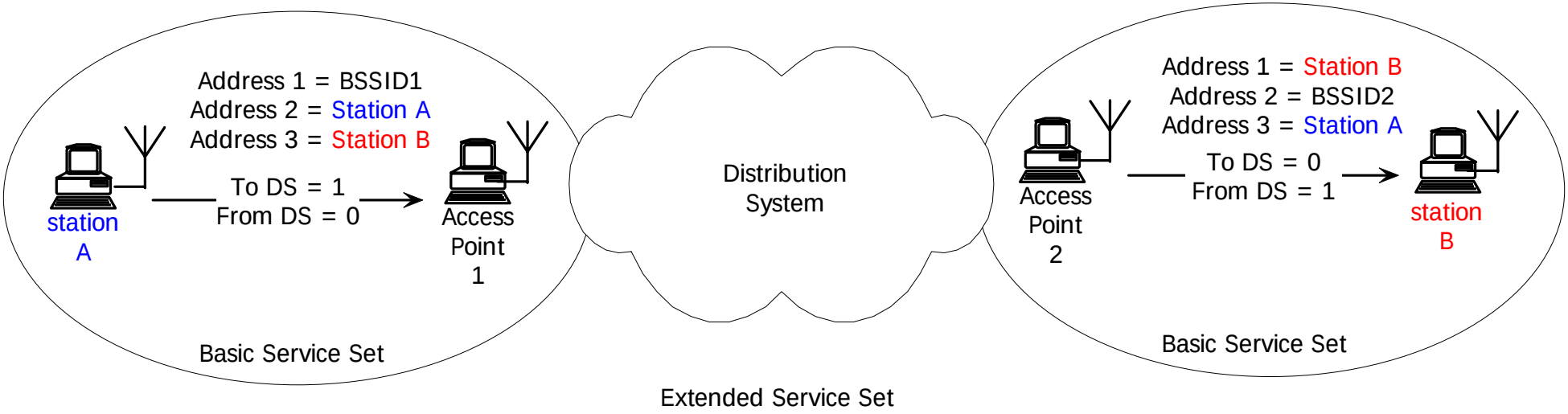
To DS	From DS	Address 1	Address 2	Address 3
0	1	DA	BSSID	SA
1	0	BSSID	SA	DA

Station A sends message to Station B via

AP1 (BSSID1) → DS → AP2 (BSSID2)

DS must forward Data, Sequence, SA, and DA

By some legal means



Mobility Management in Wireless Systems

Registration

Performed when mobile station (MS) activated in Service Area

Authentication

WiFi — managed by access point (AP)

2G / 3G Cellular

- Home Location Register (HLR)
  - Maintains account + location information for home customers
- Visitor Location Register (VLR)
  - Cache of HLR data on active roamers in each Service Area

4G / 5G — integrated mobility management replaces HLR/VLR

Call Establishment

Performed when user initiates or receives call

Security

Protects from fraud and eavesdropping

Handoff (handover)

Performed when MS changes attachment point to network

Handoff (Handover)

User moves between cells

Hard Handoff

- Old cell transfers control to new cell
- Break-Before-Make sequence
  - Transceiver in old cell stops transmitting to user
  - Transceiver in new cell begins transmitting to user
- New BS assigns user frequency pair from its frequency group

Soft Handoff

- Central transceiver coordinates with nearest cells
- Determines which transmitter is receiving strongest signal from user
- Make-Before-Break sequence
  - Transceiver in old cell transmitting to user
  - Transceiver in new cell begins transmitting to user
  - Transceiver in old cell stops transmitting to user

Cellular Networks

First Mobile Telephone

AT&T Mobile Phone System (MPS)

1970

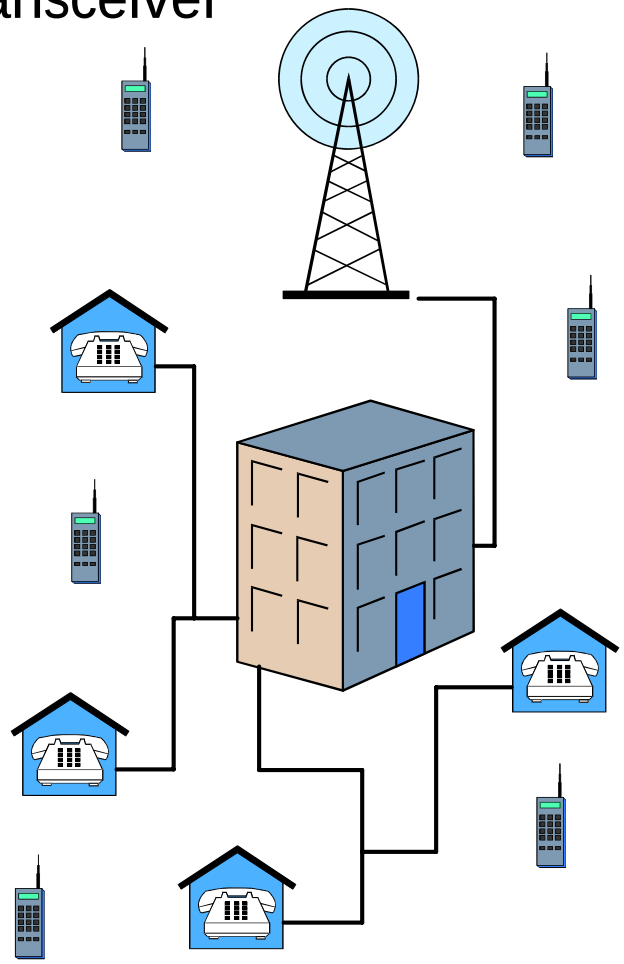
0G

One central transceiver (transmitter/receiver)

Mobile telephones communicate via central transceiver

Transmit at high power for maximum distance

System covers 65 to 80 km



Modulation is standard analog FM

Supports 12 simultaneous mobile phone calls

If 12 channels busy, other calls are blocked

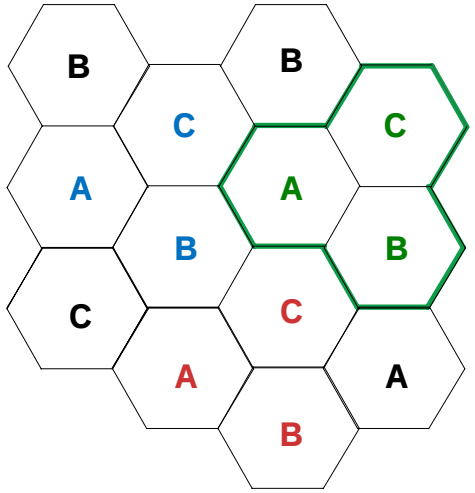
Requires 24 carrier frequencies

2 frequencies per phone:

- Dedicated transmit frequency
- Dedicated receive frequency

Cellular Concept

Divide coverage area into cells



In each cell

Central cell transceiver serves all clients in cell

Mobile Stations communicate via cell transceiver

Each active device allocated frequency pair

- Receive — downstream from base station
- Transmit — upstream to base station

Transmit at low power (just enough to cover a cell)

Use same frequencies in many cells

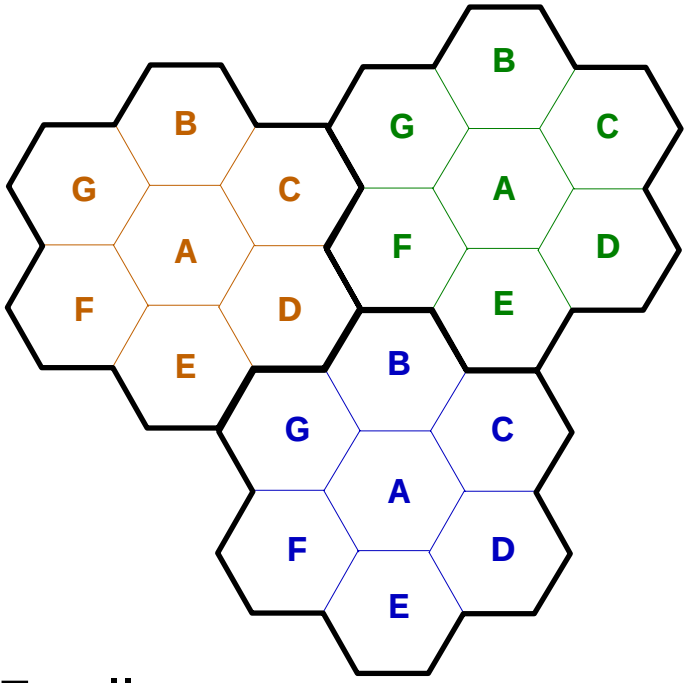
No interference between cells

Handoff

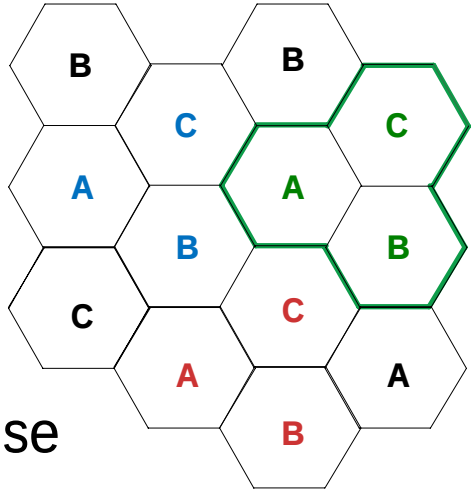
Telephone can move from cell to cell during a call

Requires cell-to-cell infrastructure and coordination

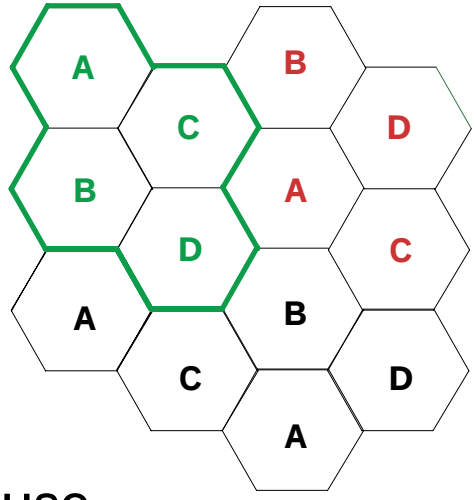
Frequency (Channel) Reuse Patterns



7 cell reuse



3 cell reuse



4 cell reuse

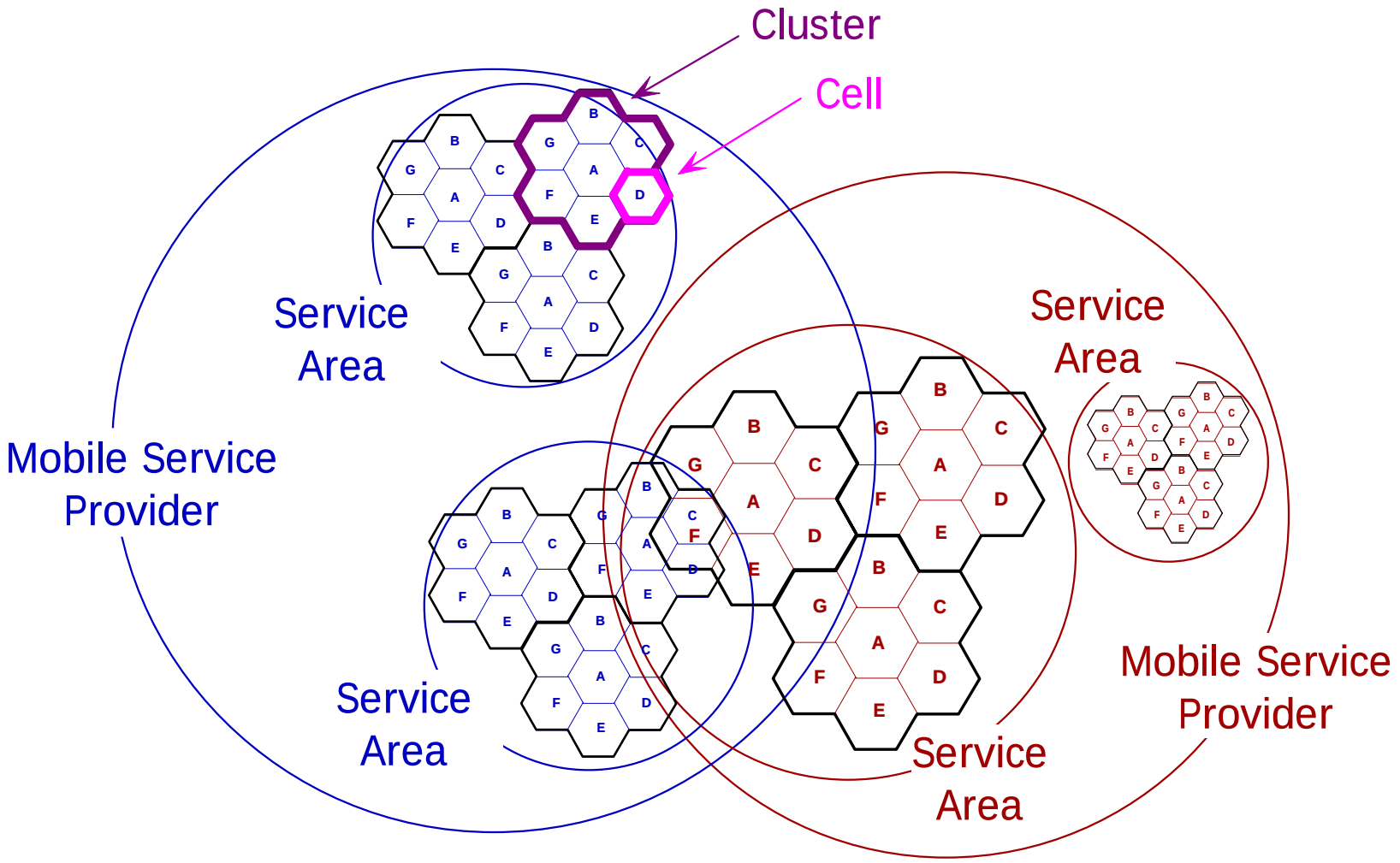
Mobile Network Switching Hierarchy

Mobile Service Provider

Service Areas or Registration Areas

Clusters

Cells

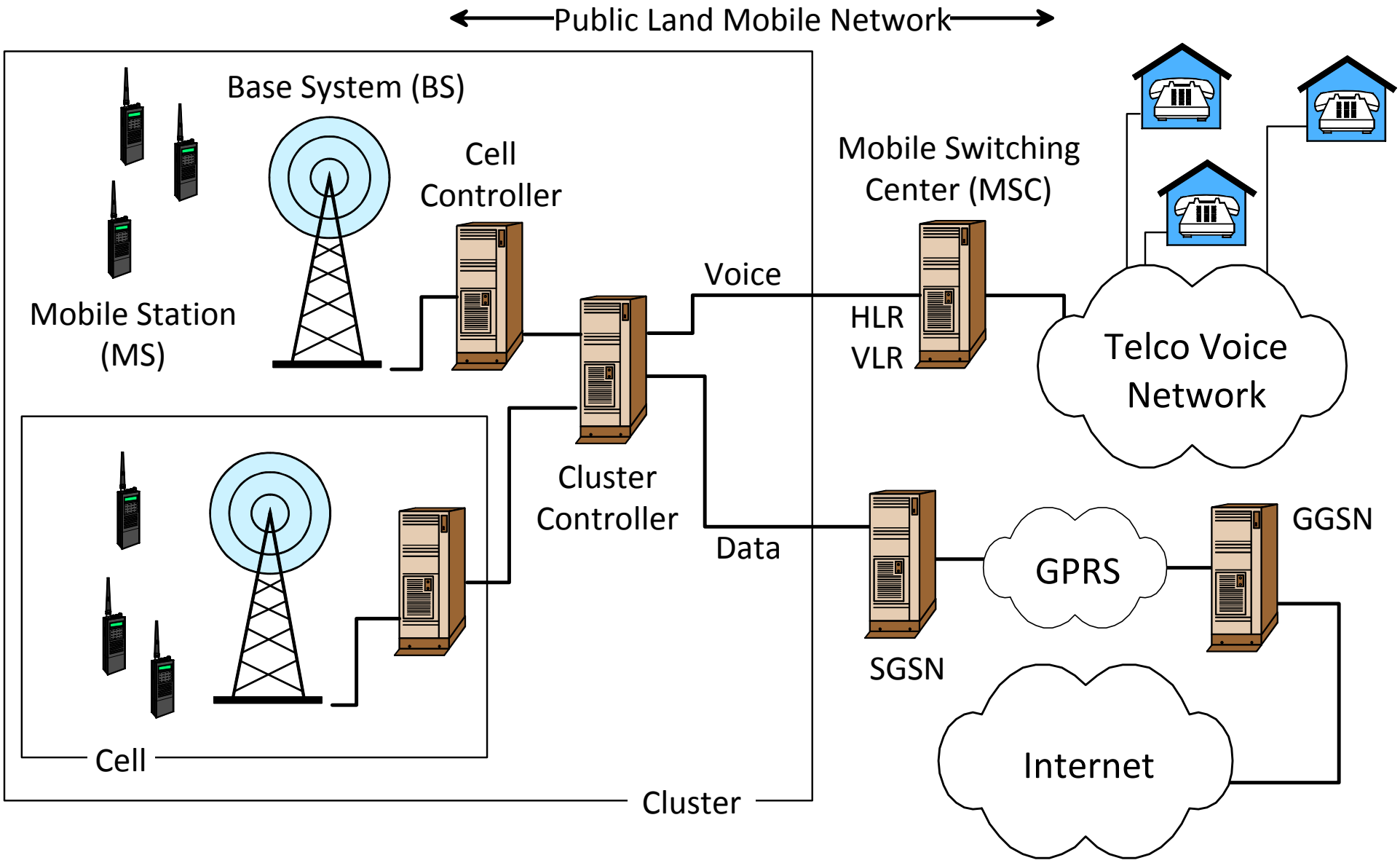


Generations in Cellular Networks

1G	Analog radio end-to-end Separate proprietary systems with no interoperability
2G	US — Separate proprietary systems with no interoperability Analog and digital radio Very limited data service EU — Public standards for European-wide interoperability GSM (Global System for Mobile telephones) digital voice channel CDMA (Code Division Multiples Access) digital voice channel GPRS (General Packet Radio Service) IP-based data network
3G	UMTS (Universal Mobile Telecommunications System) based on GSM WCDMA (Wideband CDMA) based on CDMA Adopted as international standards by ITU Higher data rates Improved mobility management
4G	ITU standard for enhanced performance (mostly still not satisfied) Eliminates voice network — voice calls carried as Voice over IP (VoIP)
5G	ITU standard for enhanced performance More complete integration of network functions

Cellular Network in 2G

Wireless to base station — uses Telco network for voice service



Registration Process in GSM

Mobile subscriber (MS) enters Service Area

Establishes low bit-rate control channel with service provider

MS requests service

Base Transmitter System (BTS) allocates a frequency pair

BTS requests access from Base Controller System (BSC)

BSC reports to Mobile Switching Center (MSC)

Location, Status, and Identity

- Dedicated hardware ID code in phone
- Subscriber Identity Module (SIM) card identifies customer in GSM
- Mobile Station generates access code to network

Transmits code by public key encryption (PKE) algorithm

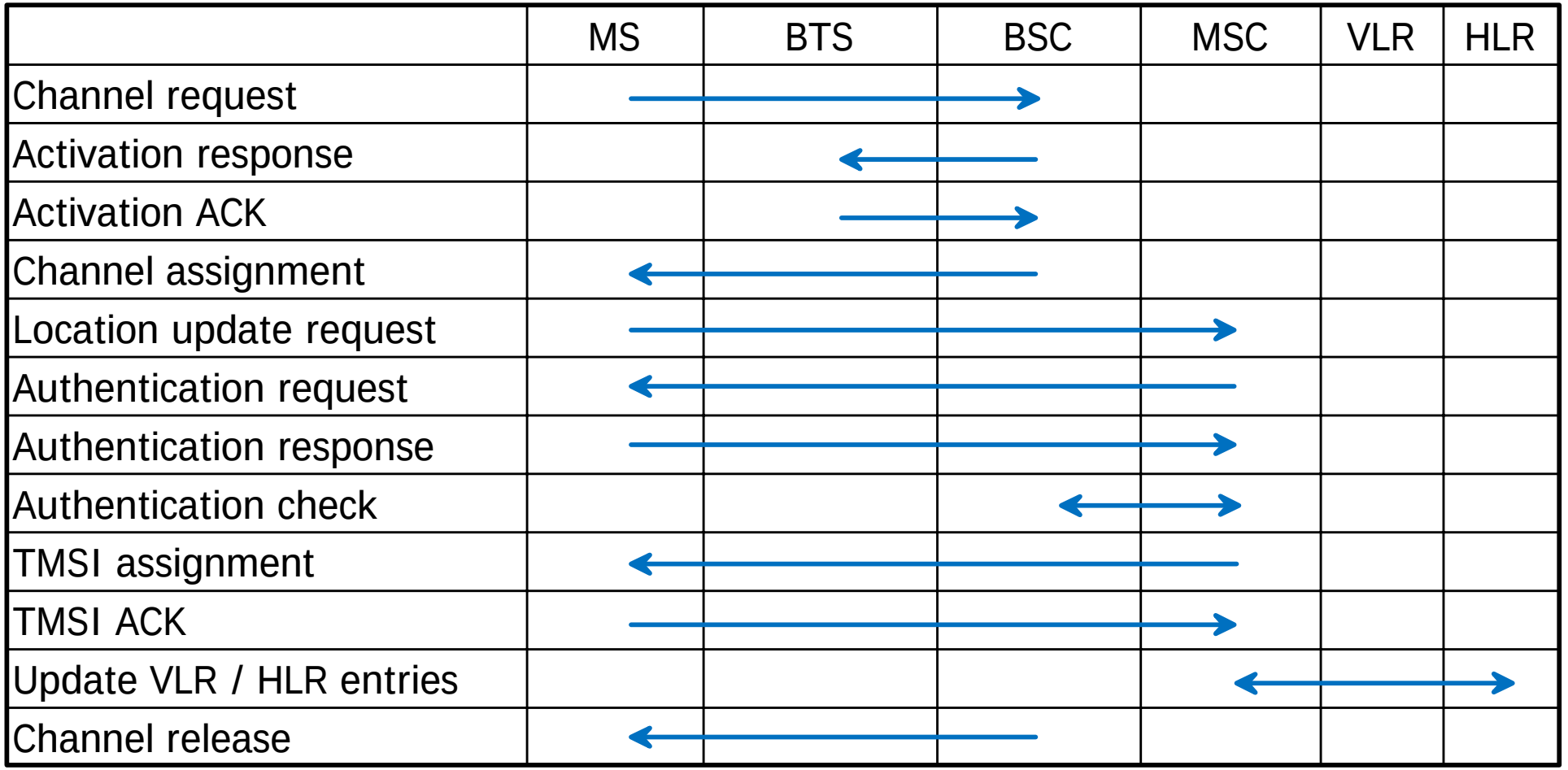
Mobile Switching Center (MSC)

Authenticates customer identity with HLR

For roaming subscriber, creates VLR entry

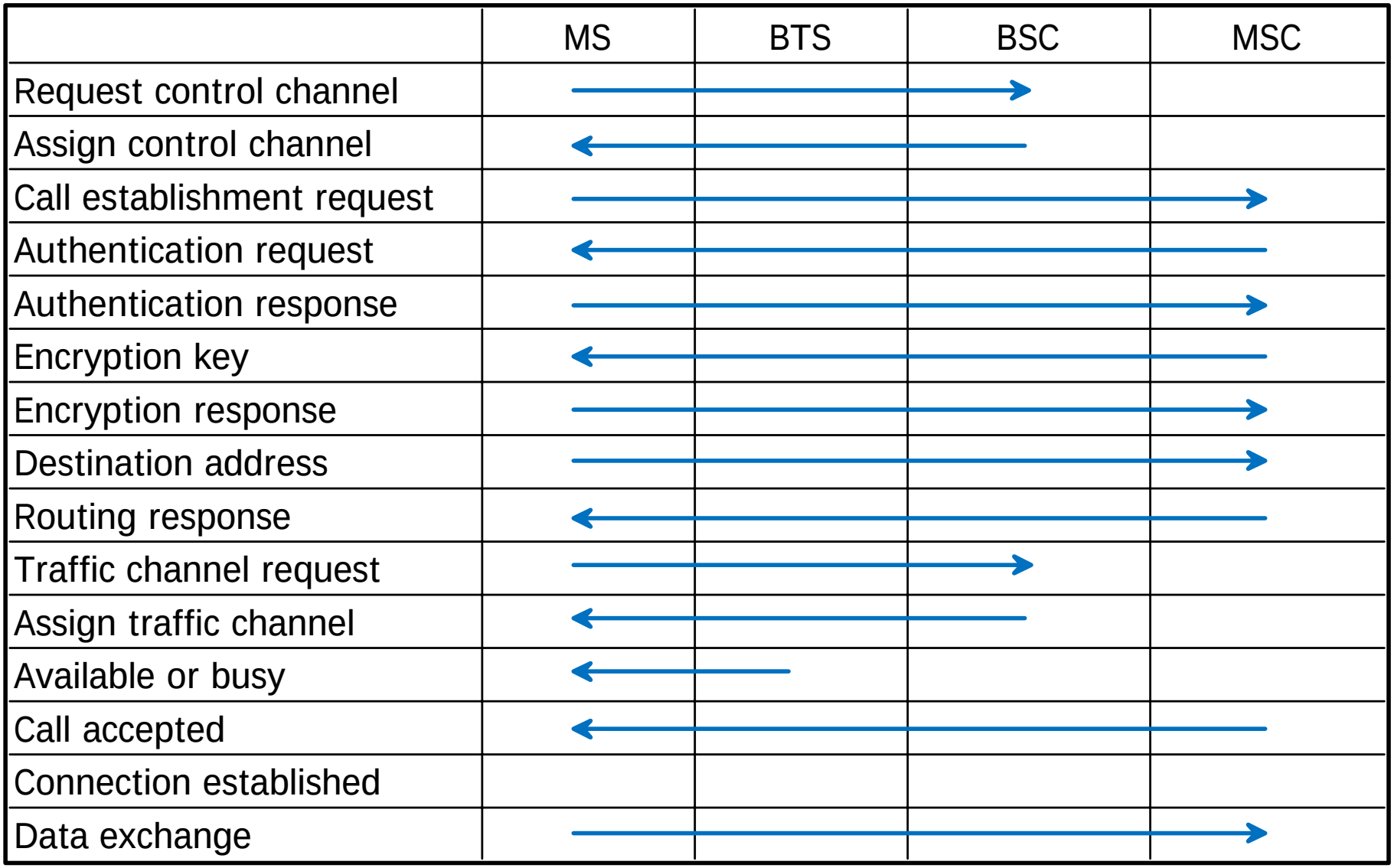
Updates Home Location Register (HLR) and billing database

GSM Registration



GSM Call Establishment

MS Initiated

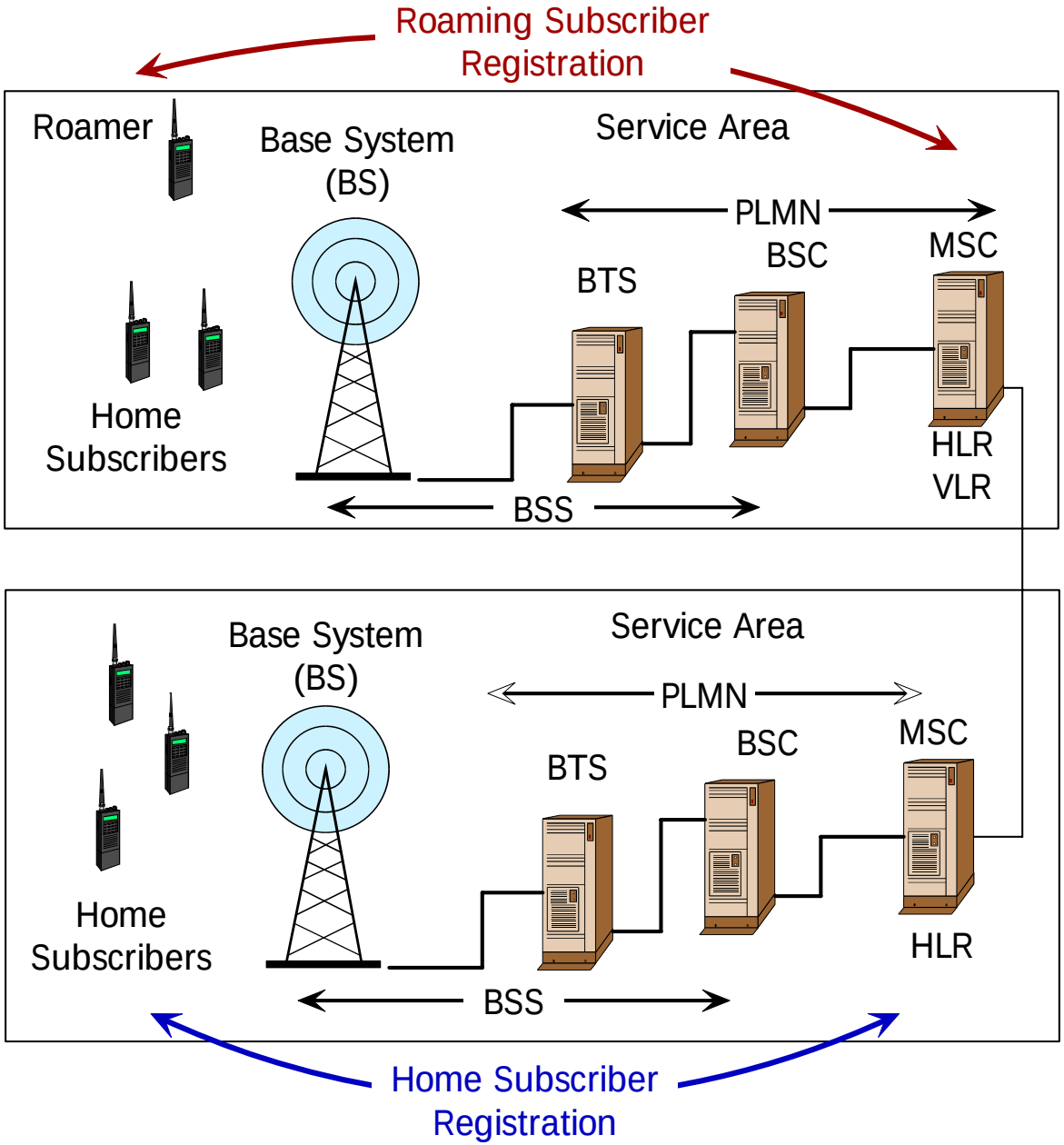


GSM Call Establishment

Mobile Terminated

	MS	BTS	BSC	MSC	VLR	HLR	GMSC	PSTN	Caller
Standard call set-up								←	
Request to Gateway MSC							←		
HLR user request						↔			
Assign roaming number					↔				
Request to MSC (user location)						←			
Update user status				↔					
Page MS	←								
Authentication				↔					
Call connection	←								

Mobility Management in GSM

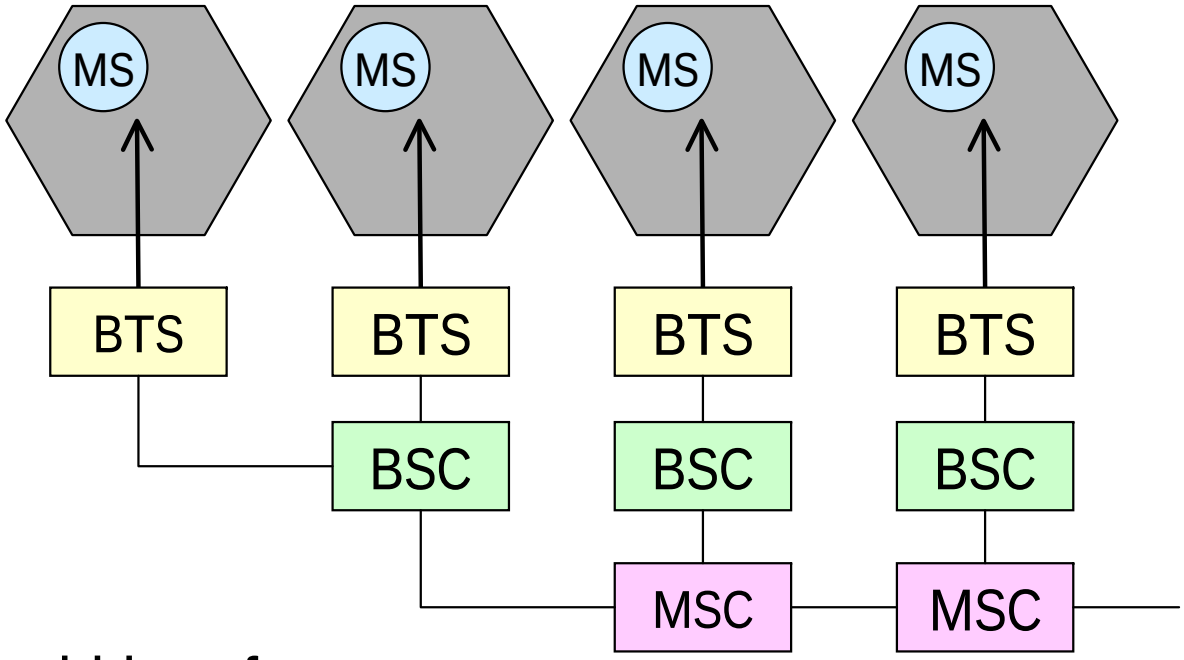


PLMN — public land mobile network, the portion of the cellular network that operates over cables.

BSS — Base Station Subsystem
GSM Access Network
Handles one cell

MSC — Mobile Switching Center
Handles multiple BSSs

Handover Types in GSM



Intra-cell

Change frequencies to avoid interference

Inter-cell — Intra-BSC

MS moves between cells within control of one BSC

Inter-BSC — Intra-MSC

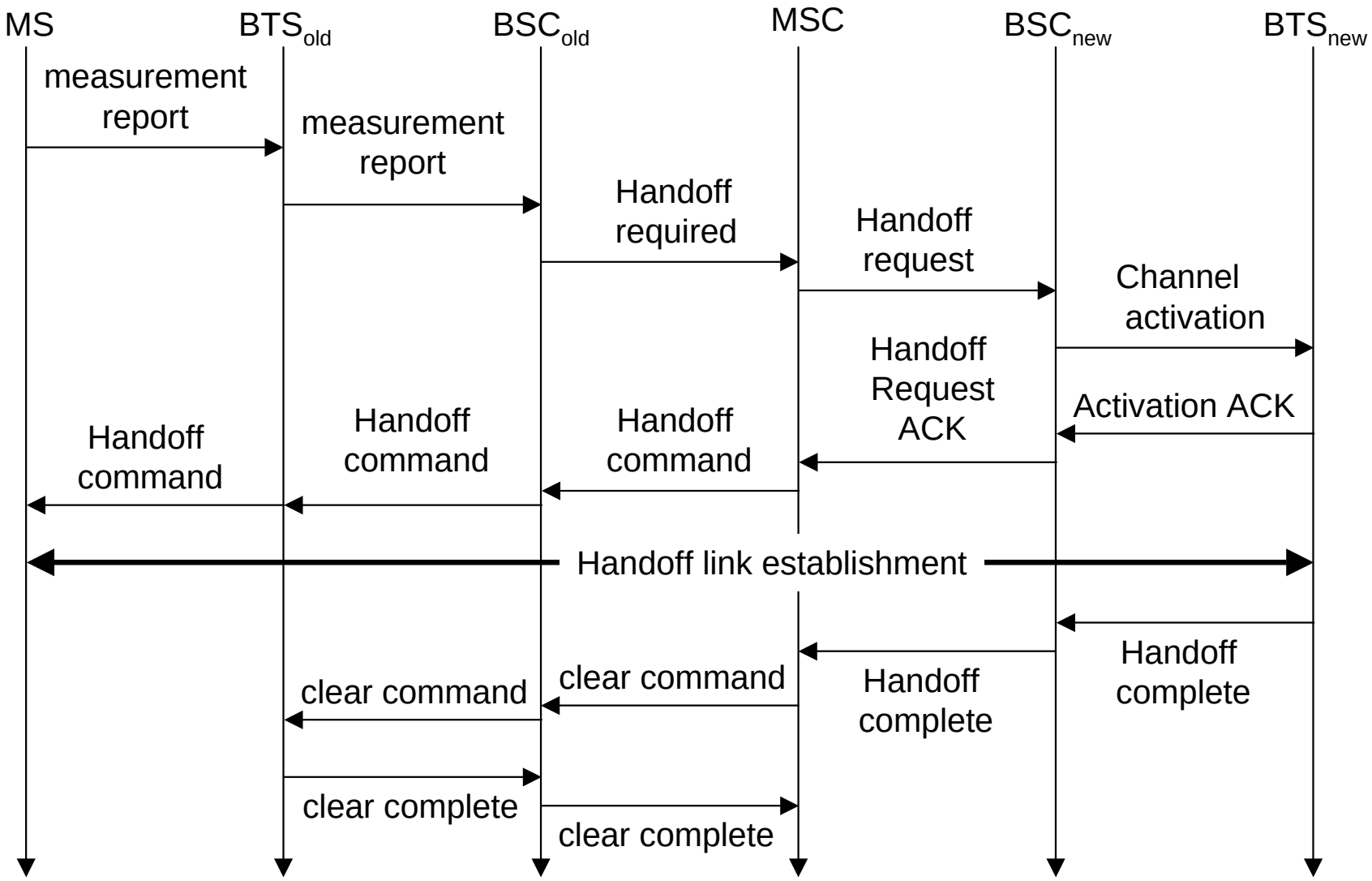
MS moves between cells controlled by different BSCs

MSC controls handover

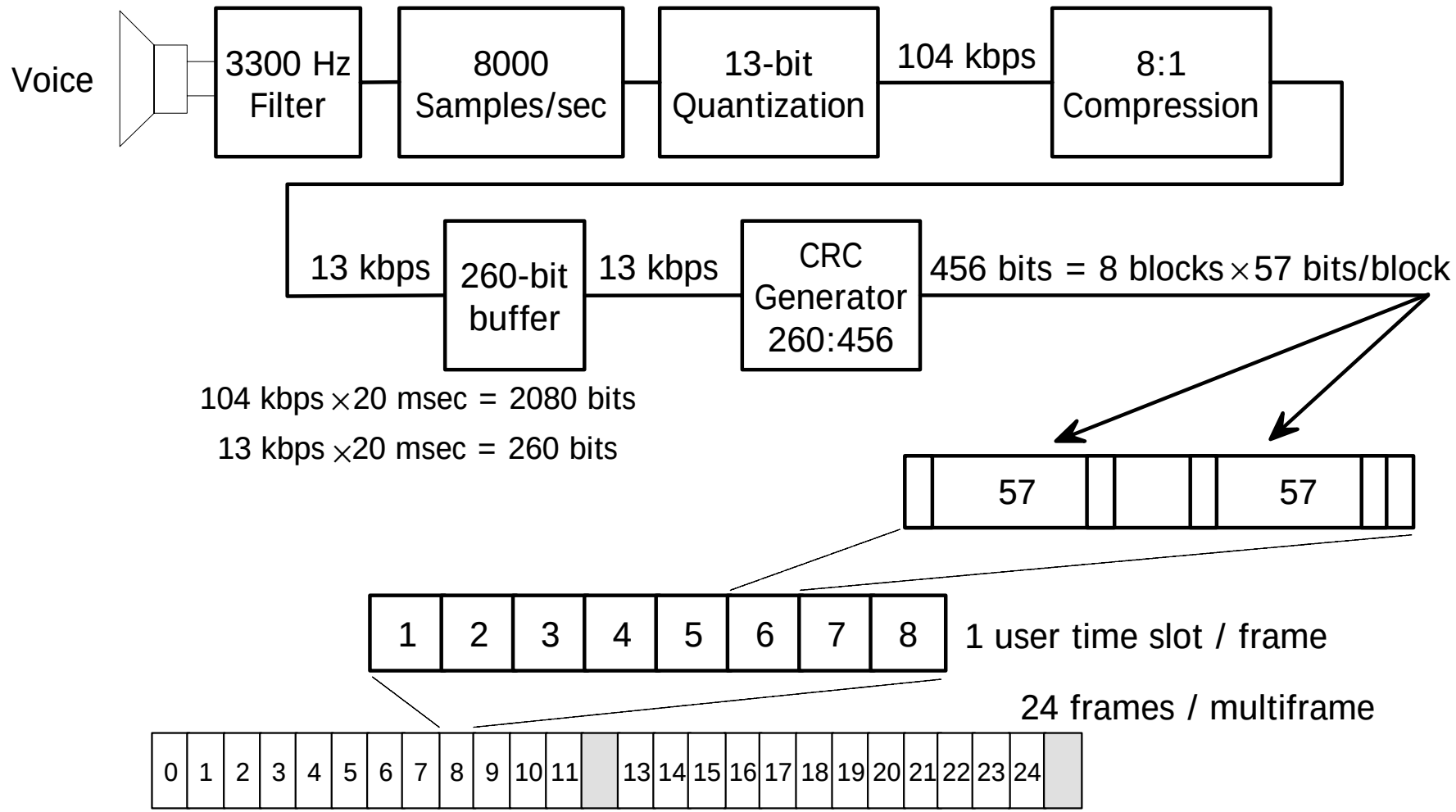
Inter MSC

MS moves between cells controlled by different MSCs

GSM Inter-BSC Handover Procedure



GSM Voice Transmission Summary

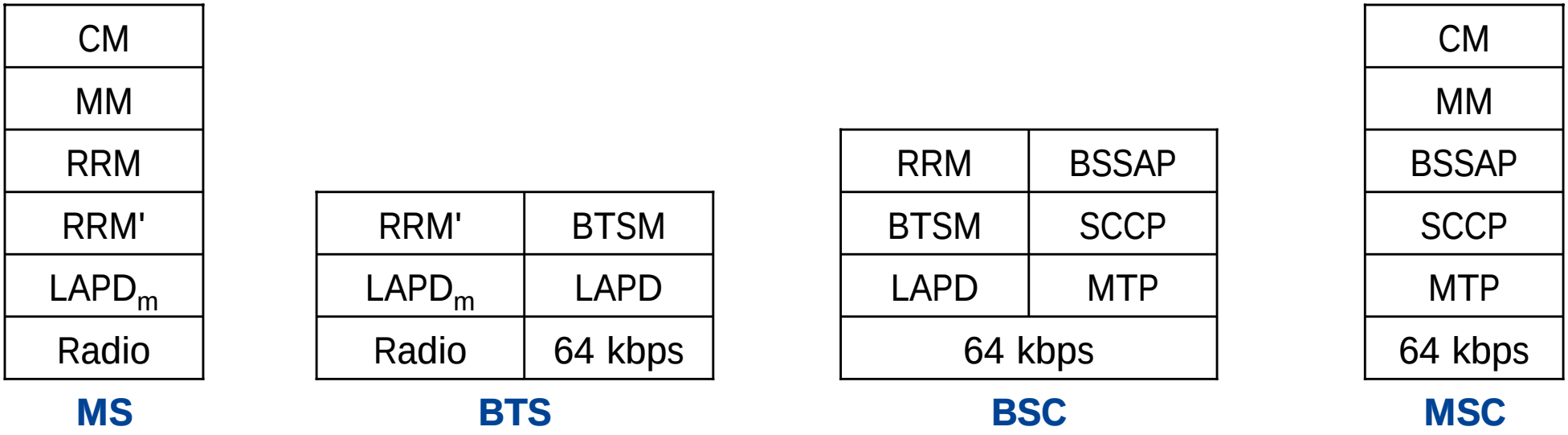


57 user bits per field × 2 fields per frame × 24 frames per multiframe = 2736 user bits per multiframe

2736 bits per multiframe / 120 ms per multiframe = 22.8 kbps

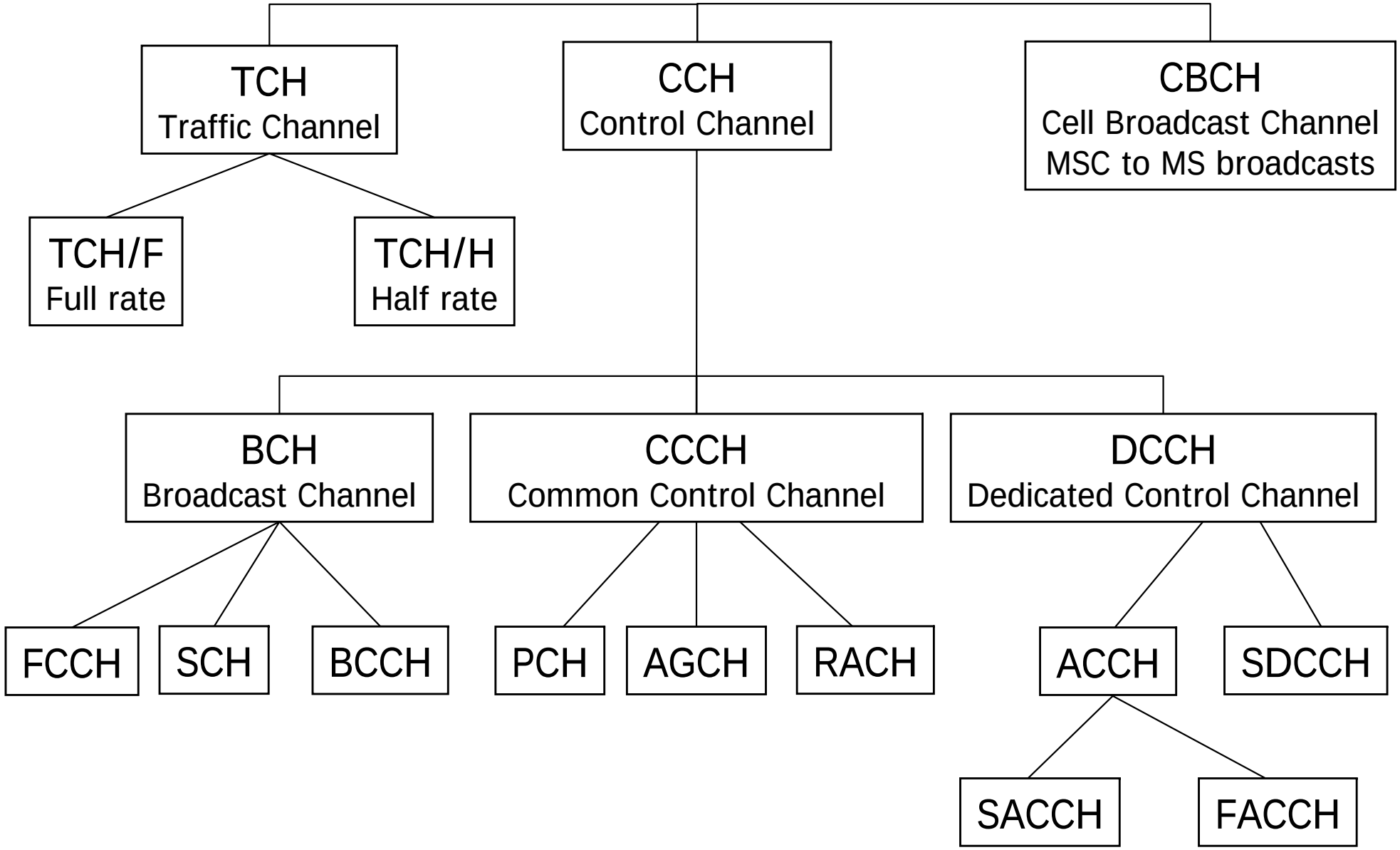
22.8 kbps / (456/260) = 13 kbps

GSM Protocol Stack



CM	Connection Management
MM	Mobility Management
RRM RRM'	Radio Resource Management — allocates physical parameters for radio system Separate protocol instances at MS/BTS layer and MS/BSC layer
LAPD	Link Access Protocol D — ISDN layer 2 protocol (Q.920/921) for LLC services
BTSM	Base Transceiver Station Management (BTS to BSC management messages)
BSSAP	BSS Application Part (call setup + management)
SCCP	Signaling Connection Control Part (SCCP) — one SCCP connection per MS
MTP	Message Transfer Part — standard PSTN signaling and management

GSM Logical Channel Structure



General Packet Radio Service (GPRS)

Provides packet mode data access for GSM

- IP-based architecture
- Began as 2.5G enhancement

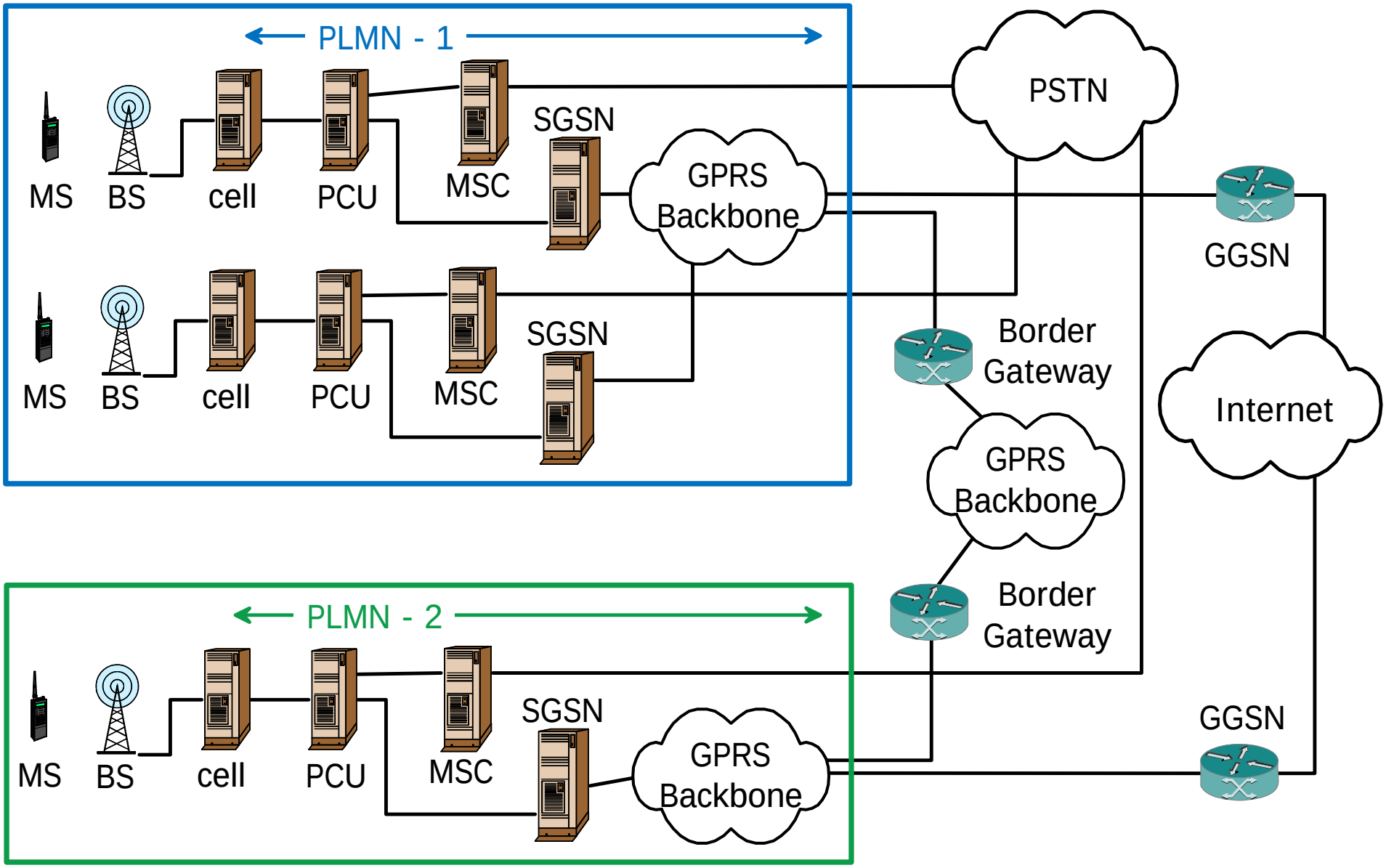
IP datagrams separated from circuit mode traffic at cluster

- Packet Control Unit (PCU)
  - Packet mode function in BSC to handle IP datagrams
- Circuit mode voice/data routed to MSC
  - Forwarded to other MSC or PSTN
- Packet mode data is routed to Serving GPRS Support Node (SGSN)
  - Forwarded to Internet
  - PCU to SGSN runs IP over Frame Relay

Mobility management

- Circuit mode traffic uses PSTN / PLMN routing
- Packet mode traffic uses IP routing

GPRS System Architecture



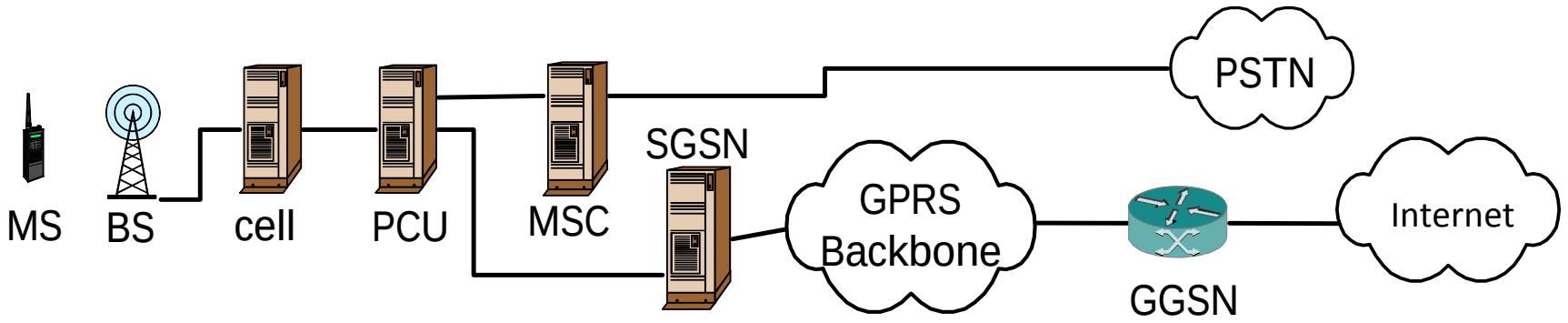
GPRS Support Nodes

Serving GPRS Support Node (SGSN)

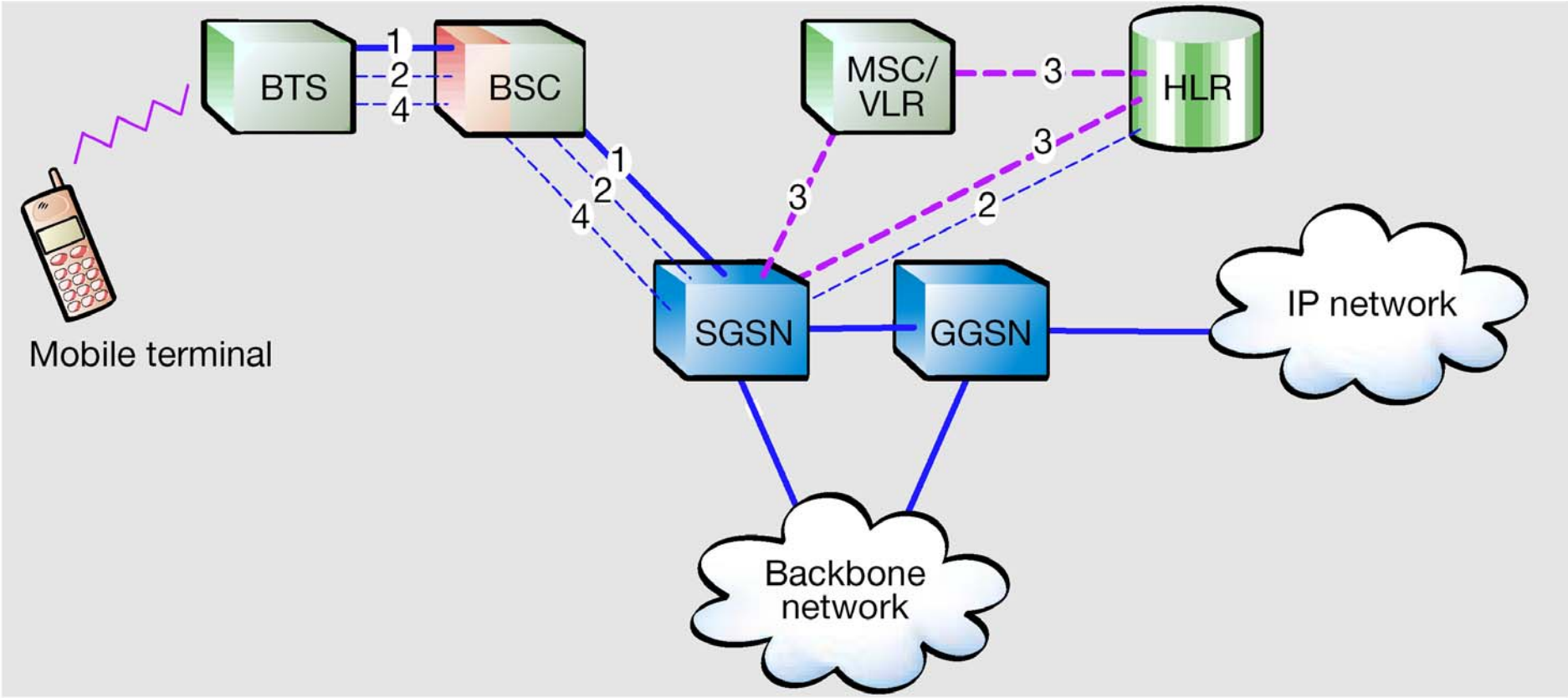
- Packet-switched version of MSC
- Handles packets to / from Mobile Stations (MS)
- Handles MS mobility management

Gateway GPRS Support Node (GGSN)

- Interfaces SGSNs to external IP networks
- Maintains routing information
  - Exterior gateway for GPRS network
  - DHCP — assigns IP addresses to MS
  - Routes incoming IP datagrams to appropriate PCU



GPRS Connection Process



GPRS Connection Process

MS switches on and sends GPRS attach request

User Registration — 1

- Associate PLMN address with Packet Data Protocol (PDP) address
  - PDP address — Static or dynamic IP
  - PLMN address — International Mobile Subscriber ID (IMSI)

Authentication

- BSC queries Home Location Registers (HLR) — 2
- HLR updates Visitor Location Registers (VLR) — 3

Call Admission Control (CAC) — 4

- Determines required network resources
- Grants resources if available

Routing — hop-by-hop IP datagram delivery

- Routing tables in GSN (GGSN or SGSN)
  - Address conversion / VC mapping
- GSN handles compression and encryption

High Speed Circuit Switched Data (HSCSD)

Circuit Switched Data (CSD)

- 14.4 kbps circuit mode data connection in 2G GSM
- User data replaces digitized voice in 1 time slot

High Speed Circuit Switched Data (HSCSD)

- 2.5G enhancement
- Up to 8 slots (full user frame) allocated to one data channel
- Up to 115.2 kbps

Transparent data transmission

- User data stream can contain signaling to network
  - Allows dynamic reconfiguration of data connection (data rate, QoS)
- HSCSD data frames carry data sub-stream numbers
  - Maintains order of transmission over GSM

Non-transparent data transmission

- Only user data in data stream
  - No signaling or reconfiguration
- LLC functions performed by GSM protocols

High Speed Downlink Packet Access (HSDPA)

Higher data rates for packet data

Downlink speeds of 1.8, 3.6, 7.2, 14.0, 337 Mbps

HS-DSCH simplified for fast packet data

Power control and variable chip rate eliminated

Hybrid automatic repeat-request (HARQ)

LLC layer added between PHY and MAC (not in RLC)

Incremental redundancy

- Corrupted packets not discarded
- Retransmitted packets combined until error-free packet assembled
- Faster than waiting for uncorrupted retransmitted packet

Fast packet scheduling

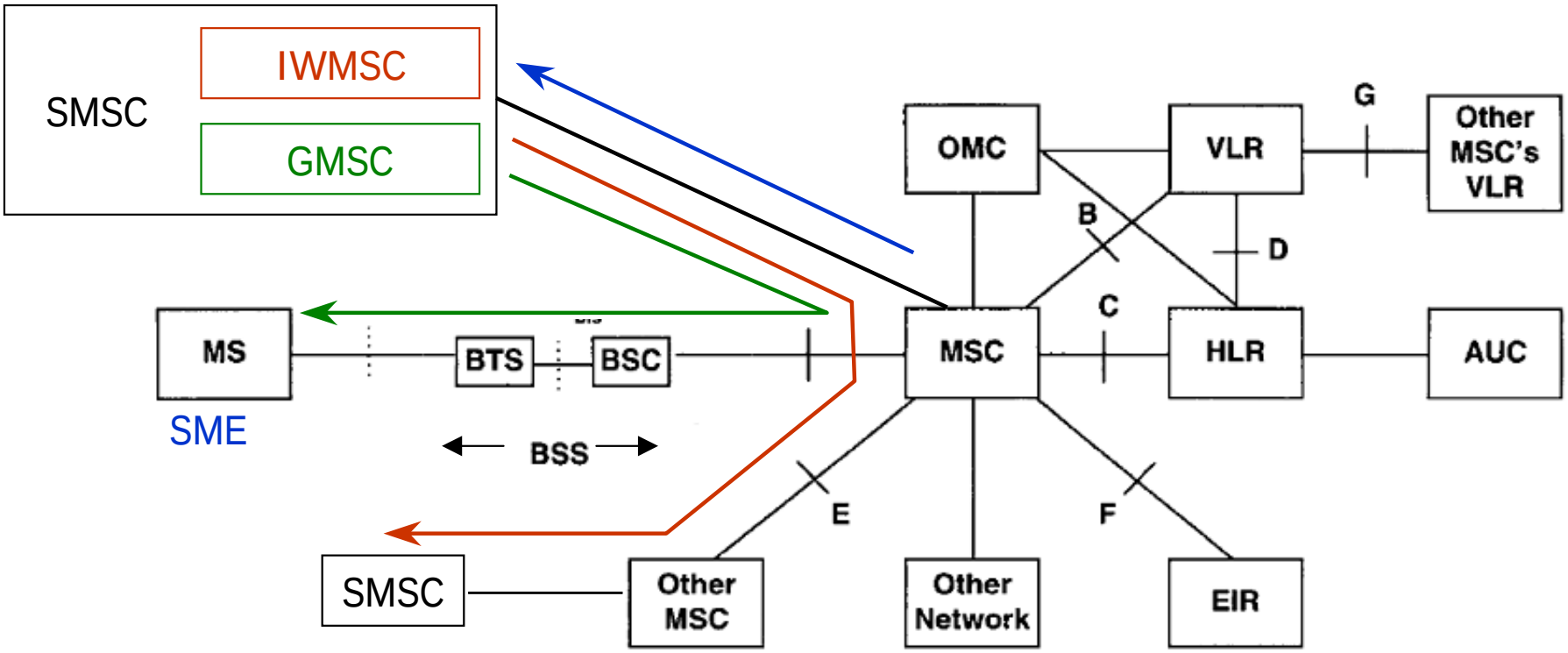
2 ms scheduling granularity (instead of 10 ms)

Transmission scheduled to UEs reporting highest power levels

Adaptive Modulation and Coding (AMC)

Modulation scheme and code rate depend on channel quality

SMS in GSM Architecture



SME: Short Messaging Entity

SMSC: Short Message Service Center

GMSC: Gateway Message Service Center

IWMSC: Interworking Message Service Center

SMS Nodes

Short Messaging Entity (SME)

Any entity that can receive or send short messages

- Fixed network element
- Mobile Station
- Another service center

Short Message Service Center (SMSC)

Store and forwarding of SMS between SME and MS

Gateway Message Service Center (GMSC)

Receives SMS in SMSC

Interrogates HLR for routing information

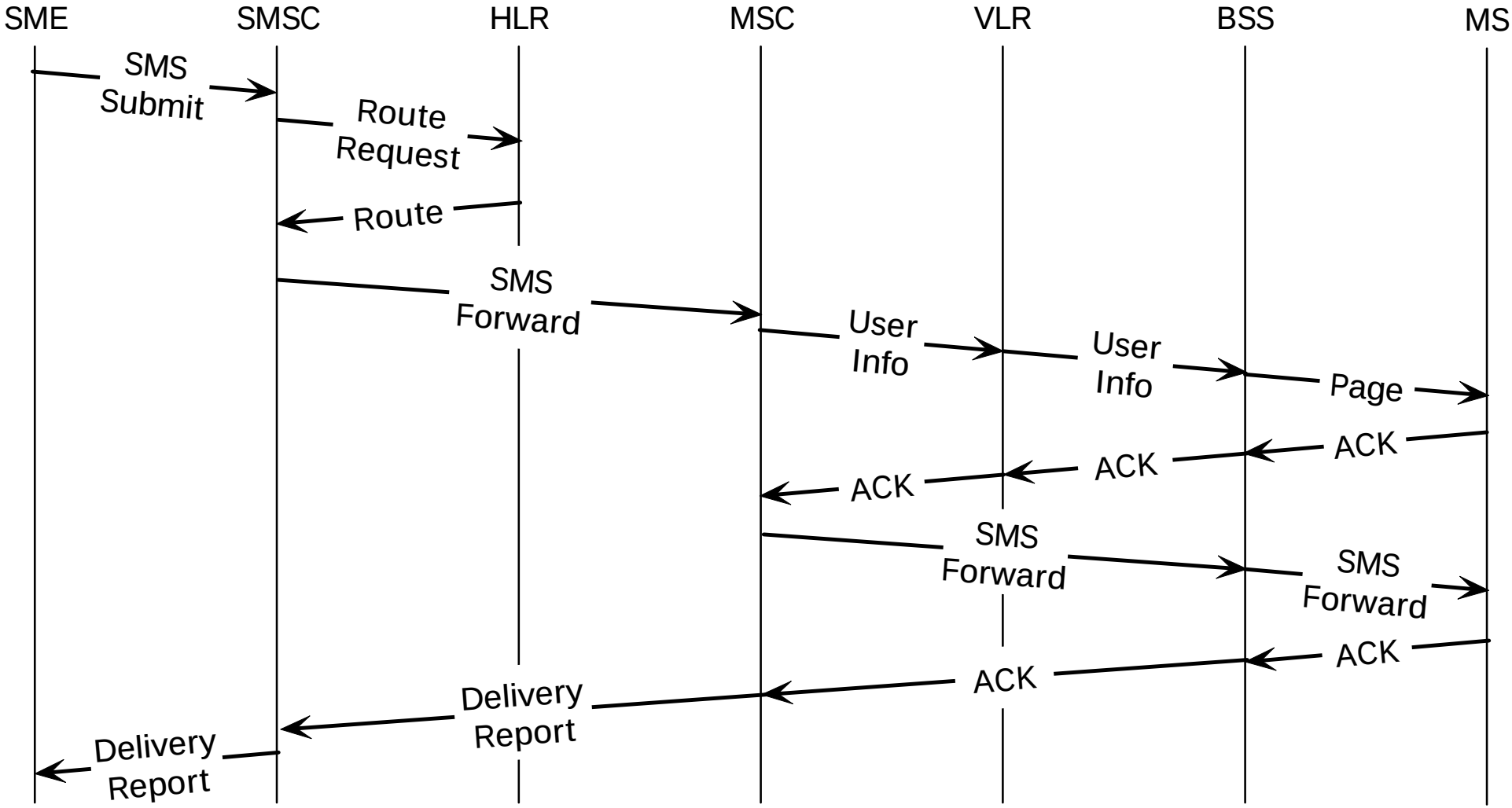
Delivers SMS to MSC for destination SME

Interworking Message Service Center (IWMSC)

Receives SMS from MSC

Delivers SMS to appropriate SMSC for forwarding

SMS Delivery to MS



CDMA

Code Division Multiple Access

- Commercial system developed by Qualcomm
- Operates on AMPS frequencies

Channelization

- 25 MHz radio band per direction
- Divide band into 1.25 MHz RF channels
- 25 MHz per cluster / 1.25 MHz per channel = 20 channels per cluster

DSSS digital transmission

- Transmit 1.2288 Mcps in 1.25 MHz radio channel
- Voice and control modulation — QPSK

Code division

- Users transmit simultaneously using independent chip sequences
  - Orthogonal (Walsh) Codes / Pseudorandom noise (PN) codes
- Receiver separates channels by decoding chip sequences

Standards

- IS-95 — now called CDMAone

Orthogonal CDMA Codes

m-dimensional vector space with inner product

$$\mathbf{U} \cdot \mathbf{V} = \frac{1}{m} \sum_{i=1}^{m} U_i \times V_i$$

m orthonormal basis vectors

$S_i, i = 1, \ldots, m$

$\mathbf{T} = \sum_{i=1}^{m} t_i \times S_i$, with coefficient t_i for any vector $\mathbf{T}$

$$S_i \cdot S_j = m \times \delta_{ij} = \begin{cases} 0, i \neq j \\ m, i = j \end{cases}$$

$$t_i = S_i \cdot \mathbf{T} = S_i \cdot \frac{1}{m} \sum_{j=1}^{m} t_j \times S_j = \frac{1}{m} \sum_{j=1}^{m} t_j \times \left(S_i \cdot S_j \right) = \frac{1}{m} \sum_{j=1}^{m} t_j \times m\delta_{ij} = t_i$$

Code scheme

Basis vector S_i is code assigned to station i

Station i transmits $t_i \times S_i$ with coefficient

Total transmission from all stations

$$t_i = \begin{cases} -1, \text{ data 0} \\ 0, \text{ no transmission} \\ +1, \text{ data 1} \end{cases}$$

$$\mathbf{T} = \sum_{i=1}^{m} t_i \times S_i$$

Example — 4-Chip CDMA

Code vectors for m = 4 stations

$$S_1 = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \quad S_2 = \begin{bmatrix} -1 \\ +1 \\ +1 \\ -1 \end{bmatrix} \quad S_3 = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix} \quad S_4 = \begin{bmatrix} -1 \\ +1 \\ -1 \\ +1 \end{bmatrix}$$

4-bit transmission levels (chips)

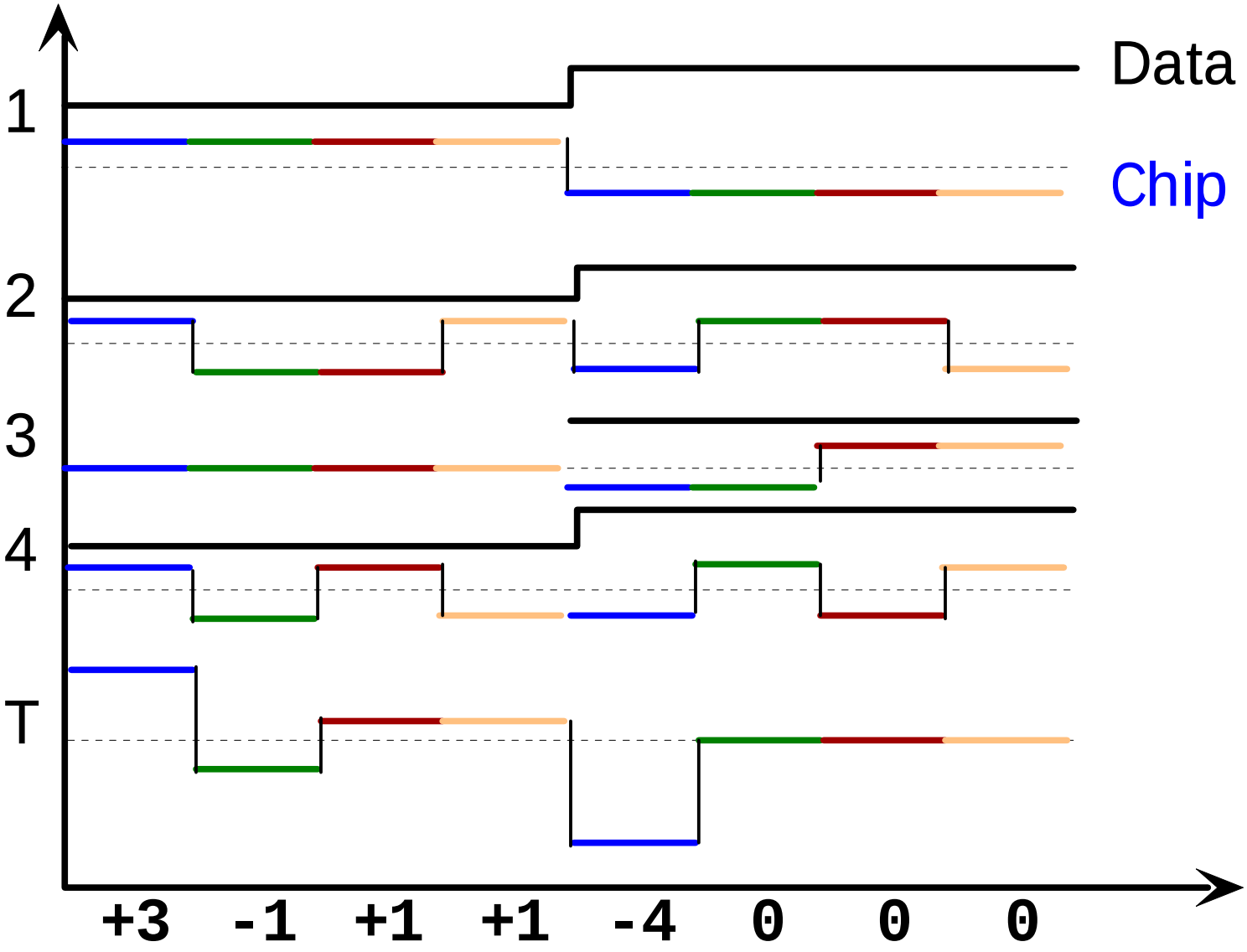
	Binary 1	Binary 0
Station 1	**–1 –1 –1 –1**	**+1 +1 +1 +1**
Station 2	**–1 +1 +1 –1**	**+1 -1 -1 +1**
Station 3	**–1 –1 +1 +1**	**+1 +1 -1 -1**
Station 4	**–1 +1 -1 +1**	**+1 -1 +1 -1**

Radio signal amplitudes added together

Example — 2-bit Transmission

Station 1	Data	0	1
	Signal	+1 +1 +1 +1	-1 -1 -1 -1
Station 2	Data	0	1
	Signal	+1 -1 -1 +1	-1 +1 +1 -1
Station 3	Data	no data	1
	Signal	0 0 0 0	-1 -1 +1 +1
Station 4	Data	0	1
	Signal	+1 -1 +1 -1	-1 +1 -1 +1
Total Transmission	Signal	+3 -1 +1 +1	-4 0 0 0

Example — 2-bit Transmission



Example — Decoding

Inner Product

$$\mathbf{U}\cdot\mathbf{V}=\frac{1}{4}\sum_{i=1}^{4}U_iV_i \qquad t_j=\mathrm{T}\cdot\mathrm{S}_j$$

First bit $\mathbf{T}=(+3,-1,+1,+1)$	$t_1=\frac{1}{4}(3,-1,+1,+1)\cdot(-1,-1,-1,-1)=\frac{1}{4}[-3+1-1-1]=-1\Rightarrow 0$ $t_2=\frac{1}{4}(3,-1,+1,+1)\cdot(-1,+1,+1,-1)=\frac{1}{4}[-3-1+1-1]=-1\Rightarrow 0$ $t_3=\frac{1}{4}(3,-1,+1,+1)\cdot(-1,-1,+1,+1)=\frac{1}{4}[-3+1+1+1]=0\Rightarrow$ no data $t_4=\frac{1}{4}(3,-1,+1,+1)\cdot(-1,+1,-1,+1)=\frac{1}{4}[-3-1-1+1]=-1\Rightarrow 0$
Second bit $\mathbf{T}=(-4,0,0,0)$	$t_1=\frac{1}{4}(-4,0,0,0)\cdot(-1,-1,-1,-1)=\frac{1}{4}[4]=+1\Rightarrow 1$ $t_2=\frac{1}{4}(-4,0,0,0)\cdot(-1,+1,+1,-1)=\frac{1}{4}[4]=+1\Rightarrow 1$ $t_3=\frac{1}{4}(-4,0,0,0)\cdot(-1,-1,+1,+1)=\frac{1}{4}[4]=+1\Rightarrow 1$ $t_4=\frac{1}{4}(-4,0,0,0)\cdot(-1,+1,-1,+1)=\frac{1}{4}[4]=+1\Rightarrow 1$

Orthogonal Walsh Codes

Walsh 0

$W_0 = \boxed{1}$ $\quad W_0' = \boxed{-1}$

Walsh 1

$$W_1 = \begin{array}{|c|c|}\hline W_0 & W_0 \\ \hline W_0 & W_0' \\ \hline \end{array} = \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & -1 \\ \hline \end{array}$$

Walsh 2

$$W_2 = \begin{array}{|c|c|}\hline W_1 & W_1 \\ \hline W_1 & W_1' \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline 1 & 1 & 1 & 1 \\ \hline 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ \hline 1 & -1 & -1 & 1 \\ \hline \end{array} = \begin{array}{|c|}\hline S_1 \\ \hline S_4 \\ \hline S_3 \\ \hline S_2 \\ \hline \end{array}$$

Walsh 3

$$W_3 = \begin{array}{|c|c|}\hline W_2 & W_2 \\ \hline W_2 & W_2' \\ \hline \end{array}$$

Walsh N

$$W_N = \begin{array}{|c|c|}\hline W_{N-1} & W_{N-1} \\ \hline W_{N-1} & W_{N-1}' \\ \hline \end{array}$$

Walsh N is $2^N \times 2^N$ matrix

Pseudo-Noise (PN) Coding

Pseudorandom Bernoulli sequence of 1 or –1

Equivalent to sequence of m coin tosses

Nearly equal number of 1 and –1 in each code

By central limit theorem

$$P_{-1} = P(-1) = \frac{1}{2} + \delta \qquad P_{+1} = P(+1) = \frac{1}{2} - \delta \qquad |\delta| < \frac{1}{\sqrt{m}}$$

Codes are "nearly orthogonal"

For codes A and B with chip patterns $C_i^{(A)}$ and $C_i^{(B)}$

$$A = B \Rightarrow \frac{1}{m}\sum_{i=1}^{m} C_i^{(A)} \times C_i^{(B)} = \frac{1}{m}\sum_{i=1}^{m} [\pm 1]^2 = 1$$

$$A \neq B \Rightarrow \frac{1}{m}\sum_{i=1}^{m} C_i^{(A)} \times C_i^{(B)}$$

$$= \frac{1}{m}\sum_{i=1}^{m} \left[P_{+1} \times P_{+1} - P_{+1} \times P_{-1} - P_{-1} \times P_{+1} + P_{-1} \times P_{-1} \right] = 4\delta^2 < \frac{4}{m}$$

Channel Coding

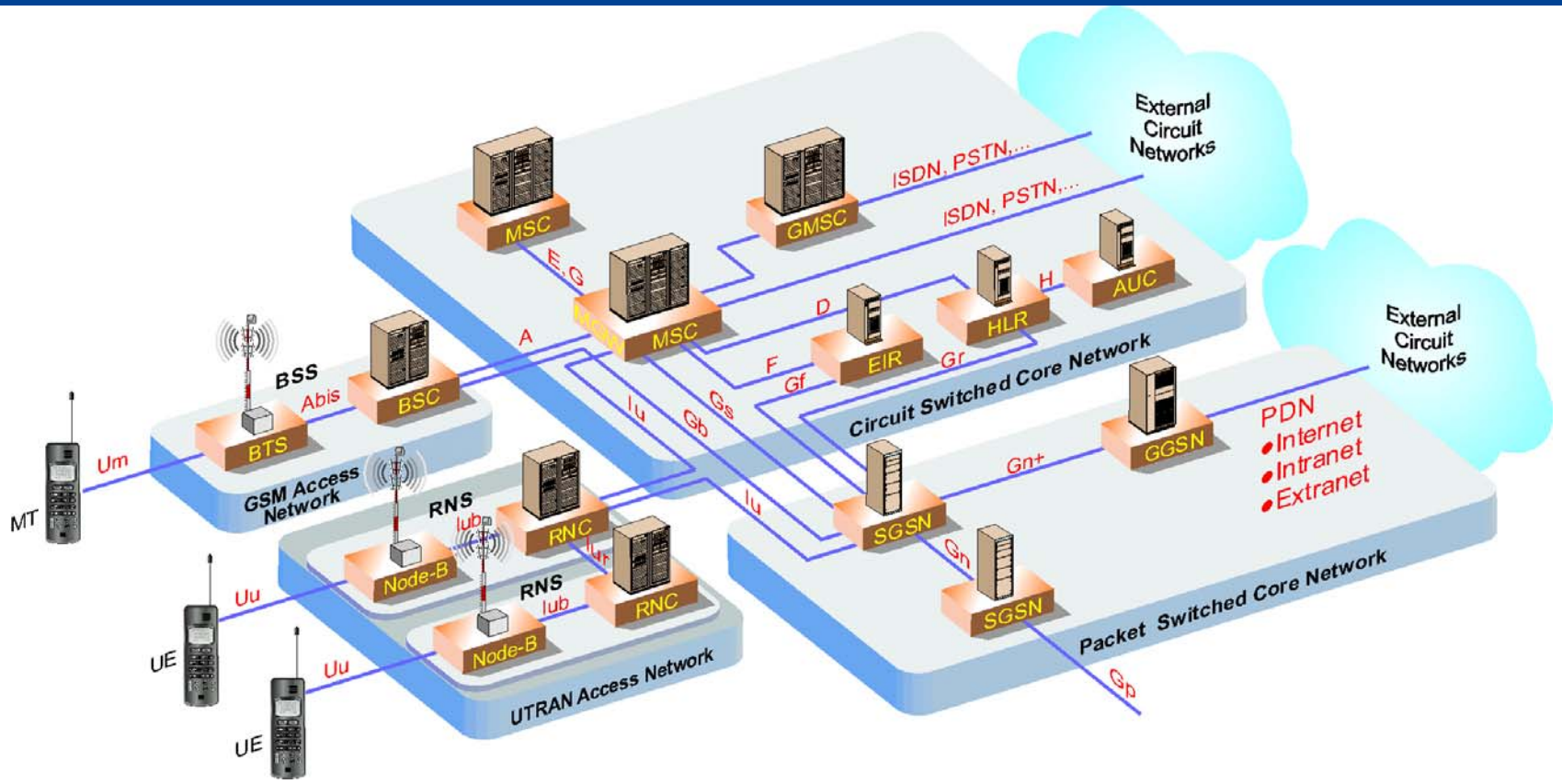
Forward channels

- 64 orthogonal Walsh codes to 64 users
  - Theoretically perfect separation between users
- All signals in same cell scrambled using PN sequence
  - Reduces interference between same Walsh code in neighboring cells
  - Short PN sequence uses cell ID as seed
  - Paging and traffic scrambled with long PN sequence before Walsh

Reverse channels

- Orthogonal codes not applicable in uplink
  - Orthogonality requires time synchronization
  - MSs transmit asynchronously
- Long PN sequence
- Stream is scrambled using short PN sequence
- Carries cell ID

GSM (2G) → UMTS (3G)



GSM Name	UMTS Name
Base Transmitter System (BTS)	Node-B
Base Controller System (BSC)	Radio Network Controller (RNC)
Base Station Subsystem (BSS)	Radio Network System (RNS)

UMTS Terrestrial Radio Access Network (UTRAN) new in 3G

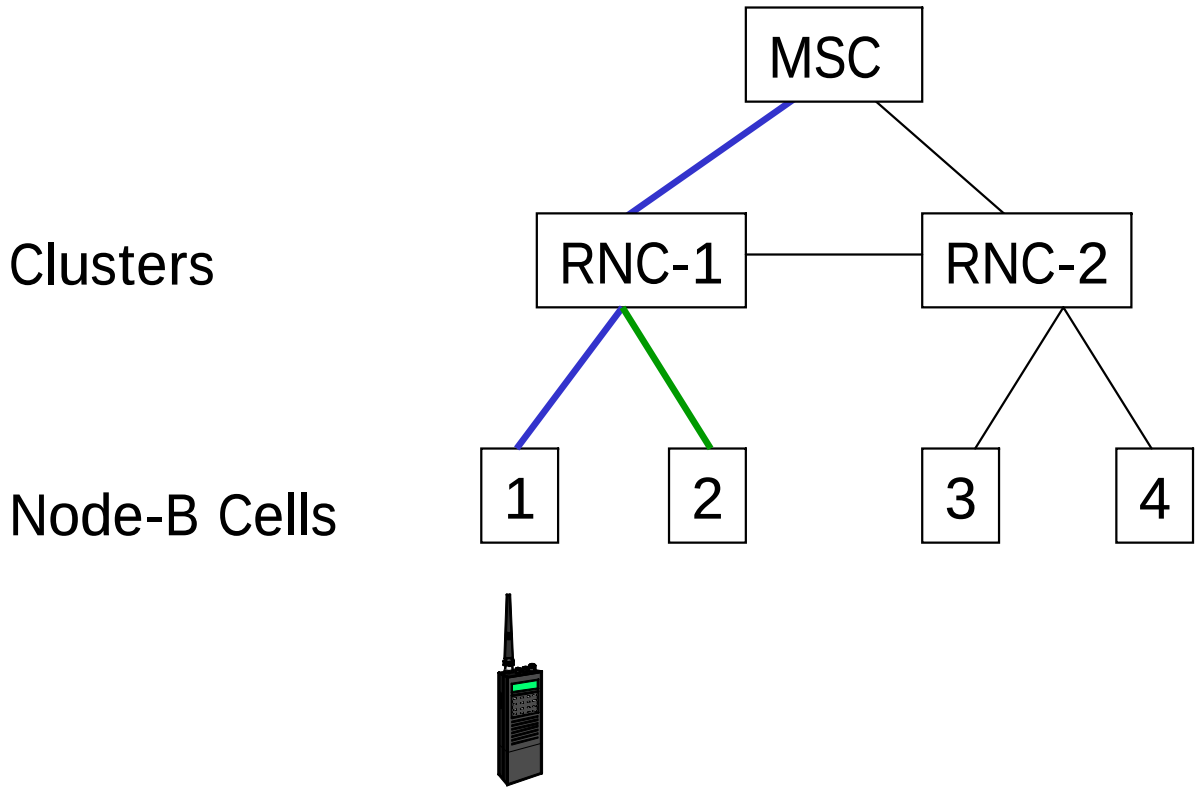
Handles mobility between related RNSs instead of MSC

3G Relocation

Serving RNC (SRNC) — RNC-1

Primary Node-B — 1

Monitoring Node-B — 2

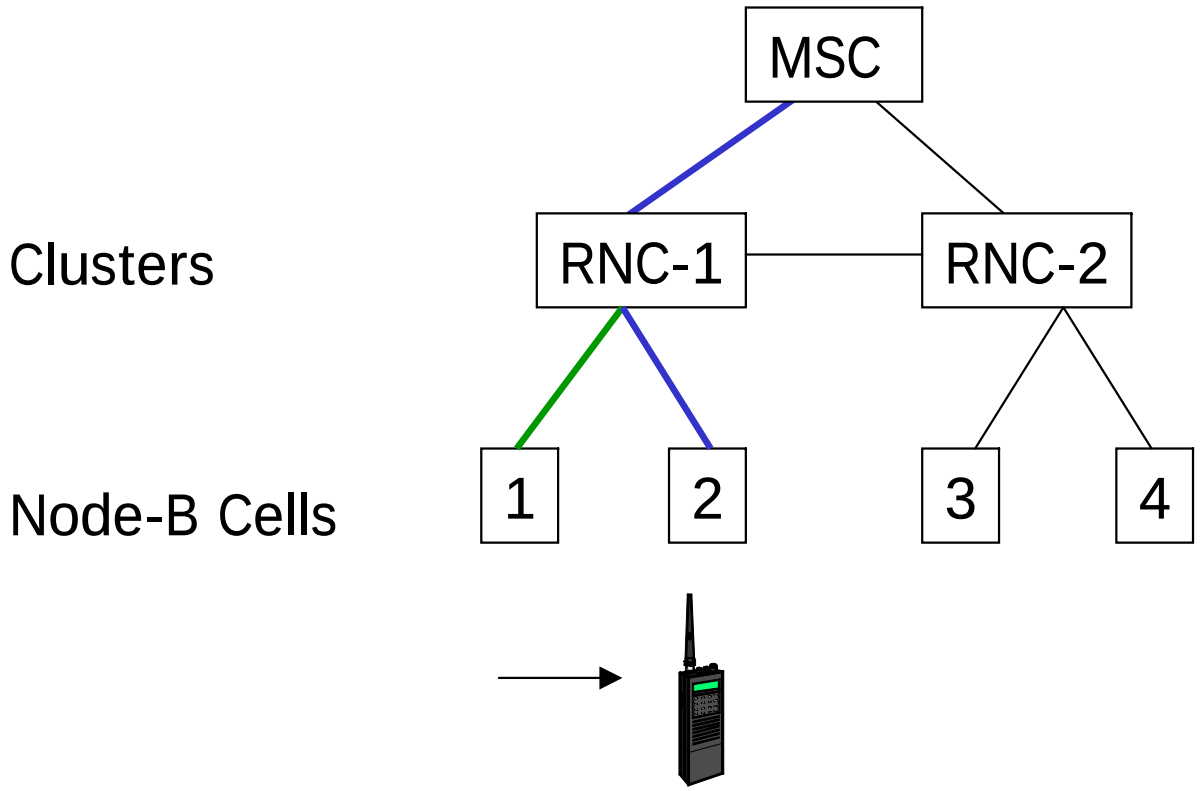


3G Relocation

Serving RNC (SRNC) — RNC-1

UE relocates to primary Node-B — 2

Monitoring Node-B — 1



3G Relocation

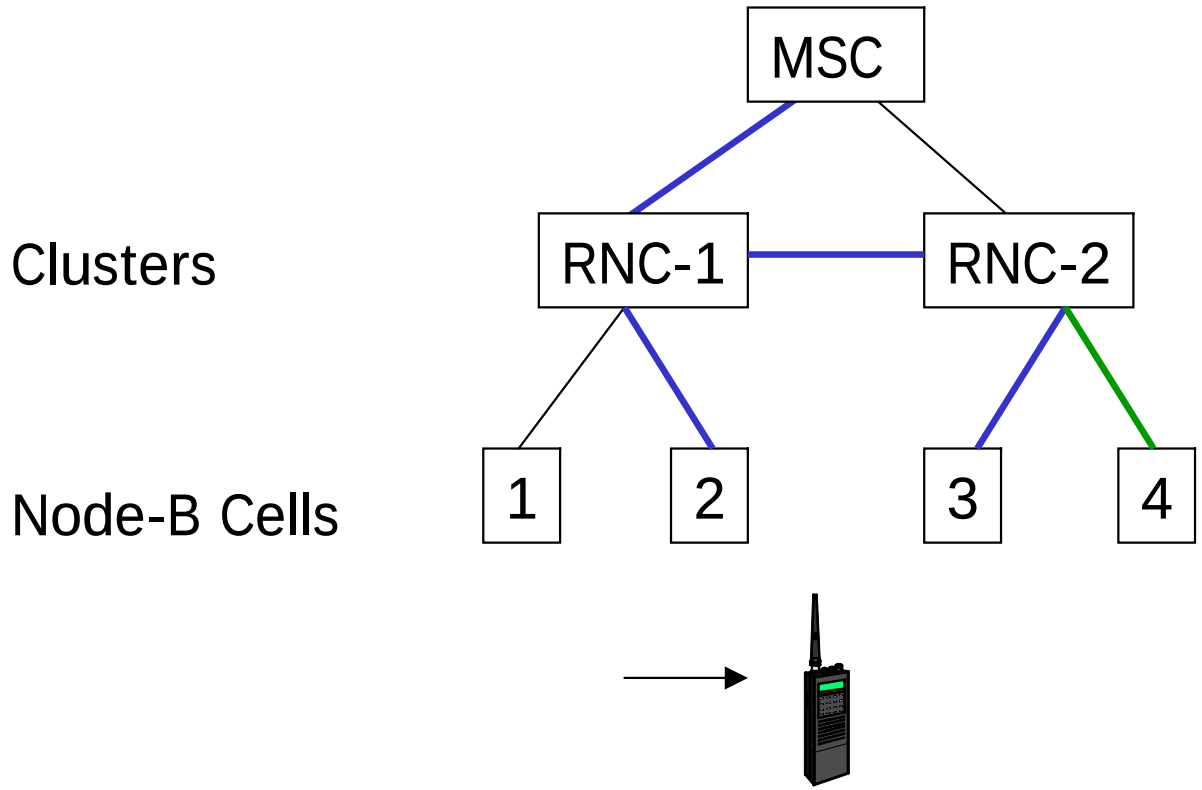
Serving RNC (SRNC) — RNC-1

Active Node-B — 2

Relaying RNC (RRNC) — RNC-2

Primary Node-B — 3

Monitoring Node-B — 4



SRNC (RNC-1)
combines data from 2 and 3

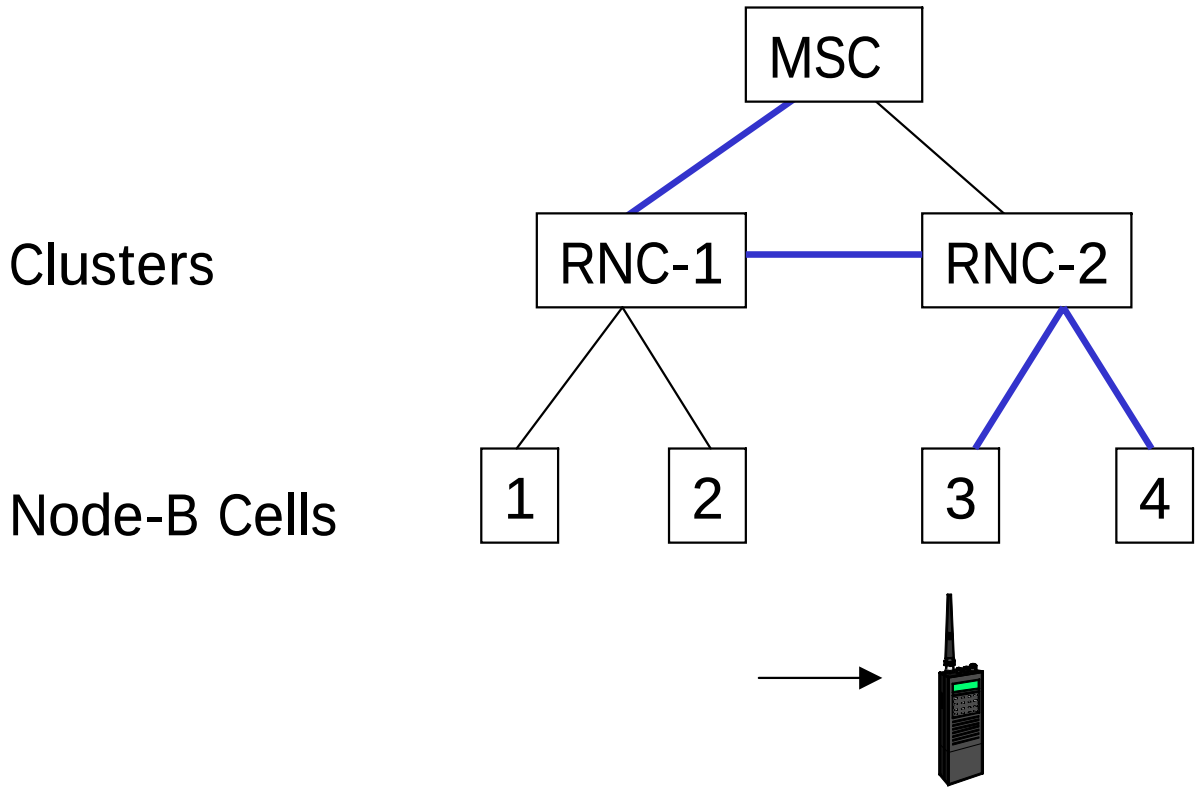
3G Relocation

Serving RNC (SRNC) — RNC-1

No active Node-B

Relaying RNC (RRNC) — RNC-2

Primary Node-Bs — 3 + 4



RNC-2
combines data from 3 and 4

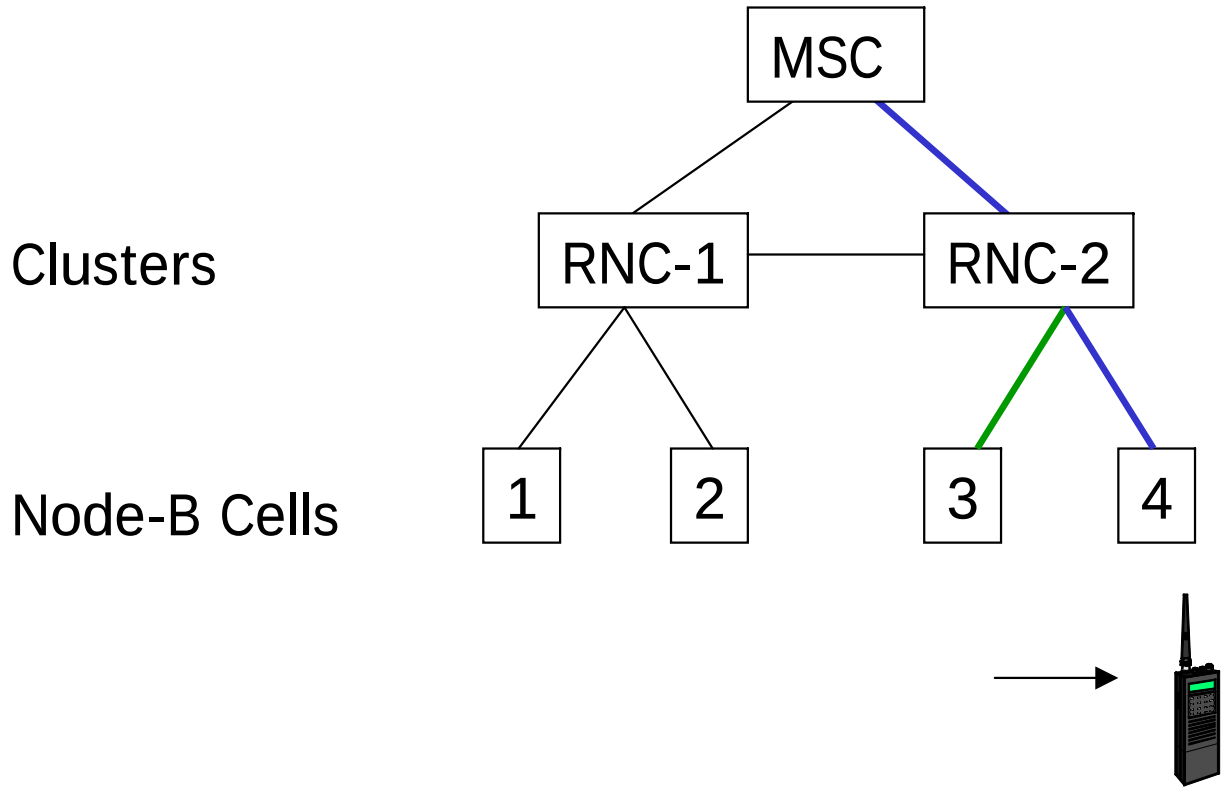
SRNC (RNC-1)
receives combined data from RNC-2

3G Relocation

Serving RNC (RRNC) — RNC-2

Monitoring Node-B — 3

Primary Node-B — 4



3G → 4G

Initial planning for 4G

International Mobile Telecommunications Advanced (IMT-Advanced)

Conceived as general network supporting mobility

- Data + Voice over IP

Convergence with Next Generation Networks (NGN)

4G objectives

Higher network capacity than 3G

Spectral efficiency (high bps / Hz and bps / Hz /site)

100 Mbps for moving client and 1 Gbps for stationary client

100 Mbps between any two points in world

Smooth handoff across heterogeneous networks

Global roaming across multiple networks

QoS for multimedia support — audio, HDTV, etc

Interoperability with existing wireless standards

All IPv6 packet switched network — eliminate circuit mode entirely

Long Term Evolution (LTE)

Extension ("evolution") of GSM / UMTS

- Marketed as 4G — does not meet all IMT/NGN standards
- Attaches "e" (evolved) to UMTS terminology

Improved radio interface + data rates

- 299.6 Mbps downstream
- 75.4 Mbps upstream

Improved mobility

- Supports MS moving at 500 km/h

Voice over LTE (VoLTE)

- Packet switched voice over IP
- Replacement for circuit switched voice network
- Enhanced voice encoding as VBR stream

Circuit-switched fallback (CSFB)

- Intermediate migration path
- Data over LTE + circuit switched voice

System Architecture Evolution (SAE)

LTE core network architecture

- All IP network
- Replaces GPRS packet network of GSM/UMTS

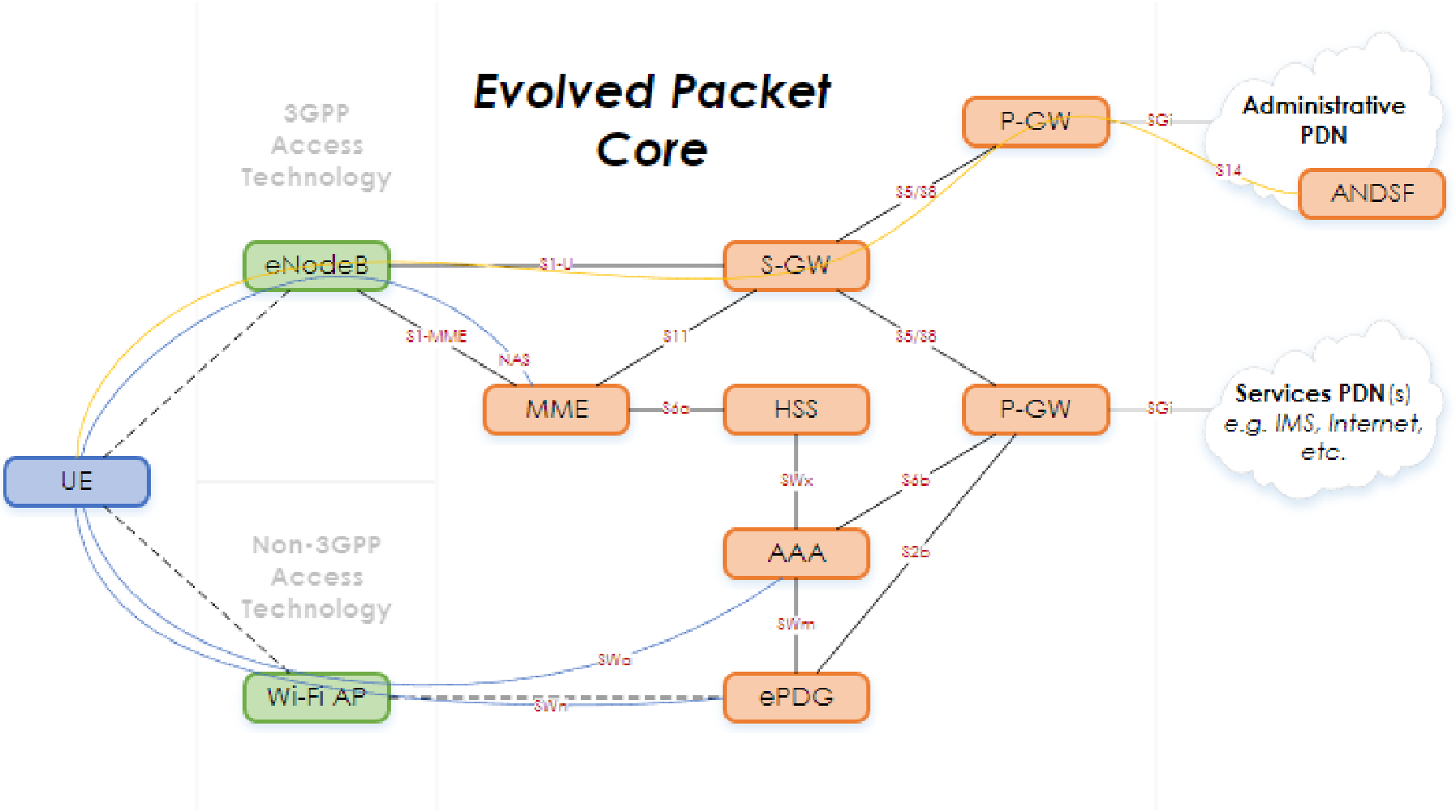
Evolved UTRAN (e-UTRAN)

- Replaces UTRAN in 3G networks
- eNodeB — replaces UMTS NodeB in e-UTRAN

Evolved Packet Core (EPC)

- Mobility Management Entity (MME) — replaces GSM VLR
- Home Subscriber Server (HSS) — replaces GSM HLR
- Serving Gateway (SGW)
  - Routes and forwards user data packets within cellular network
  - Replaces hierarchical network of MSC / SGSN / nodes in GSM/UMTS
- Packet Data Network Gateway (PGW)
  - Routes and forwards user data packets to Internet
  - Replaces GGSN in GSM/UMTS

Evolved Packet Core (EPC) Interfaces



Ref: https://en.wikipedia.org/wiki/System_Architecture_Evolution

4G → 5G

Non-Standalone mode (NSA)

- Supports Mobile Broadband (eMBB)
- Enhanced 4G as a 5G use case
- Provides 5G New Radio (NR) add-on to 4G LTE radio
- Higher data rates + lower latency
  - Up to 10 Gbps

5G Next Generation Core (NGC)

- New abstract networking model
  - Service Based Architecture (SBA)
  - Network functions defined as services — not hardware
- Extensive implementation of
  - Software Defined Network (SDN)
  - Network Function Virtualization (NFV)
- Network designed by standards of software engineering
- Support for Internet of Things (IoT)

Congestion and Flow Control in TCP

Flow Control and Congestion Control

Flow control

Sender avoids overflow of receiver buffer

Congestion control

All senders avoid overflow of intermediate network buffers

Buffer fill rate

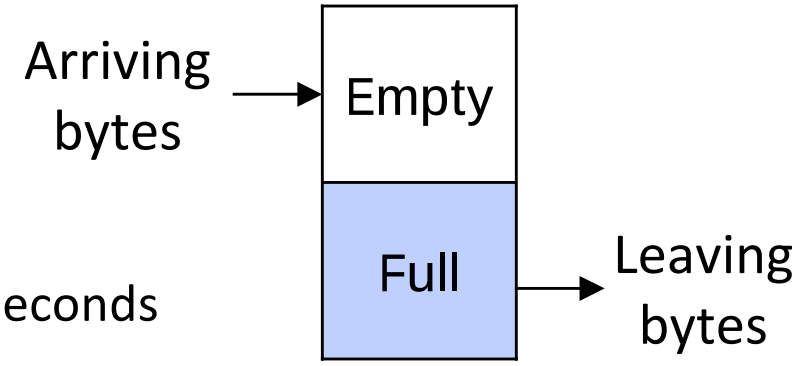
Bytes / second arriving from network

Buffer empty rate

Bytes / second leaving to network or application layer

Buffer file time

$$T_{overflow} = \frac{\text{buffer size}}{\text{buffer fill rate} - \text{buffer empty rate}}$$



Example

$$T_{overflow} = \frac{64\ \text{KB}}{8\ \text{KB/sec} - 4\ \text{KB/sec}} = \frac{64\ \text{KB}}{4\ \text{KB/sec}} = 16\ \text{seconds}$$

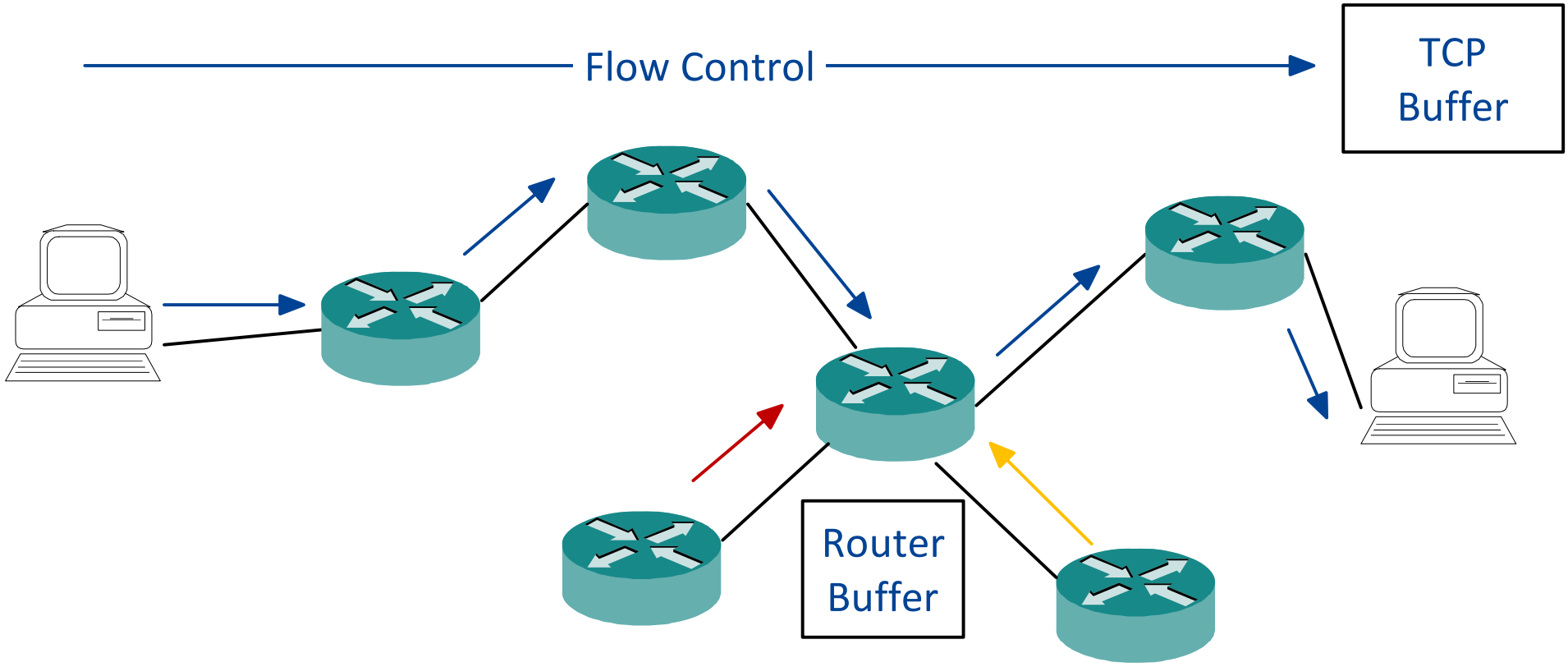
Congestion Control

Flow control

Avoid overflow in TCP receiver buffer

Congestion control

Avoid overflow in router buffers



Queuing Theory

Assumptions

Segments arrive independently (Poisson statistics)

Random length (bytes)

Average arrival rate in steady state — Packets/sec, Bytes/sec, or bps

Segments leave independently (Poisson statistics)

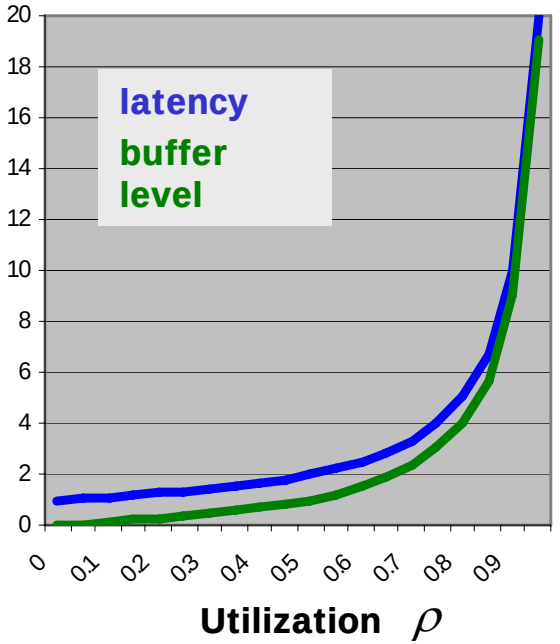
Average emptying rate in steady state — Packets/sec, Bytes/sec, or bps

Results

$$\rho = \text{Utilization} = \frac{\text{arrival rate}}{\text{empty rate}}$$

$$\text{Latency} = \frac{1}{\text{empty rate} - \text{arrival rate}} = \frac{1}{\text{empty rate}}\left(\frac{1}{1-\rho}\right)$$

$$\text{Buffer Level} = \text{Latency} \times \text{arrival rate} = \frac{\rho}{1-\rho}$$



Example

Arrival rate = 90 packets / sec Empty rate = 100 packets

Utilization = (90 packets / sec) / (100 packets / sec) = 0.9 = 90%

Buffer level = 0.9 / (1 – 0.9) = 9 packets in buffer

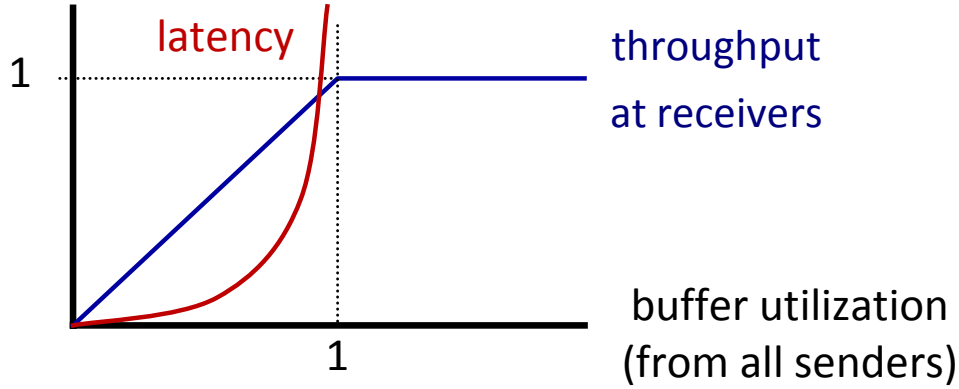
Latency = 1 / (100 packets / sec) × 9 packets in buffer = (9 / 100) sec = 0.09 sec

Buffer Throughput

(Over)-simplified throughput model

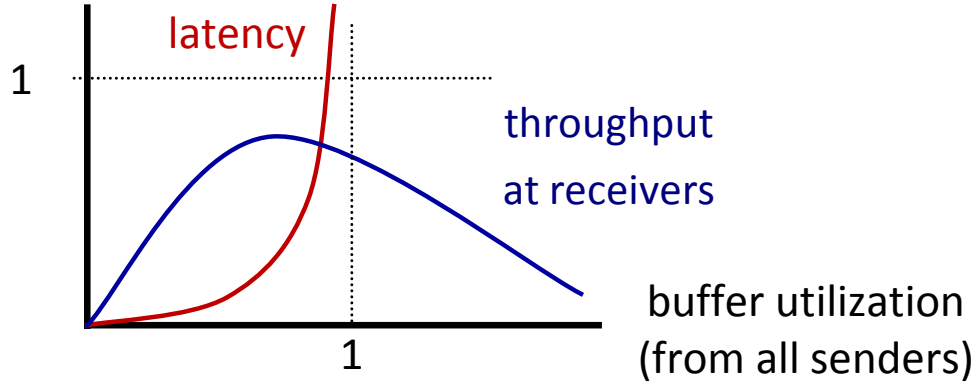
$$\text{throughtput} = \frac{\text{receive rate}}{\text{maximum receive rate}}$$

$$\text{buffer utilization} = \frac{\text{arrival rate}}{\text{empty rate}}$$



$$\text{goodput} = \frac{\text{receive rate (error-free in-order)}}{\text{maximum receive rate}}$$

Realistic throughput behavior



High arrival rate at buffer

Longer latency + overflow

Sender timeouts

Re-transmit ⇒ more segments ⇒ higher arrival rate at buffer

TCP Flow Control

Source window

Initial source window = maximum number of "unACKed" bytes

Determined by congestion + flow control

Destination window

Number of bytes receiver can accept

Determined by available space in receiver buffer

Buffer level = Previous level + arriving bytes – bytes read by App

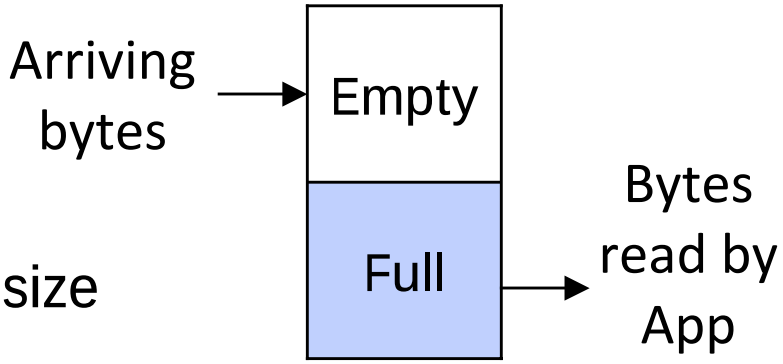
Application reads too slowly ⇒ decrease destination window

Sliding window

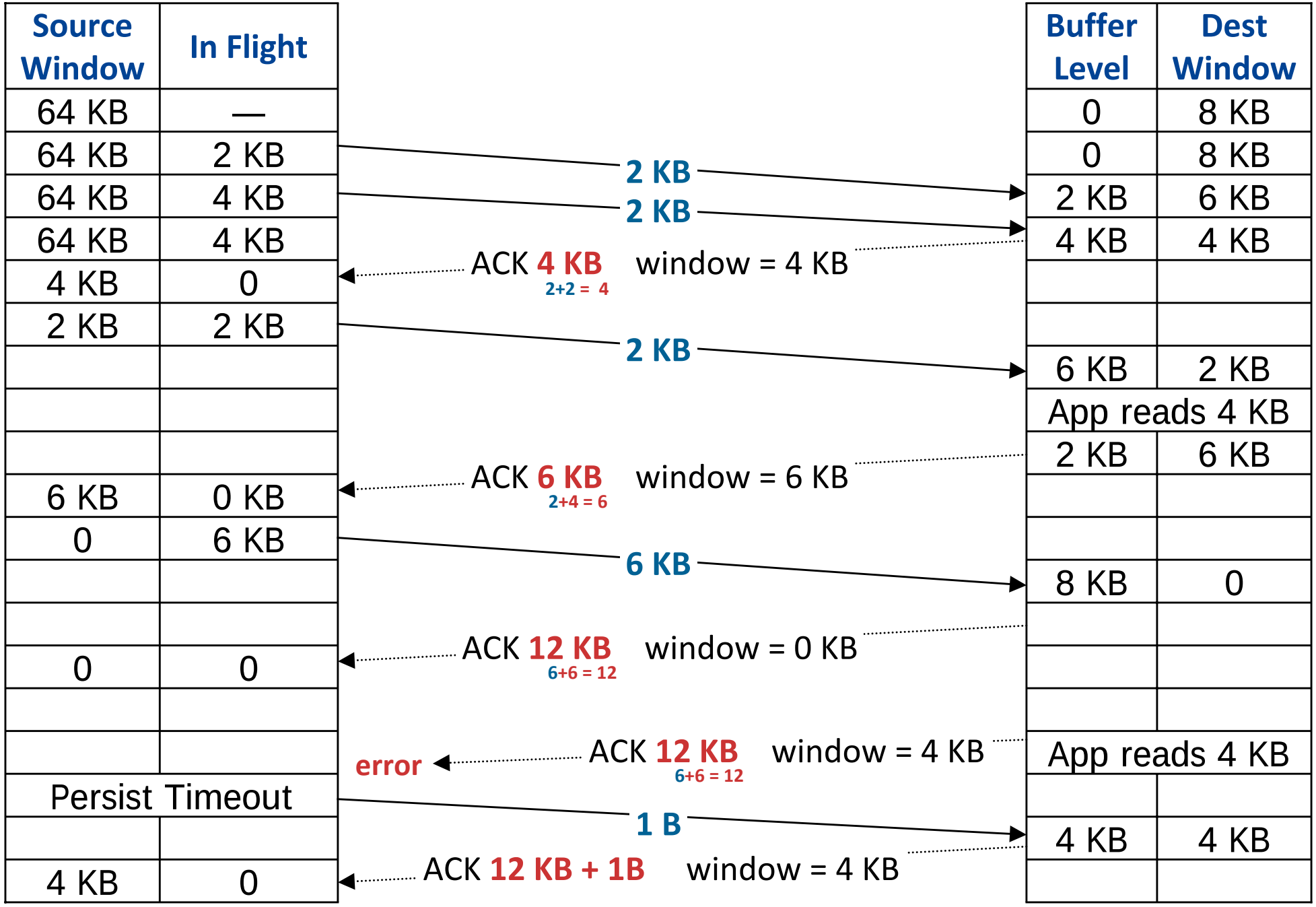
Window field in TCP header

Number of bytes receiver will accept

Receiver discards bytes above window size



Flow Control Example

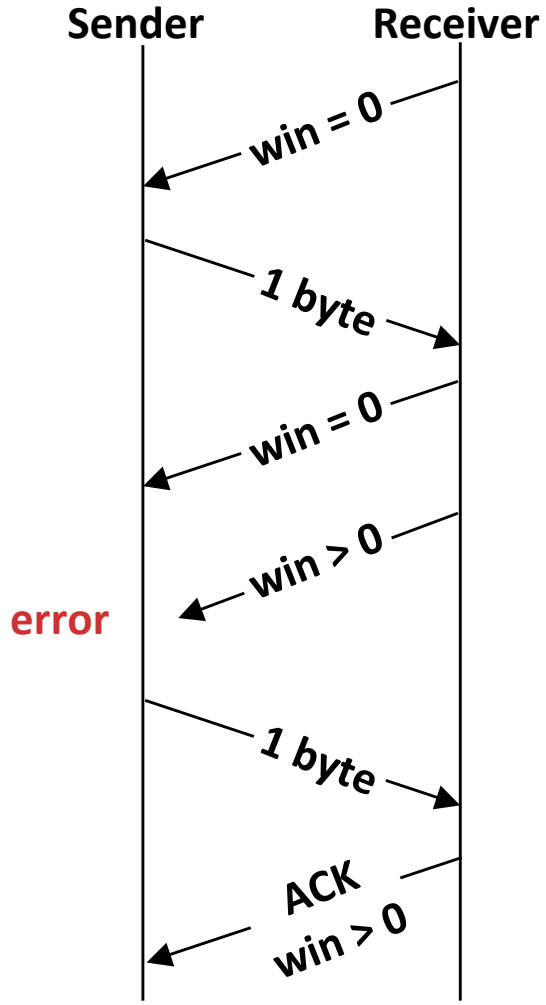


Receive Window Bugs — 1

Bug — deadlock

Receiver advertises window = 0

Window update with window > 0 is lost → deadlock



Fix — persist timeout

Sender attempts small segment

ACK contains new window size

Receive Window Bugs — 2

Silly Window Problem

Application reads received data slowly

Receiver advertises small window

Data bytes ~ header bytes

More segments / file transfer ⇒ larger total traffic (data + headers)

Nagle Algorithm — bug fix for Silly Window

Sender accumulates application data — sends large segments

Works badly with Telnet (requires small segments)

Receiver side bug fix

Receiver keeps 0 window size until it can advertise large window

TCP Congestion Control

End-to-end congestion control

Based on host estimates

No feedback from intermediate network nodes

Slow-start

Begin session with low transmission rate

Increase rate until timeouts begin

Fast retransmit

Do not wait for timeout

Re-transmit after duplicate ACKs (dupACKs)

Congestion avoidance

Limit transmission rate after duplicate ACKs

Growth rate of transmission rate slows

Slow-Start

Congestion window (cwnd)

Source window

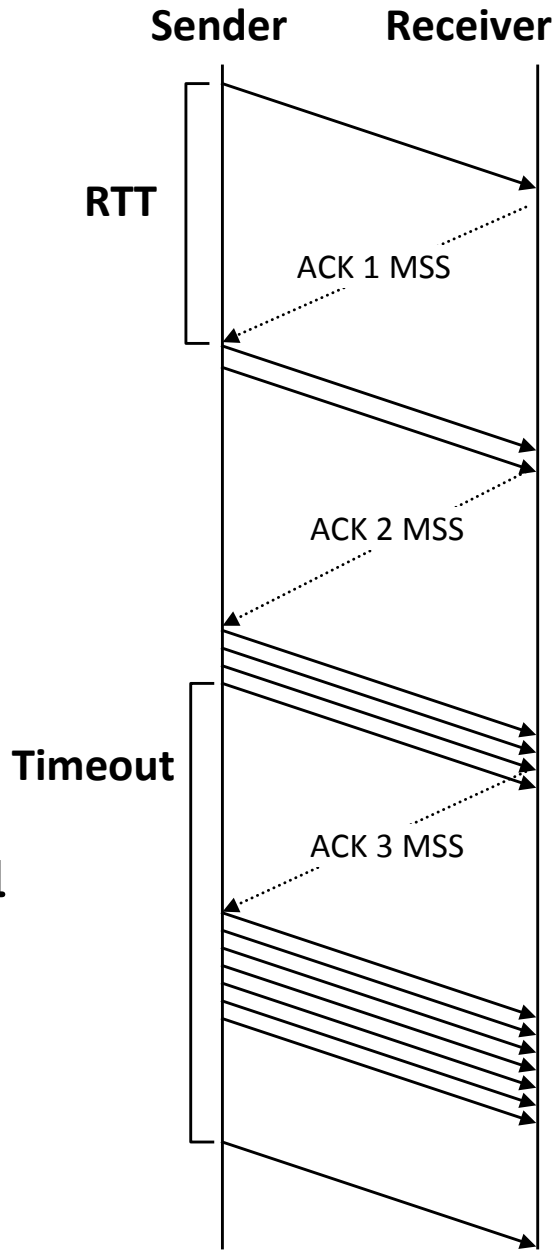
Maximum number of "unACKed" bytes

Initial **`cwnd`** = 1 MSS (maximum segment size)

Data rate = 1 MSS / RTT

RTT = round trip time = time from send to ACK

Maximum **`cwnd`** = destination window

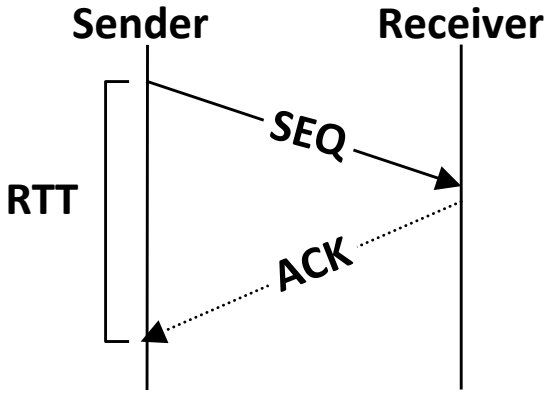


Exponential growth

```
On (ACK)
    cwnd ← cwnd + size of data ACKed
    if (cwnd > maximum cwnd)
        cwnd ← max cwnd
On (ACK timeout)
    cwnd ← initial cwnd = 1 MSS
```

Computing TCP's Retransmission Timer — RFC 2988

Initialize



RTO ← 3 seconds

G ← clock granularity (typically 500 ms)

R ← first **RTT** measurement (round trip time)

SRTT ← **R**

RTTVAR ← **R/2**

RTO ← **max(1 sec, SRTT + max(G, 4 * RTTVAR))**

Update after measurements R'

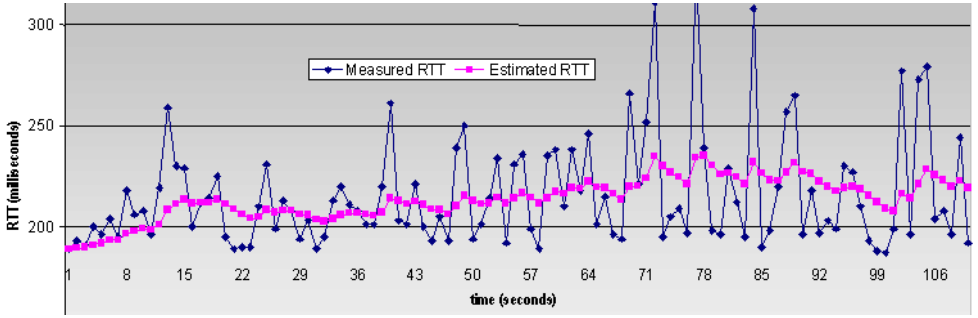
RTTVAR ← **(1 - β) * RTTVAR + β * |SRTT – R'|**

SRTT ← **(1 - α) * SRTT + α * R'**

RTO ← **max(1 sec, SRTT + max (G, 4 * RTTVAR))**

α = 1/8

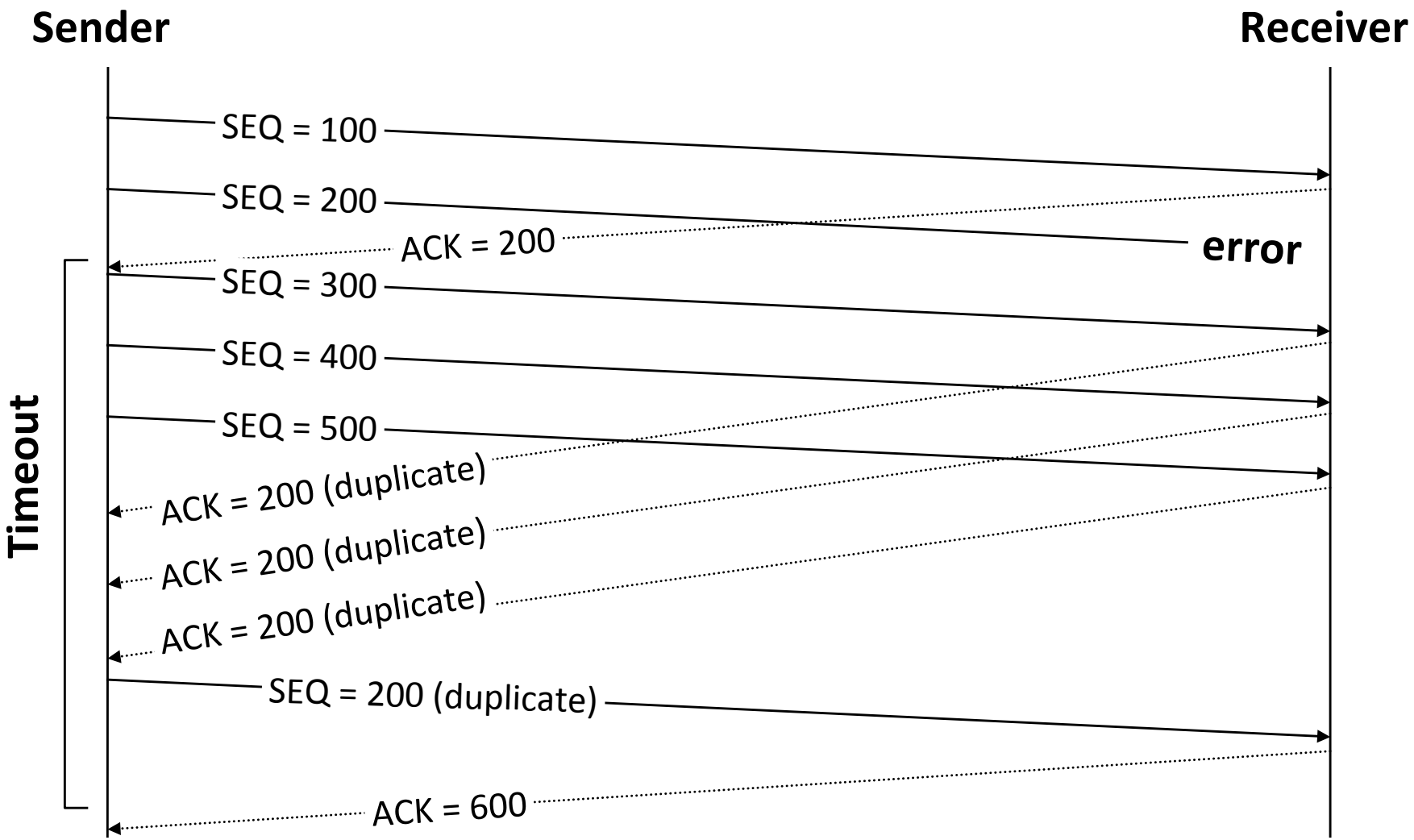
β = 1/4



Fast Retransmit

Better performance with RTO >> RTT

3 duplicate ACKs (dupACKs) for segment ⇒ re-send segment



Congestion Avoidance

Reno protocol

Slow start phase

```
On (ACK && cwnd < ssthresh)
    cwnd ← cwnd + size of data ACKed
On (ACK timeout)
    ssthresh ← cwnd
    cwnd ← initial cwnd = 1 MSS
    RTO ← 2 * RTO
```

Congestion avoidance phase

```
On (ACK && cwnd > ssthresh)
    cwnd ← cwnd + 1 MSS
```

Fast retransmit with fast recovery

```
On (3 dupACKs)
    ssthresh ← cwnd / 2
    cwnd ← ssthresh + 3
    retransmit, wait 1 RTT → continue
```

If dupACKs > 3
cwnd++ on each dupACK

TCP Sender with Reno — 1

```
// initialize
    SEQ = ISN + 1
    SendBase = ISN + 1
    InFlight = 0
    cwnd = 1 MSS
    Set ssthreshold large (local policy)
    RTO = timeout interval

on (new data from application)
    Prepare data segment:sequence number = SEQ
    if InFlight < min{cwnd,SendWindow,RecvWindow)
        Pass segment to IP
        SEQ = SEQ + length(data)
        InFlight = InFlight + length(data)
        if !(timer running) timer = RTO
```

TCP Sender with Reno — 2

```
if (receive ACK = y)
    stop timer
    if (y > SendBase)
        dupACK = 0
        newACKs = y - SendBase                  // bytes ACKed
        SendBase = y
        InFlight = InFlight - newACKs
        if (cwnd < ssthresh) cwnd = cwnd + newACKs
        else cwnd = cwnd + 1 MSS
        if (InFlight > 0) timer = RTO
```

TCP Sender with Reno — 3

```
        // if (y > SendBase)
        else
            dupACK++
            if (dupACK = 3)
                SEQ = SendBase = min{unACKed SEQ} and retransmit
                timer = RTO
                ssthresh = cwnd / 2
                cwnd = ssthresh + 3
                wait 1 RTT    // wait for ACK of resent packet
            if (dupACK > 3)
                cwnd = cwnd + 1 MSS and resend again
    if (timeout)
            SEQ = SendBase = min{unACKed SEQ} and retransmit
            ssthresh = cwnd
            cwnd = initial cwnd = 1 MSS
            RTO = 2 * RTO
            timer = RTO
```

TCP Receiver with Reno — 1

```
// initialize
    Set RecvWindow = receiver buffer size
    expected = Sender ISN + 1
    ack_buffer = 0
    ack_max (local policy: delayed ACK trigger)
    ack_delay = 250 msec (local policy: < 500 msec)
    Start ACK delay timer = ack_delay

if (ACK delay timer = 0 && ack_buffer > 0)
    Send ACK = expected with updated RecvWindow
    ACK delay timer = ack_delay
    ack_buffer = 0
```

TCP Receiver with Reno — 2

```
if (receive SEQ = x)
    if (x = expected && error-free)
        expected = expected + length(data)
        if (NACK = 1)
            Send ACK = expected with updated RecvWindow
            ACK delay timer = ack_delay
            ack_buffer = 0
            NACK = 0
        else if (ack_buffer < ack_max)
            nextACK = expected
            ack_buffer++
        else if (ack_buffer = ack_max)
            Send ACK = expected with updated RecvWindow
            ACK delay timer = ack_delay
            ack_buffer = 0
    else
        Send ACK = expected with updated RecvWindow
        ACK delay timer = ack_delay
        NACK = 1
```

Reno Example — 1

Counting in bytes: SEQ = SEQm × (1 MSS) ACK = ACKm × (1 MSS)

ssthresh = 32 Receiver sends ACK for every 4 packets (or ACK delay)

Sender				Receiver	
cwnd	**in-flight**	**SEQm Sent**	**Packets Sent**	**ACKm Sent**	**New ACKs**
1	1	1	1 (1)		
				2	1
1+1 = 2	0				
2	2	2	2 – 3 (2)		
				4	2
2+2 = 4	0				
4	4	4	4 – 7 (4)		
				8	4
4+4 = 8	0				
8	8	8	8 – 15 (8)		
				12	4
8+4 = 12	4				

Reno Example — 2

cwnd	in-flight	SEQm Sent	Packets Sent	ACKm Sent	New ACKs
12	4				
				16	4
12+4 = 16	0				
16	16	16	16 – 31 (16)		
				20	4
16+4 = 20	12				
20	20	32	32 – 39 (8)		
				24	4
20+4 = 24	16				
24	24	40	40 – 47 (8)		
				28	4
24+4 = 28	20				
				32	4
28+4 = 32	16				
32	32	48	48 – 63 (16)		

Reno Example — 3

cwnd	in-flight	SEQm Sent	Packets Sent	ACKm Sent	New ACKs
32	32	48	48 – 63 (16)		
				36	4
32+1 = 33	28				
33	32	64	64 – 67 (4)		
				40	4
33+1 = 34	28				
				44	4
34+1 = 35	24				
				48	4
35+1=36	20				
				52	4
36+1 = 37	16				
				56	4
37+1 = 38	12				

Reno Example — 4

cwnd	in-flight	SEQm Sent	Packets Sent	ACKm Sent	New ACKs
38	20	68	68 – 75 (8)		
				60	4
38+1 = 39	16				
				60	0
39	16				
				60	0
39	16				
		3rd dupACK		60	0
ssthresh ← 19 19 + 3 = 22	16				
22	16	60	60	Retransmit 1 packet	
				76 (cumulative)	16
23	0				
23	23	76	76 – 98 (23)		

Reno: First Alternative Scenario

cwnd	in-flight	SEQm Sent	Packets Sent	ACKm Sent	New ACKs
1	1	1	1 (1)		
				2	1
1+1 = 2	0				
2	2	2	2 – 3 (2)		
				3	1
2+1 = 3	1				
3	3	4	4 – 5 (2)		
				3	0
3+0 = 3	3				
				3	0
cwnd = in-flight ⇒ no more packets ⇒ no 3 dupACKs ⇒ Timeout					
1	0				
1	1	3	3 (1)		
				4	1
1+1 = 2	0				

Reno: Second Alternative Scenario – 1

cwnd	in-flight	SEQm Sent	Packets Sent	ACKm Sent	New ACKs
1	1	1	1 (1)		
				2	1
1+1 = 2	0				
2	2	2	2 – 3 (2)		
				4	2
2+2 = 4	0				
4	4	4	4 – 7 (4)		
				7	3
4+3 = 7	1				
7	7	8	8 – 13 (6)	7 packets can cause 7 dupACKs	
				7	0
				7	0
			3rd dupACK	7	0
retransmit	7	7	7 (1)		
ssthresh ← 3 3 + 3 = 6	8				

Reno: Second Alternative Scenario – 2

cwnd	in-flight	SEQm Sent	Packets Sent	ACKm Sent	New ACKs
			4th dupACK	7	0
7	8				
			5th dupACK	7	0
8	8				
			6th dupACK	7	0
9	8				
9	9	7	7 (1)		
				14 (cumulative)	7
9+1 = 10	0				
10	14	10	14 – 23 (10)		

Selective Acknowledgment Option

Selective ACK (SACK)

Permits ACK for segments with gaps

Option negotiated between hosts

Defined in RFC 2018

Example

Last ACK = 5000

Send 8 segments × 500 data bytes / segment

Case 1

First 4 segments received and last 4 dropped

Receiver returns normal ACK = 5000 + 4 * 500 = 7000

No SACK option field

Case 2

First segment lost and 7 segments received

For each segment receiver returns segment with

ACK = 5000

SACK option field with start + end ACK

Data	ACK	Option Field	
		Start	End
5000	—	—	—
5500	5000	5500	5999
6000	5000	5500	6499
6500	5000	5500	6999
7000	5000	5500	7499
7500	5000	5500	7999
8000	5000	5500	8499
8500	5000	5500	8999

Active Queue Management (AQM)

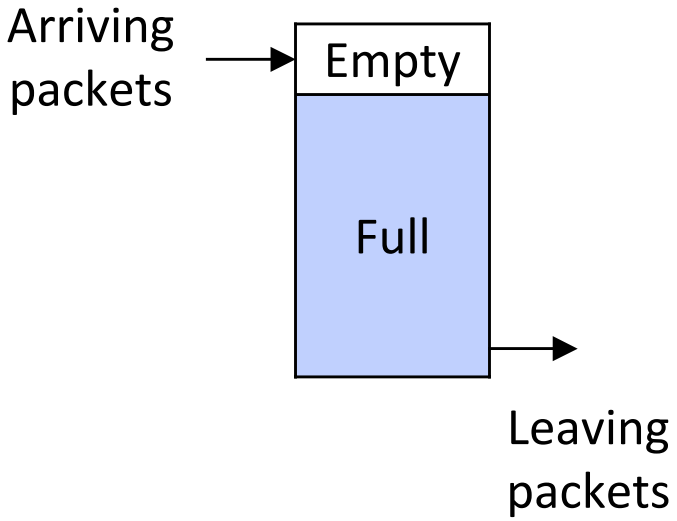
Standard Queue

At receiver

Full buffer → drop excess packets

At sender

No ACK → timeout → signal congestion



Random Early Detection (RED)

Router

Detects congestion early

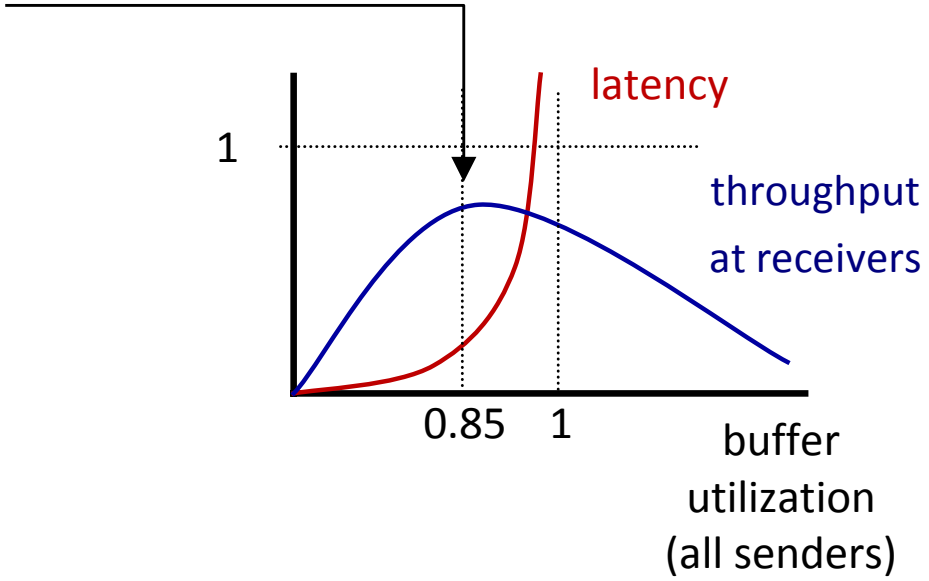
Drops random packets

Sender

Sees dupACKs or timeout

Assumes congestion

Lowers **`cwnd`**



RED Algorithm

Algorithm

```
for each packet arrival
    calculate avg = average queue size
    if min_th ≤ avg < max_th
        calculate probability p_a
        with probability p_a:
            mark arriving packet for drop
    else if max_th ≤ avg
        mark arriving packet for drop
```

Parameters

max_p = maximum mark probability (0.1 to 0.5)

min_{th} ~ 5

max_{th} ~ 30

$p_b \leftarrow max_p (avg - min_{th}) / (max_{th} - min_{th})$

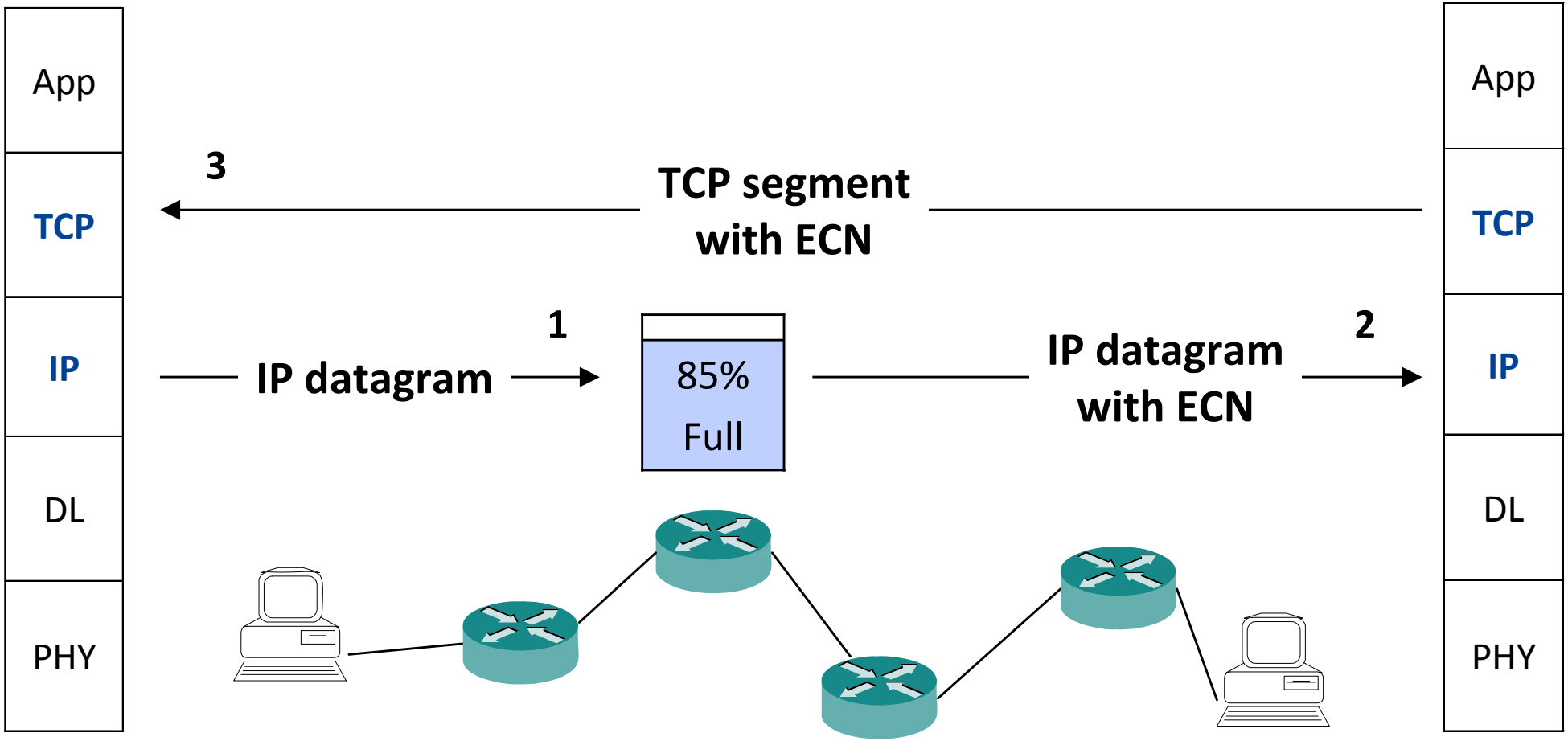
$p_a \leftarrow p_b / (1 - count \times p_b)$

count = number of consecutive dropped packets

AQM with ECN

Explicit Congestion Notification (RFC 3168)

1. IP router predicts congestion — RED with mark (no drop)
2. IP router indicates congestion to receiver in IP header
3. Receiver indicates congestion to sender in TCP ACK header



Explicit Congestion Notification (ECN)

IP datagram

4 bits	4 bits	6 bits	2 bits	16 bits
Version	Hlen	**DSCP**	**ECN**	Total Length (header + data in bytes)
Identification			Flags	Fragment Offset (13 bits)
Time to Live		Protocol		Header Checksum
Source IP Address				
Destination IP Address				
Options				
Data				

Differentiated Services Code Point (DSCP)

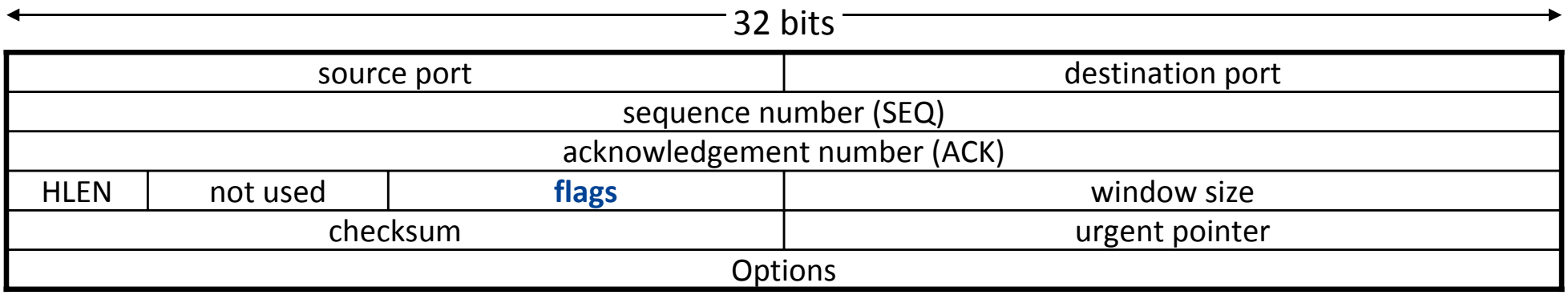
QoS requirements

Explicit Congestion Notification (ECN)

00	Not ECN capable	For retransmissions
01	ECT(0) – ECN Capable Transport (0)	To allow protocol error checking
10	ECT(1) – ECN Capable Transport (1)	
11	CE (Congestion Experienced)	

Explicit Congestion Notification (ECN)

TCP header flags



NS	ECN-nonce concealment protection
CWR	Congestion Window Reduced (CWR) flag
ECE	ECN-Echo

URG	Urgent pointer
ACK	**Acknowledgment**
PSH	Push buffer
RST	**Reset**
SYN	**Synchronize**
FIN	**No more data**

ECN Negotiation

TCP client

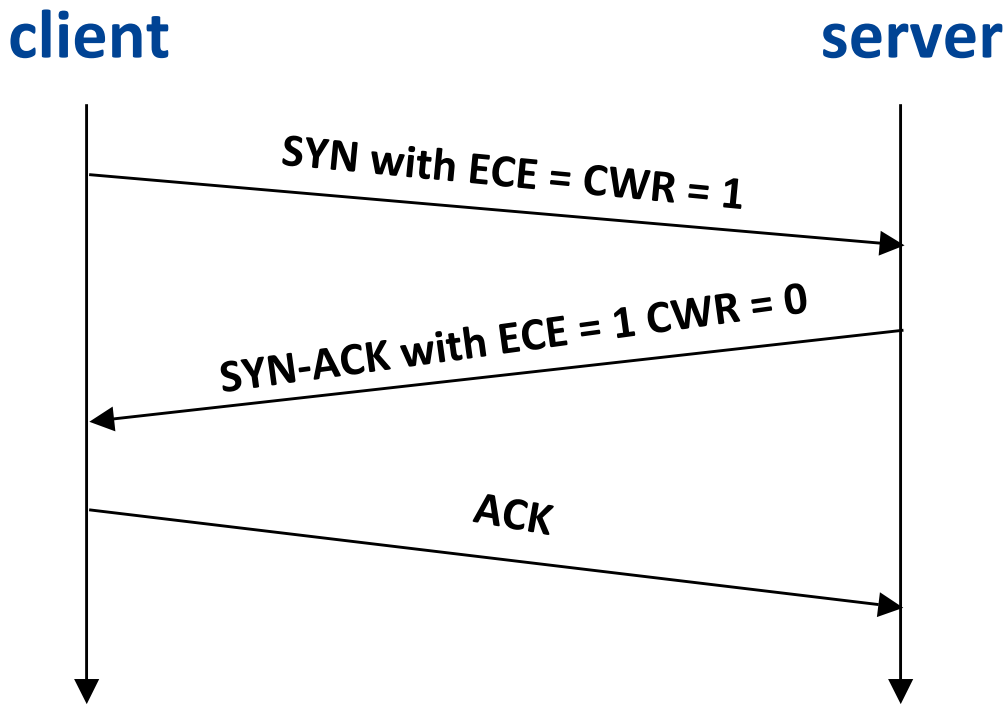
SYN ECE = CWR = 1 in SYN

TCP server

ECE = 1 in SYN-ACK

IP

ECT(0) , ECT(1) in SYN and SYN-ACK

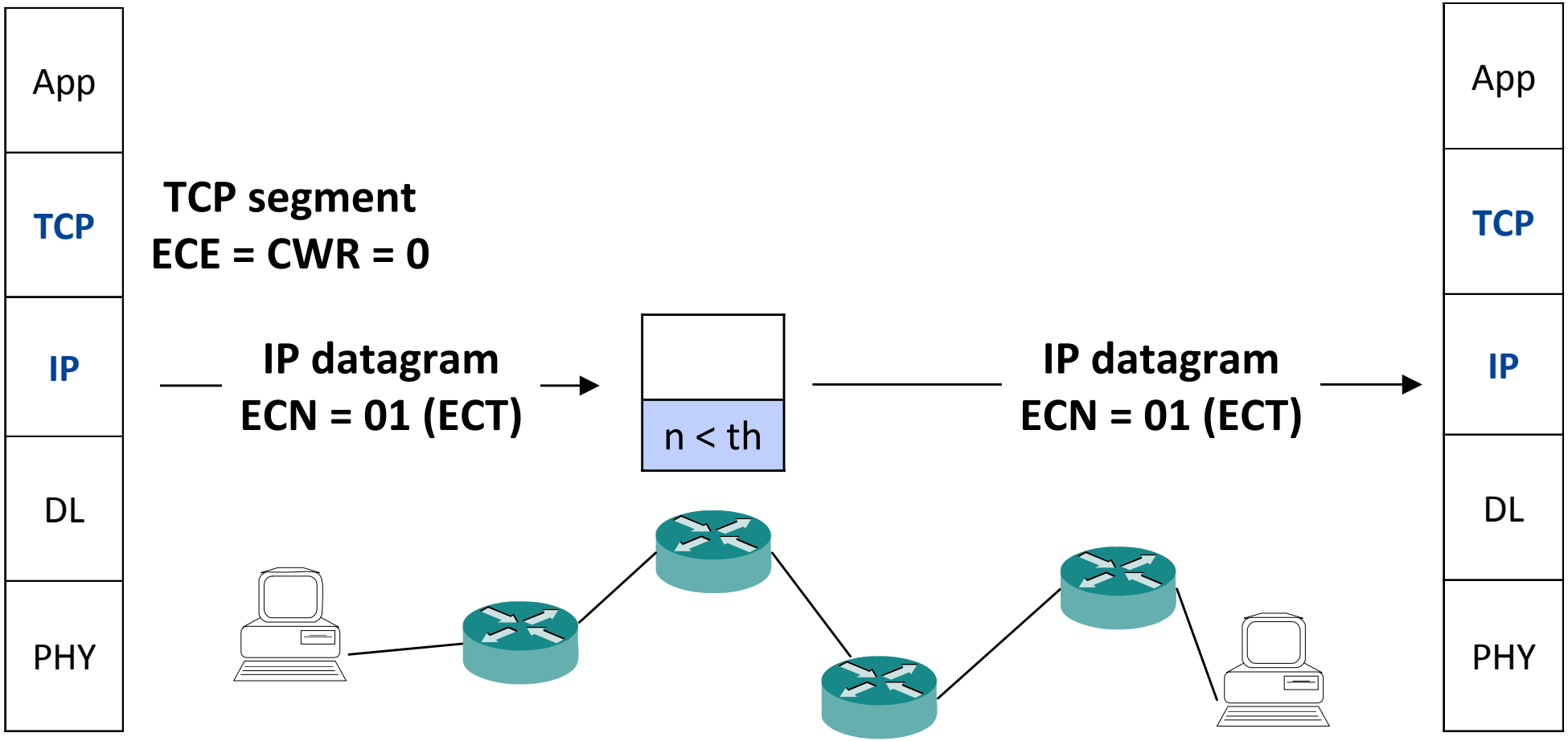


ECN Operation — 1

No congestion

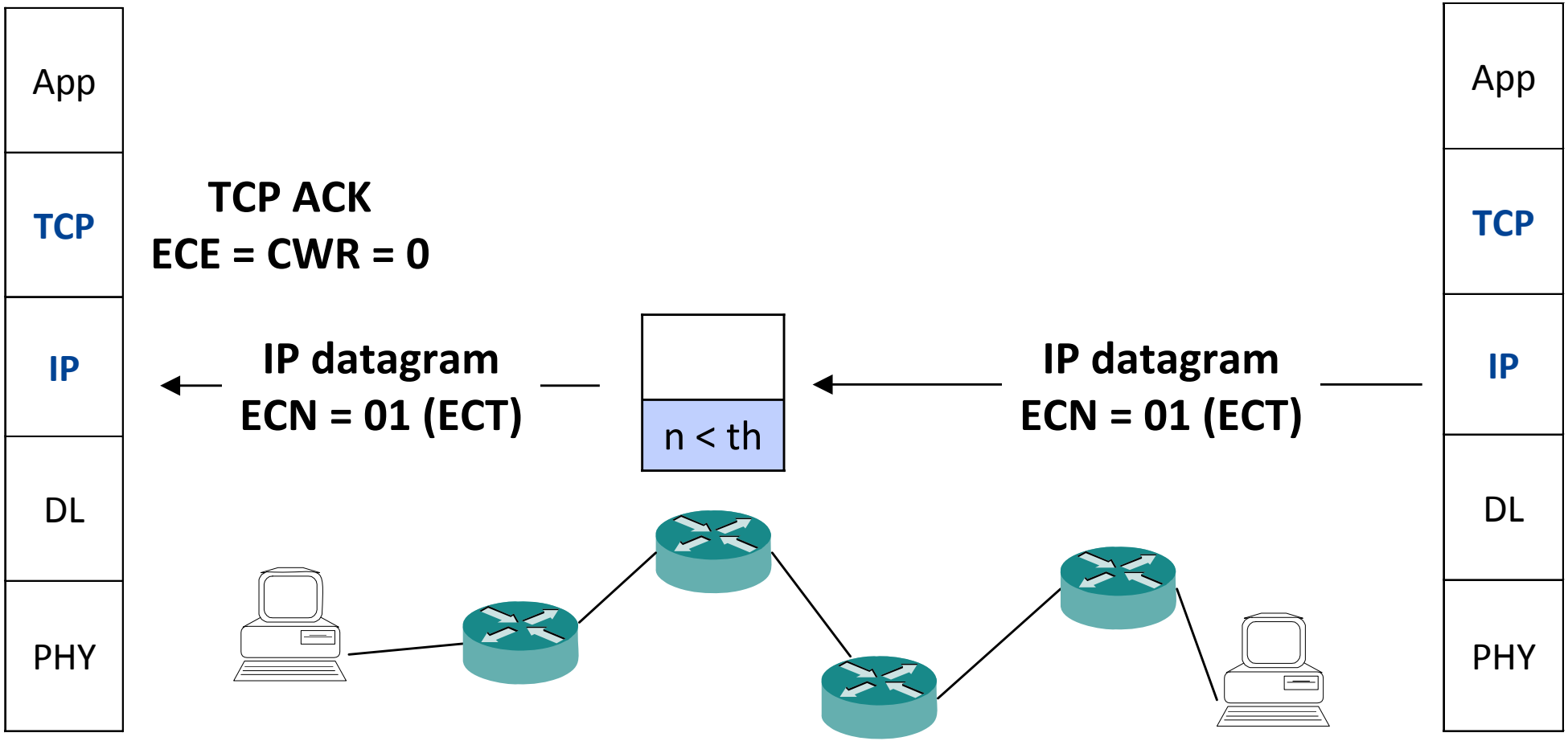
Measure long term average buffer level **n**

Compare with threshold level **th**



ECN Operation — 2

No congestion



ECN Operation — 3

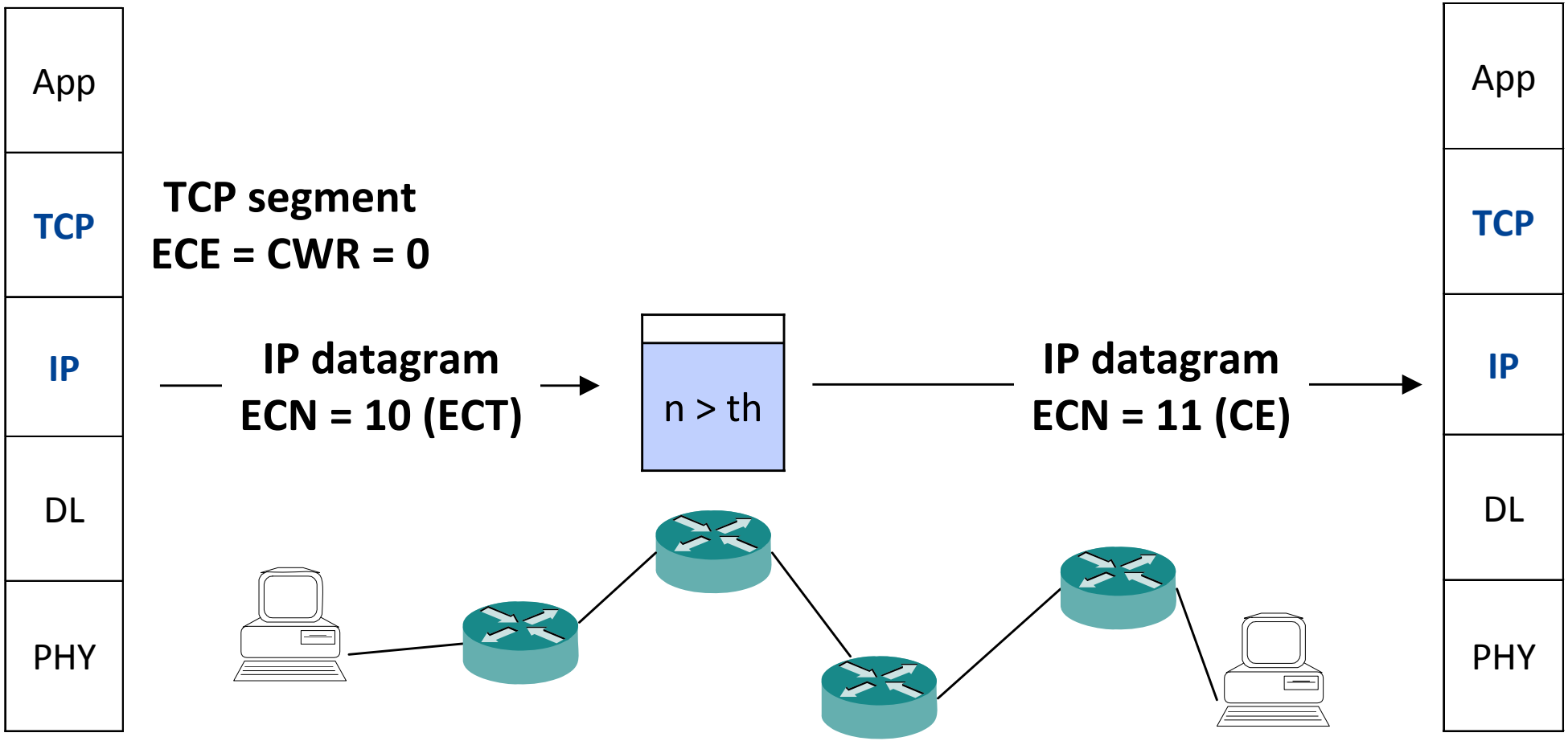
Incipient congestion

Router

Sees ECN = ECT in incoming IP header

Sets ECN = CE in outgoing IP header

Notifies receiver of incoming congestion



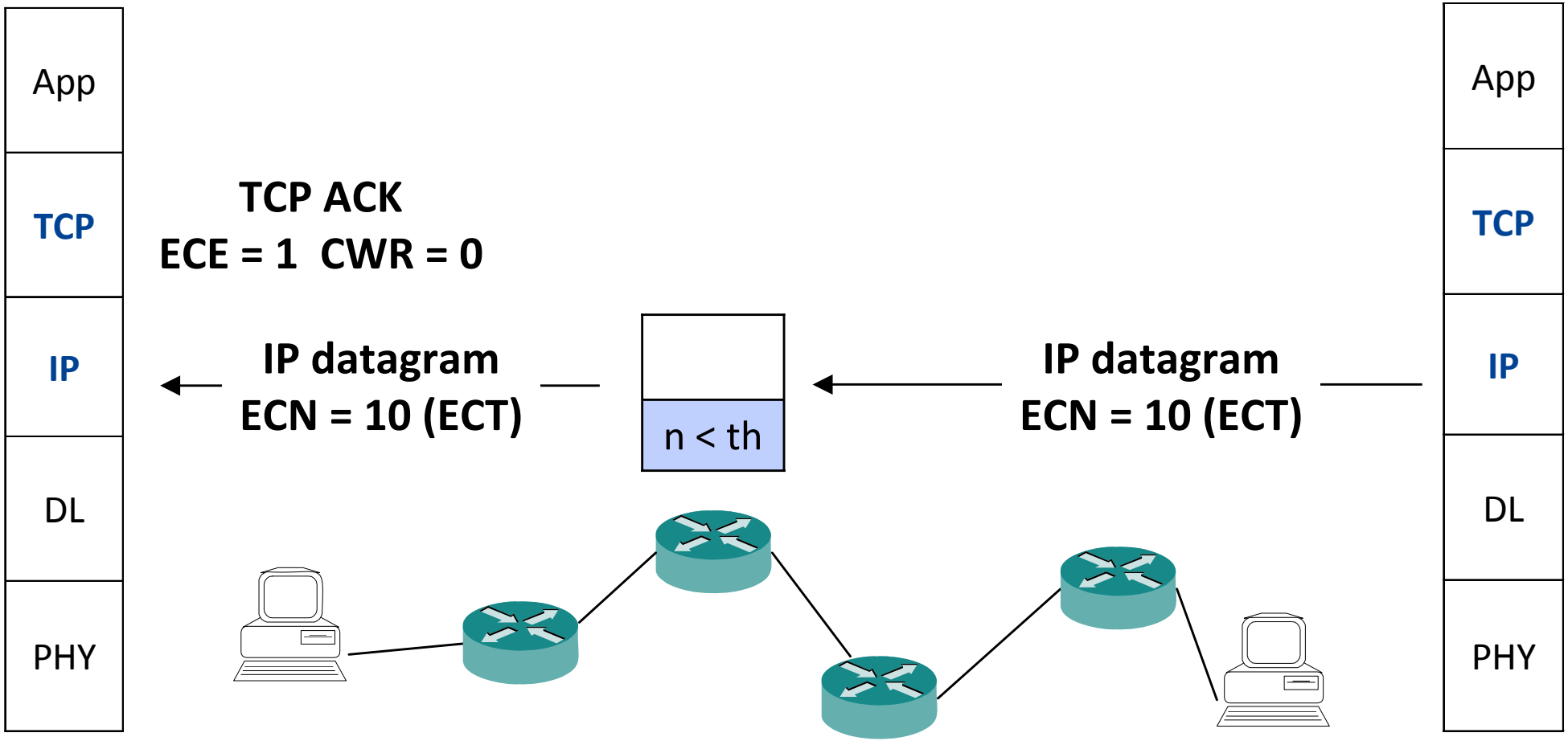
ECN Operation — 4

Incipient congestion

Receiver

Sets ECE = 1 in TCP header

Notifies sender of congestion



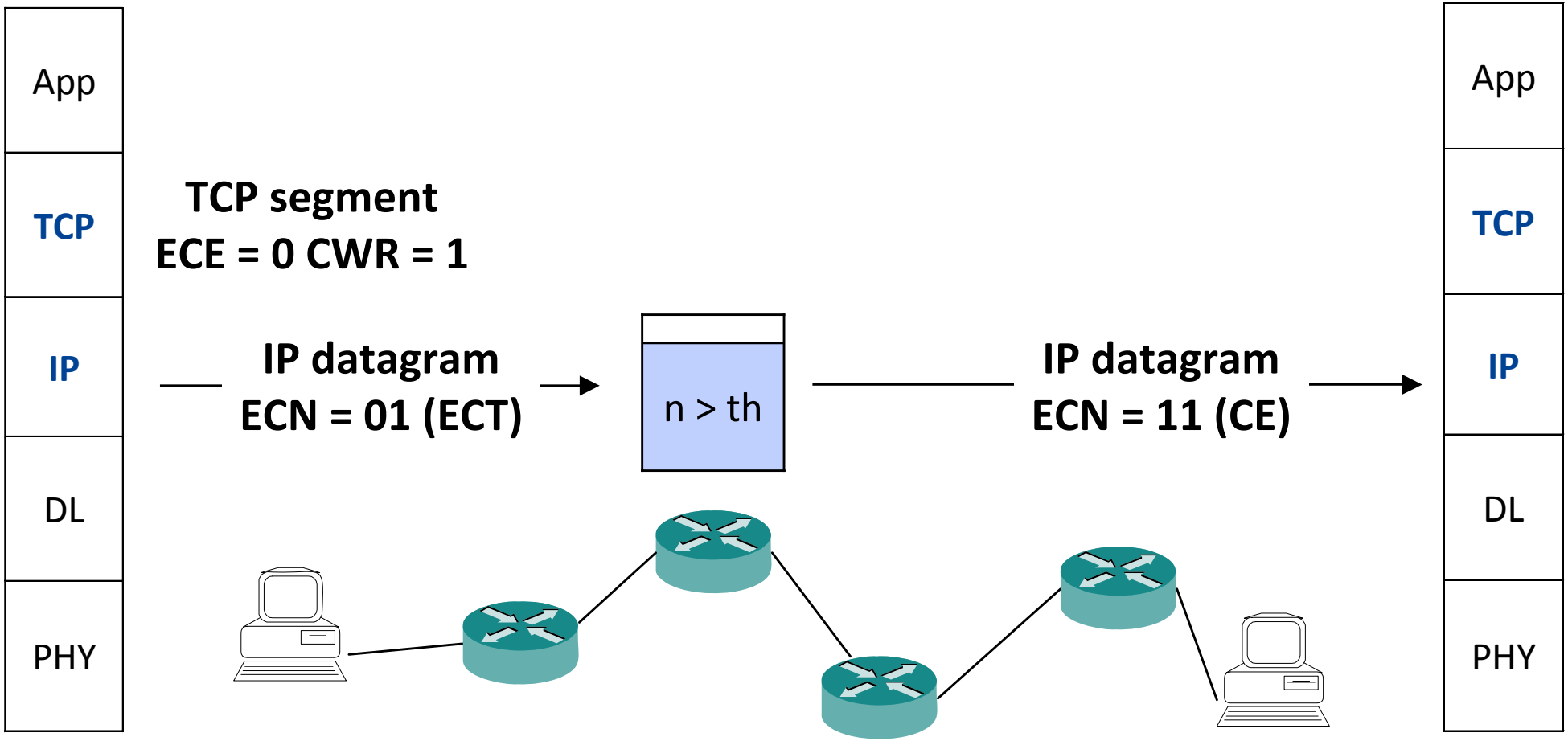
ECN Operation — 5

Incipient congestion

Sender

Lowers congestion window (once per RTT)

Sets CWR = 1 in TCP header (ACK of ECE to receiver)



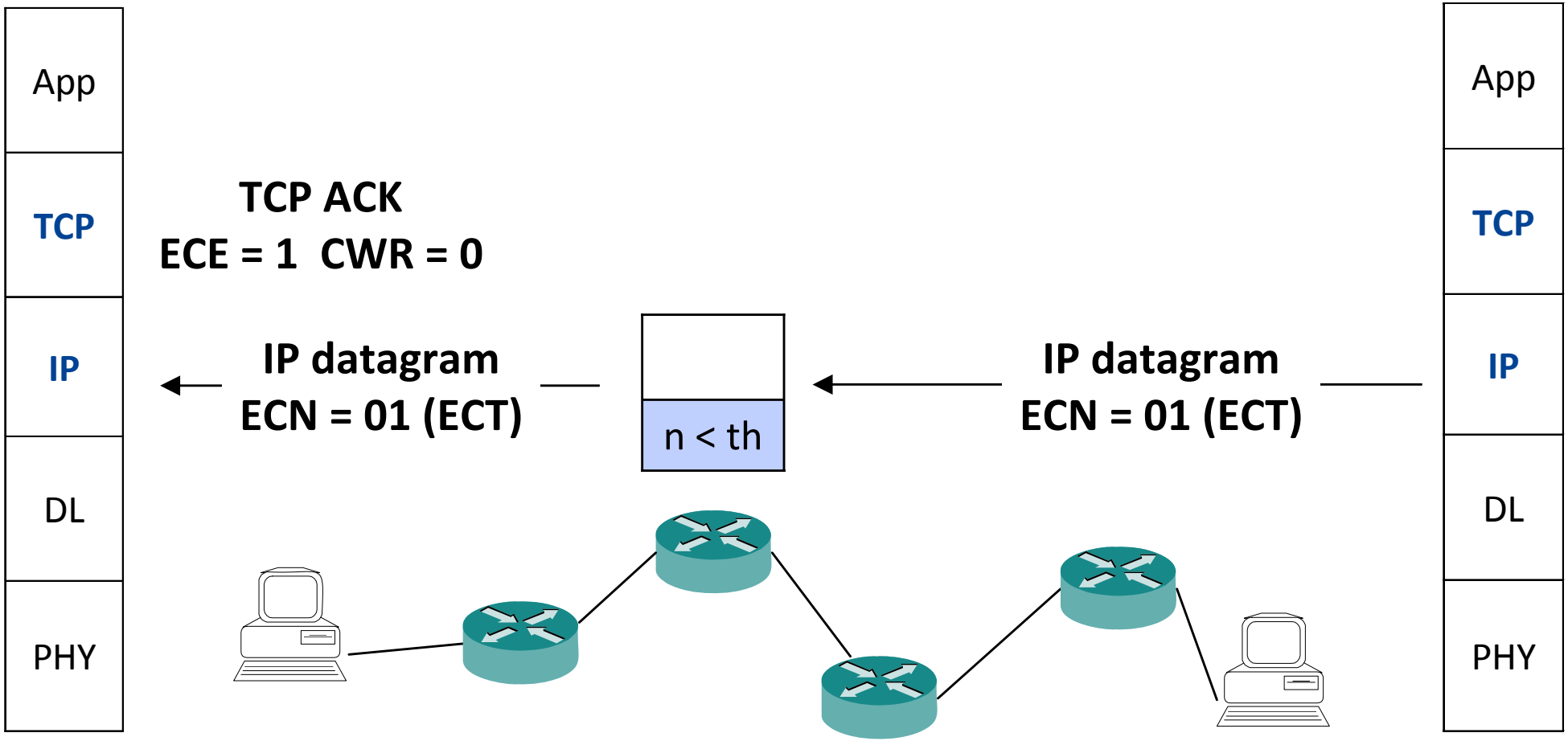
ECN Operation — 6

Incipient congestion

Receiver

Sees CWR = 1 in sender TCP header

CE in IP header new incoming ⇒ ECE = 1 in ACK TCP header



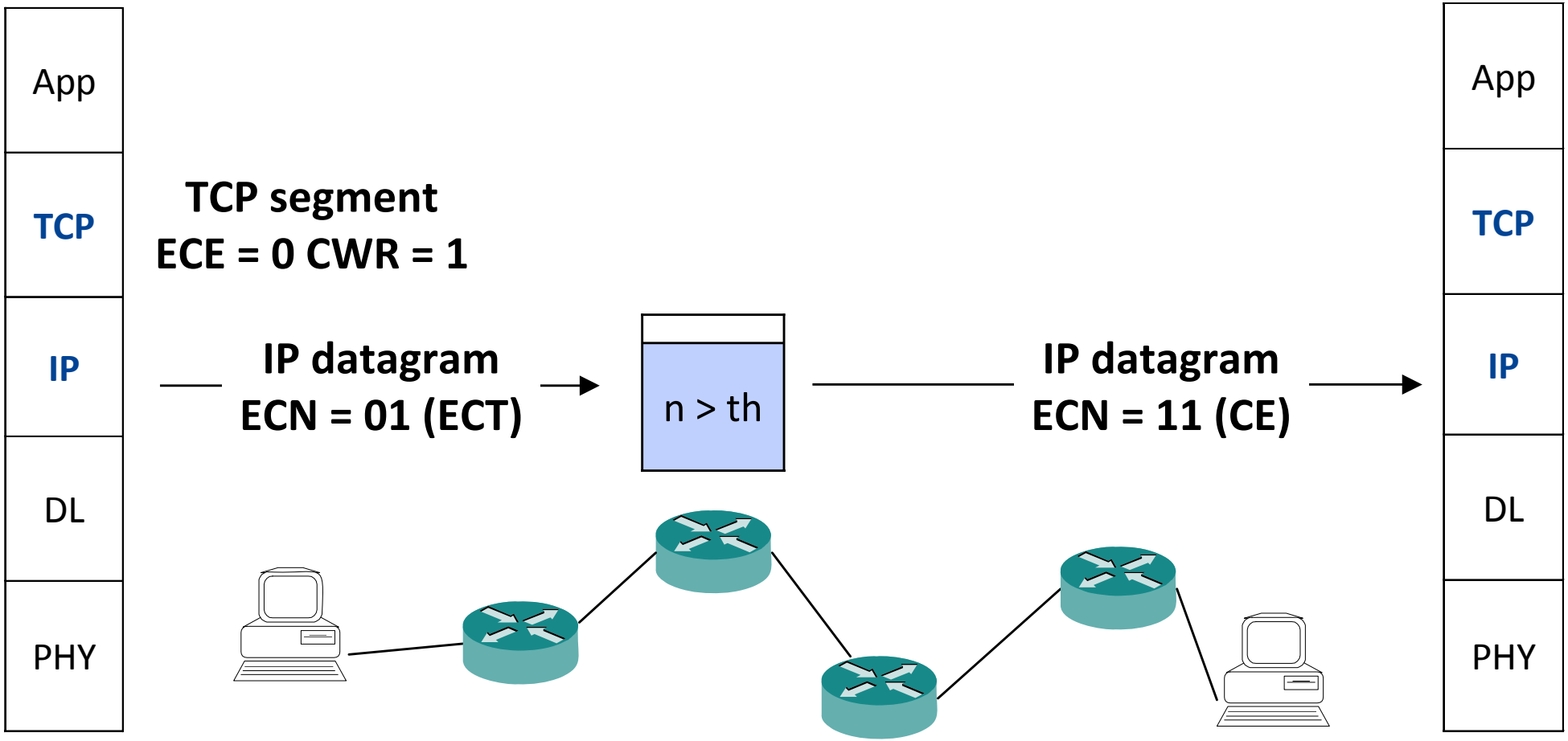
ECN Operation — 7

Continued congestion

Sender

Lowers congestion window once per RTT

Sets CWR = 1 in TCP header (ACK of ECE)



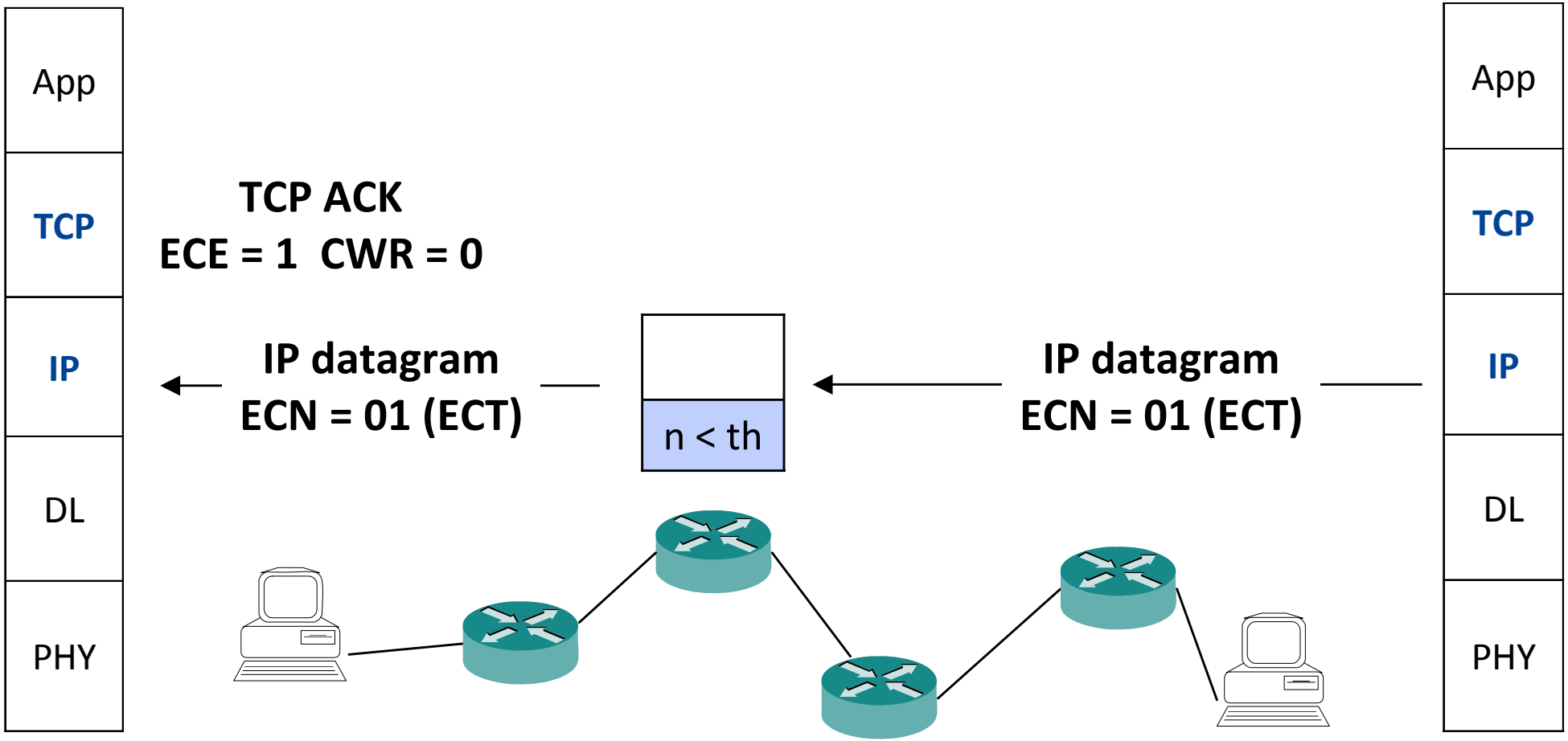
ECN Operation — 8

Continued congestion

Receiver

Sees CWR = 1 in sender TCP header

CE in IP header new incoming ⇒ ECE = 1 in ACK TCP header

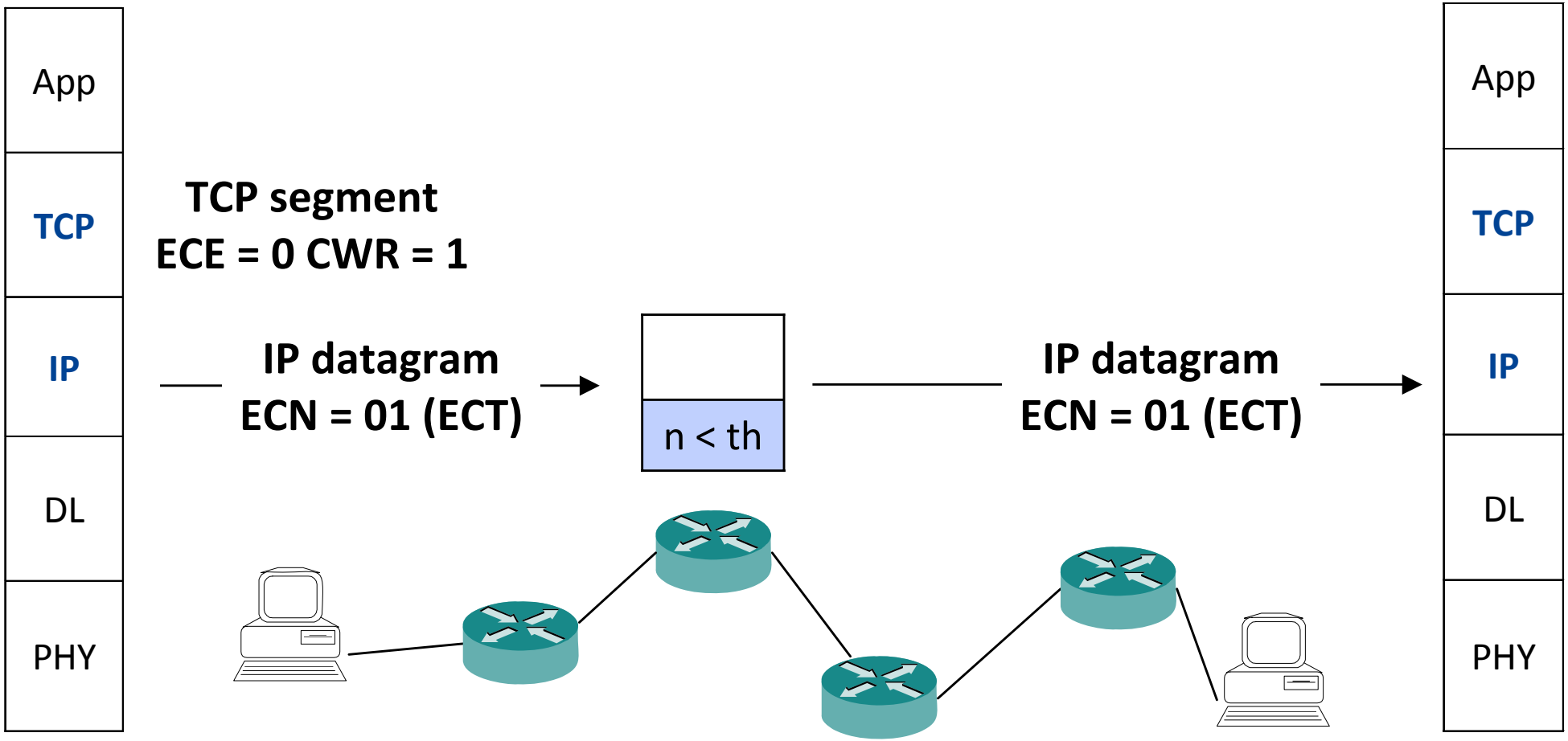


ECN Operation — 9

End of congestion

Sender sets CWR = 1 in TCP header (ACK of ECE)

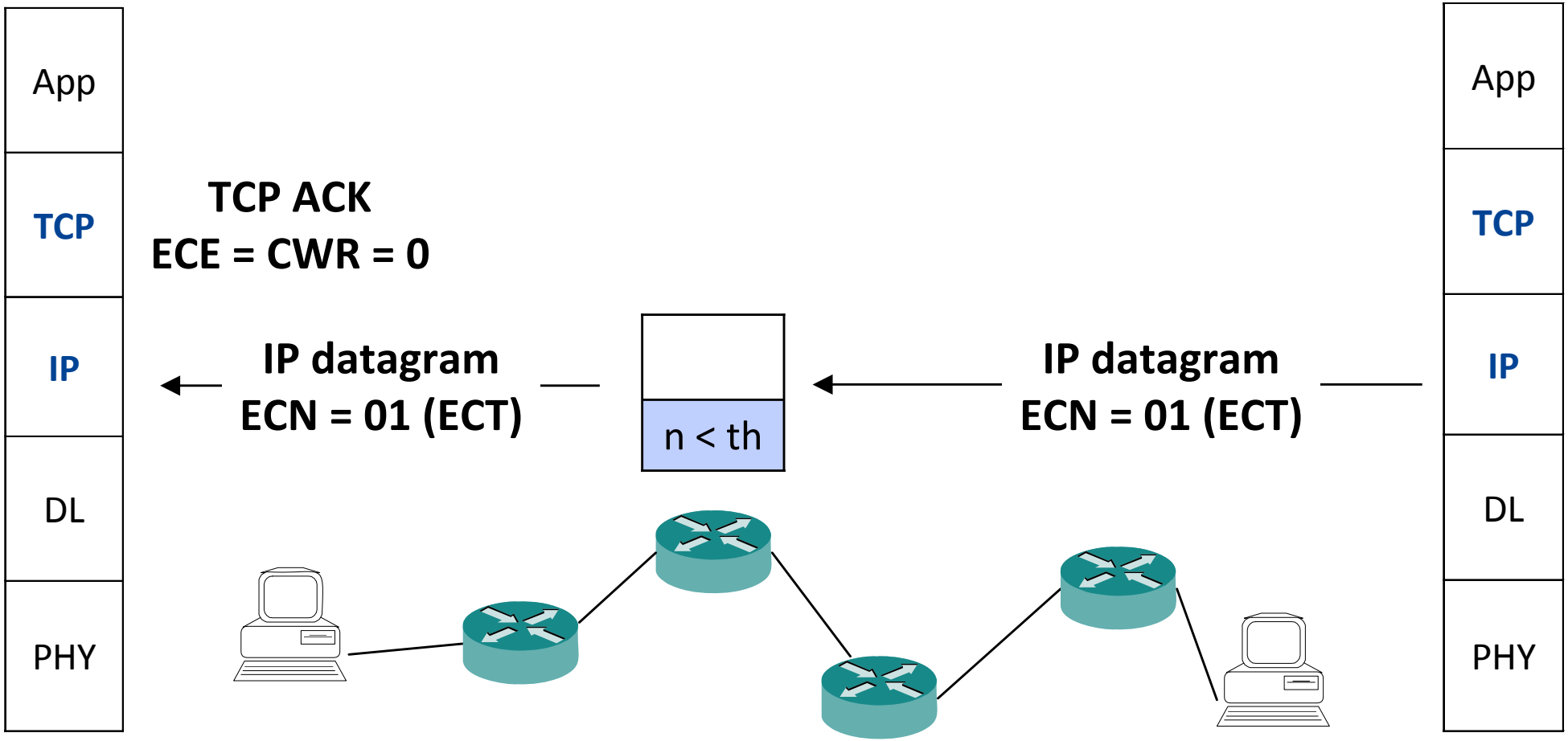
Router sends ECN = 01 in IP header (signals no congestion)



ECN Operation — 10

End of congestion

Receiver sends ECE = 0 in TCP header (signals no congestion)

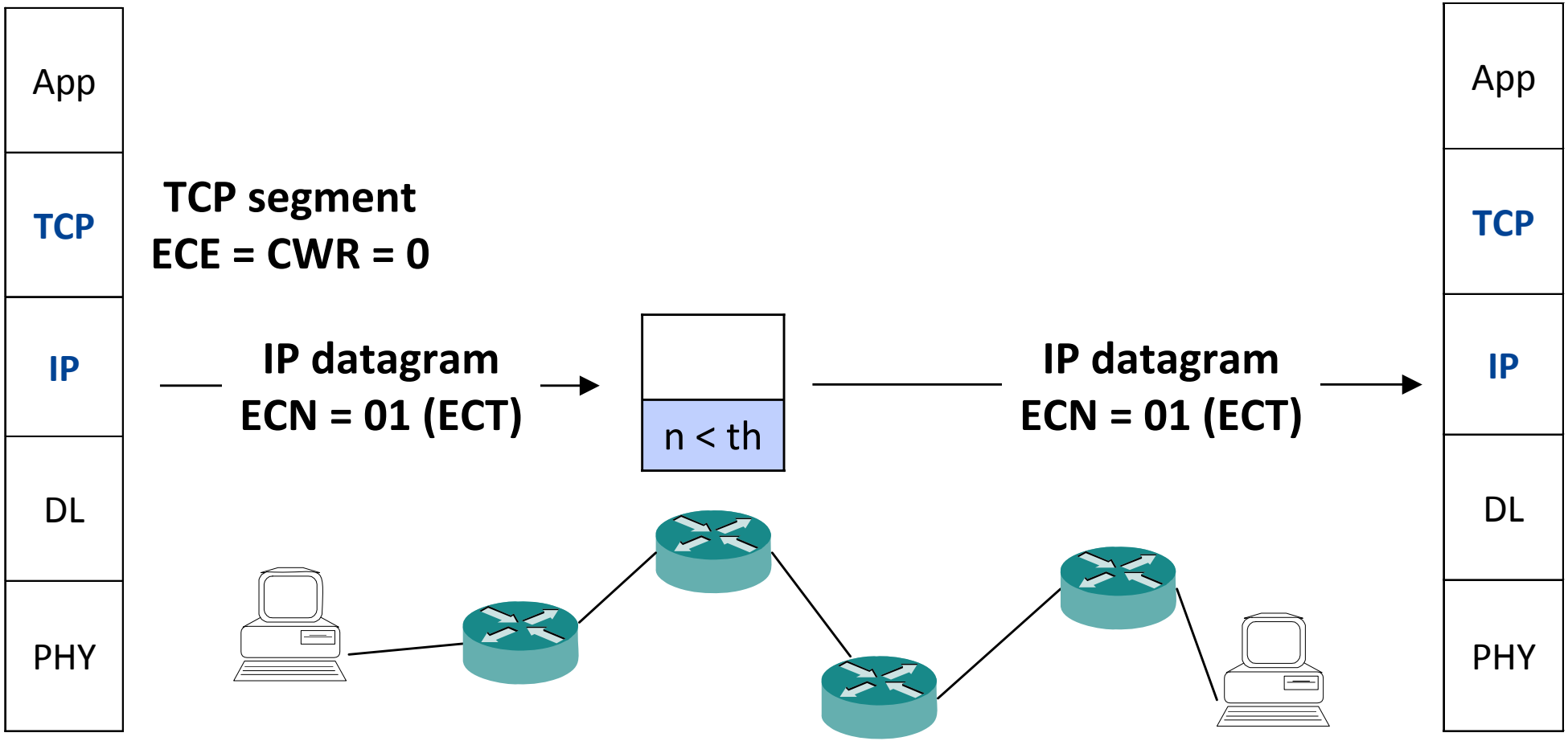


ECN Operation — 11

End of congestion

Sender clears CWR and begins raising congestion window

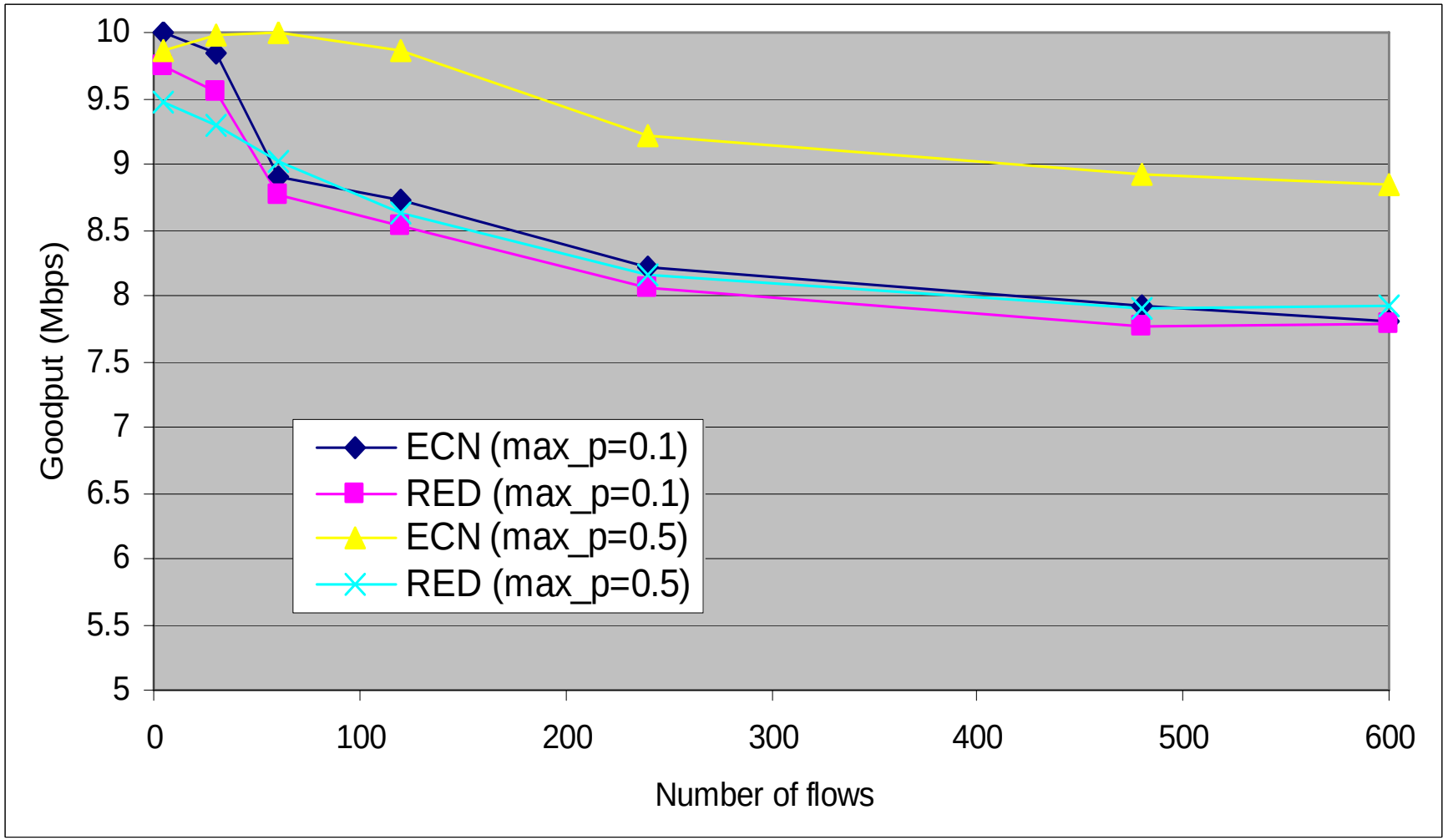
Router sends ECN = 01 in IP header



RED and ECN Goodput

Parameters

$min_{th} = 5 \quad max_{th} = 30$

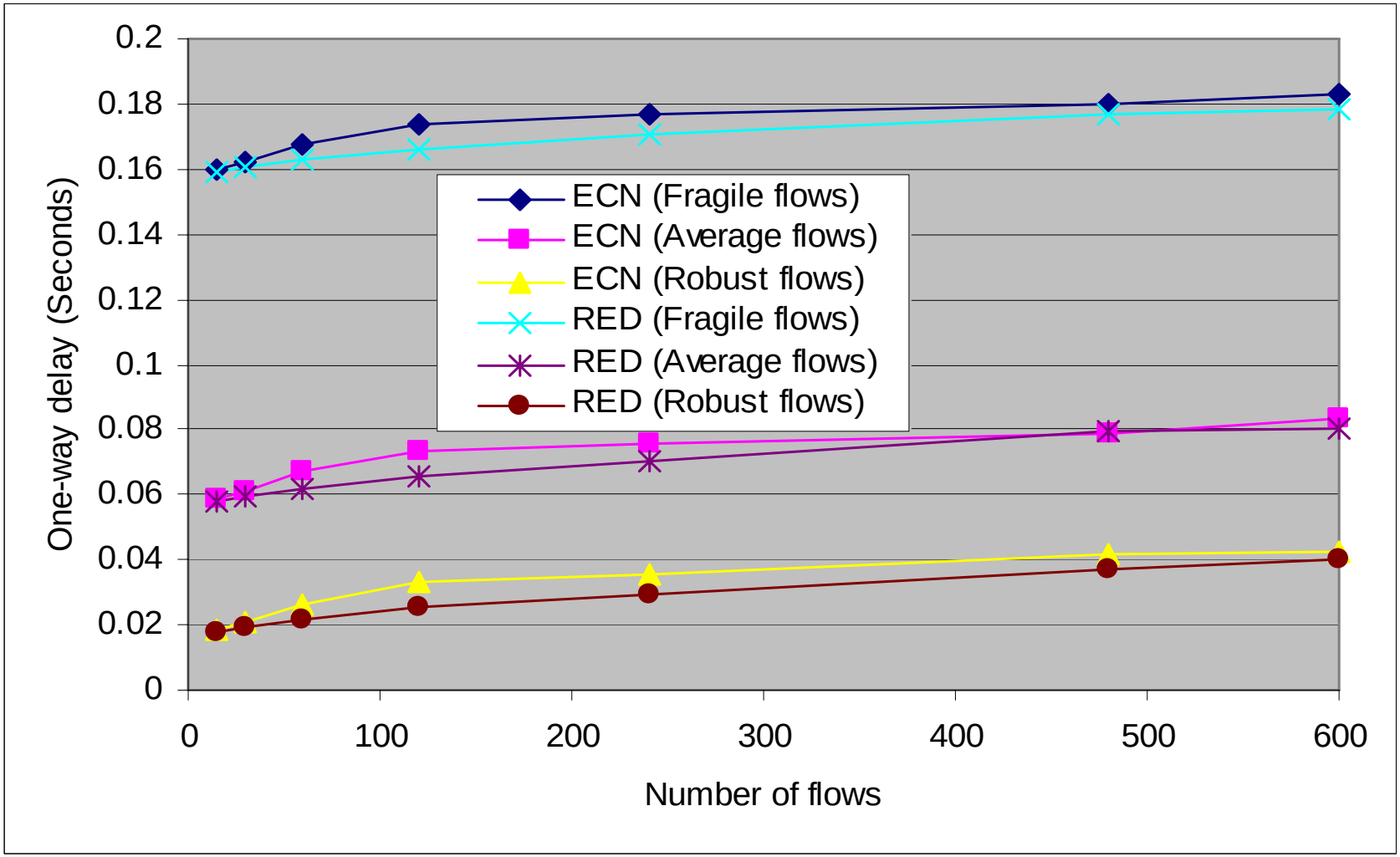


Ref: Kinicki and Zheng, A Performance Study of Explicit Congestion Notification (ECN) with Heterogeneous TCP Flows

RED and ECN Delay

Parameters

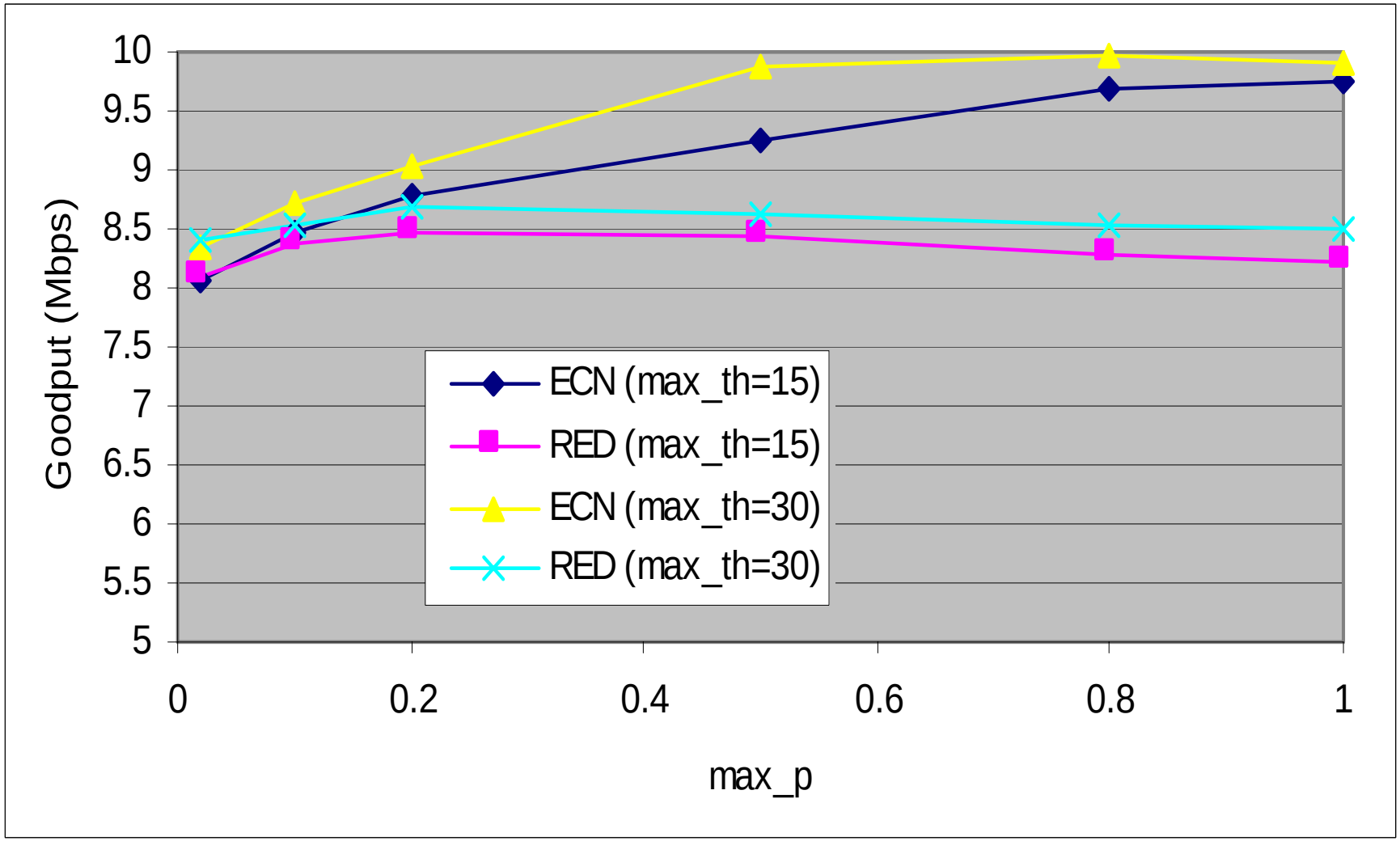
$min_{th} = 5$ $max_{th} = 30$ $max_p = 0.5$



Goodput with 120 flows

Parameters

$min_{th} = 5$ $max_{th} = 30$



ECN Nonce (RFC 3540)

Problem

Unscrupulous or poorly implemented receiver

Clears ECN-Echo — no congestion signals to sender

Gives receiver advantage over connections that behave properly

Sender

IP header with **`ECN = 01 = ECT(0)`** or **`ECN = 10 = ECT(1)`**

Except retransmissions (**`Not ECN Capable`**) and **`CE`** packets

Keeps per-packet map of **`SEQ`** to nonce (`0` or `1`)

Router

Forwards packet or overwrites ECT with **`ECN = 11 = CE`**

Receiver

Keeps cumulative **`ACK`** number (standard TCP)

Keeps cumulative **`sum % 2`** of received nonces for ACKed packets

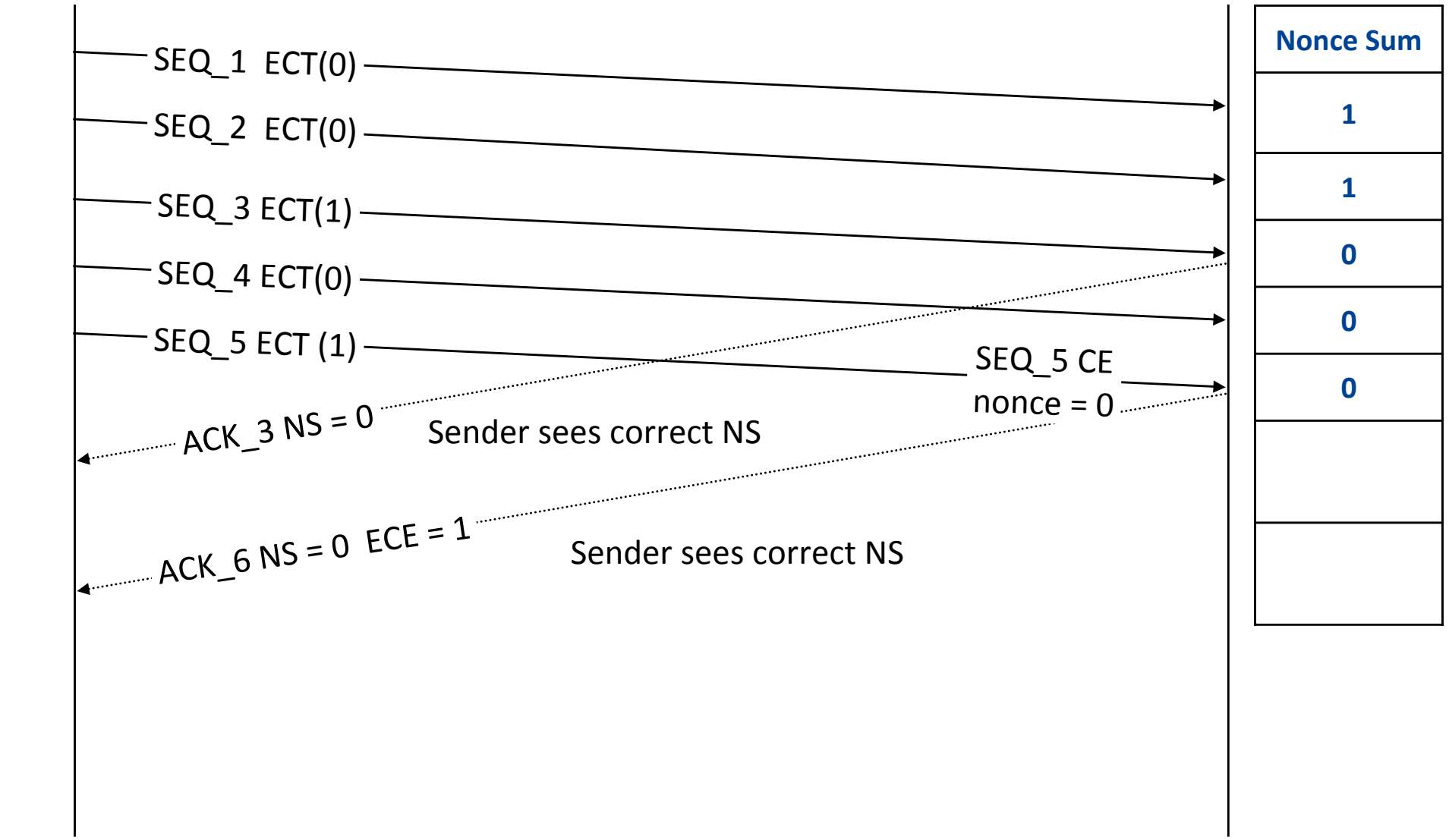
`NS` flag in TCP header = sum of nonces for ACKed packets

`CE` packets — use nonce = 0

Nonce Example

Honest Receiver

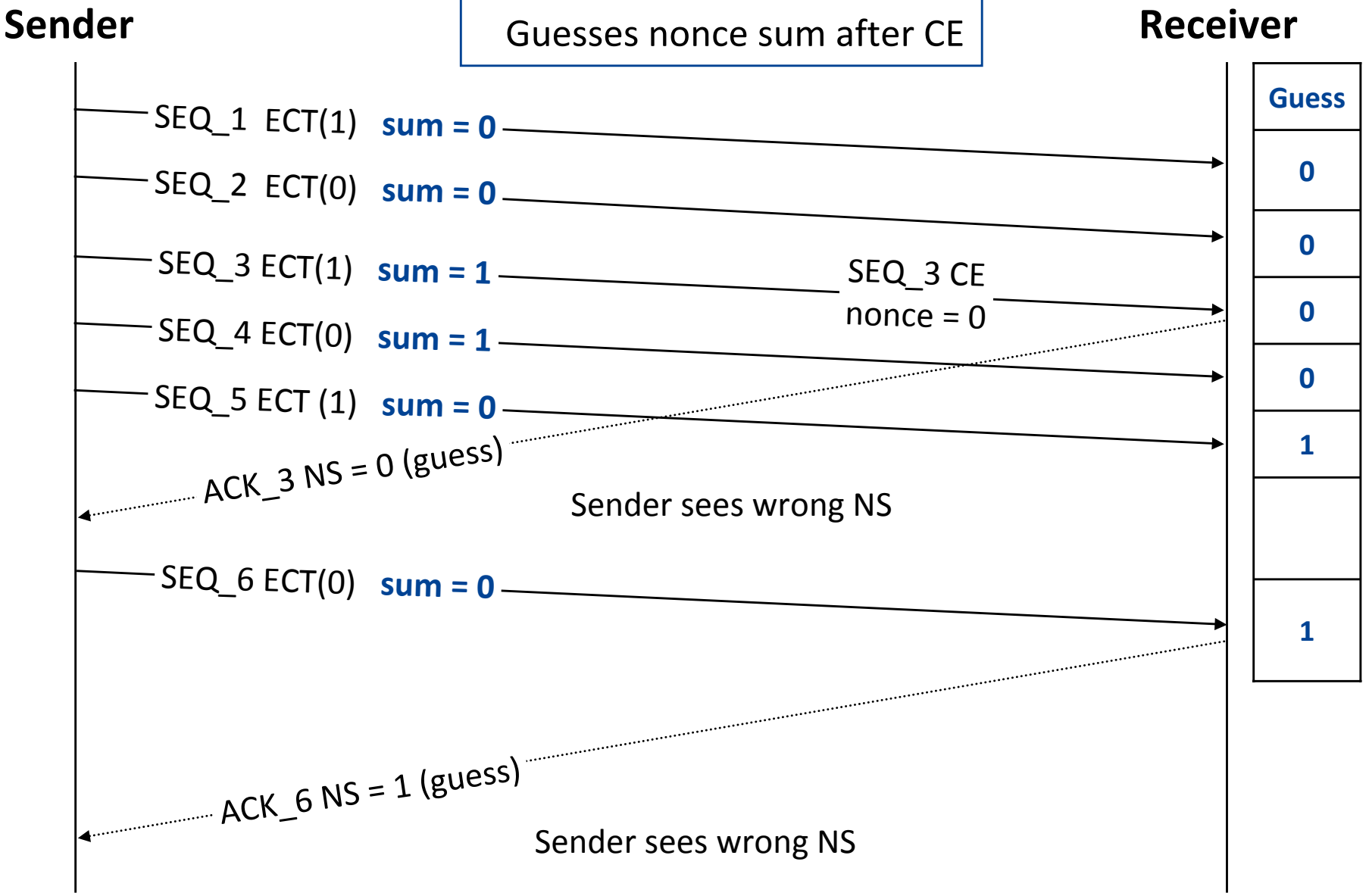
NS initialized to 1
Sent in SYN-ACK and ACK of handshake



Nonce Example

Cheating Receiver

Receiver ignores CE
- Does not set ECE
- Guesses nonce sum after CE



TCP
over
Wireless

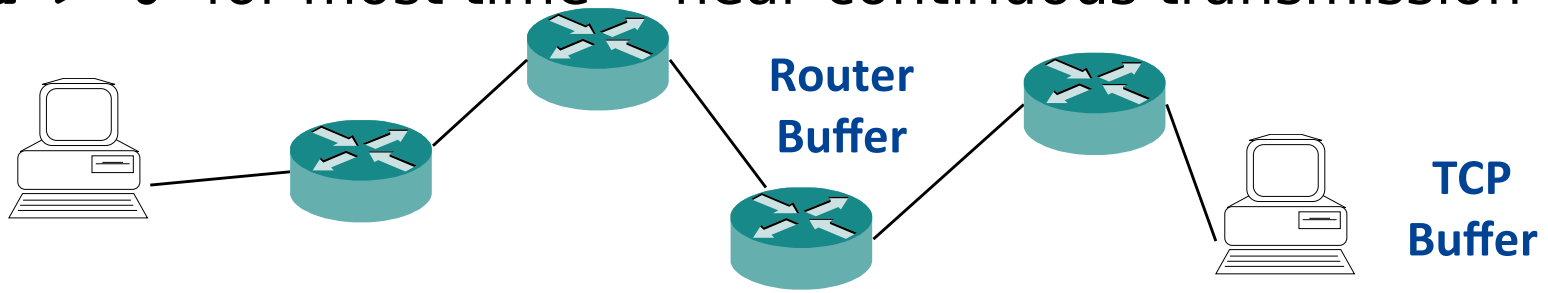
Classic TCP-Reno

Ideal operation

`in-flight` segments = **`cwnd`** (send **`cwnd`** without stopping)

Cumulative **`ACK`** for each 2 or 3 segments

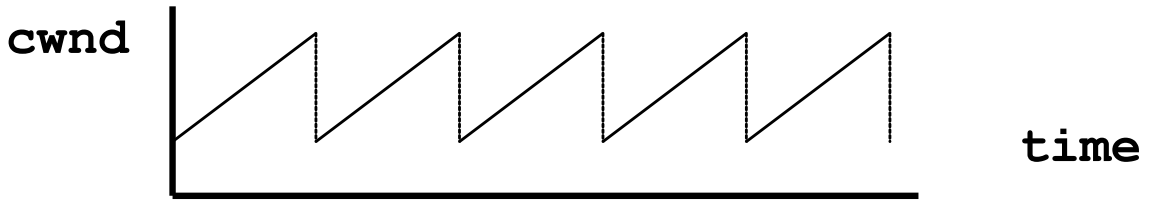
`cwnd > 0` for most time ⇒ near-continuous transmission



Finite buffers in routers

Collision avoidance

`cwnd ← cwnd + 1` until buffers fill ⇒ **`cwnd ← cwnd / 2`**



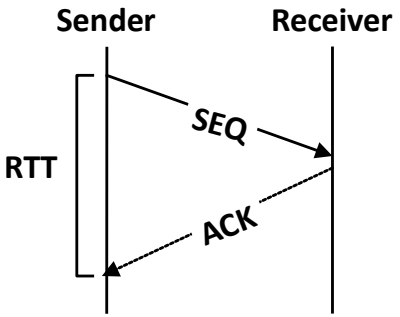
RTT and Buffer Errors

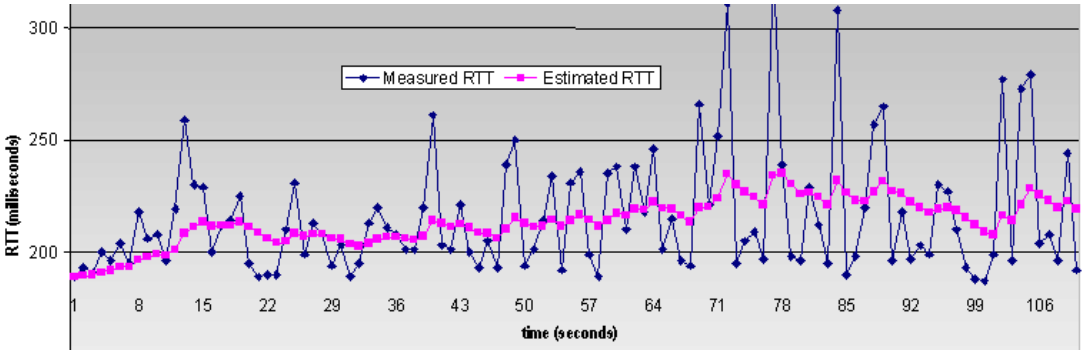
Round Trip Time

RTT = data transmit time + send buffer times
+ ACK transmit time + ACK buffer times

Buffer time ~ typical service time × buffer level

RTT = random variable (rise / fall sharply)





Buffer time in TCP

Timeout

Buffer level ⟶ RTT > RTO (retransmit timeout)

Packet considered lost

Out-of-order packet

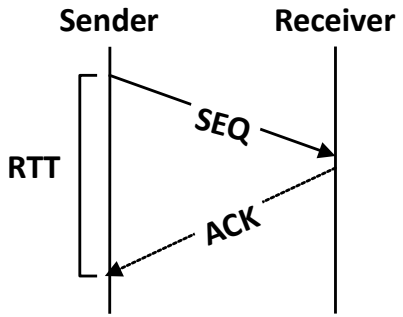
Buffer level ⟶ RTT(packet k) > RTT(packet k+1)

Receiver will send cumulative ACK if OOO packet not lost

RTT and Congestion

Buffer error condition

Buffer level $\longrightarrow$ RTT > RTO for time T_{error}

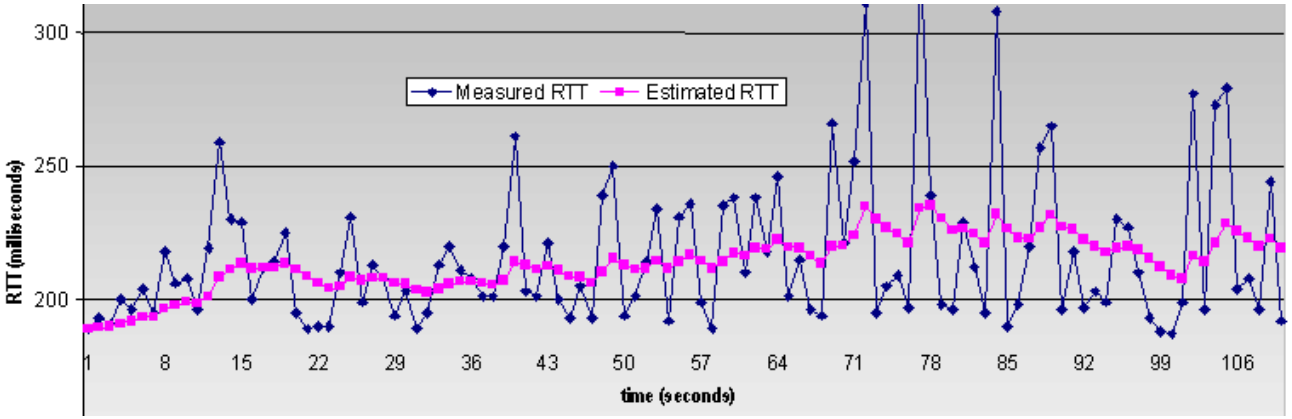


Isolated error

T_{error} < time between packets

Buffer empties before next packet

No need to lower transmission rate



Congestion

T_{error} > time between packets

Multiple buffer errors

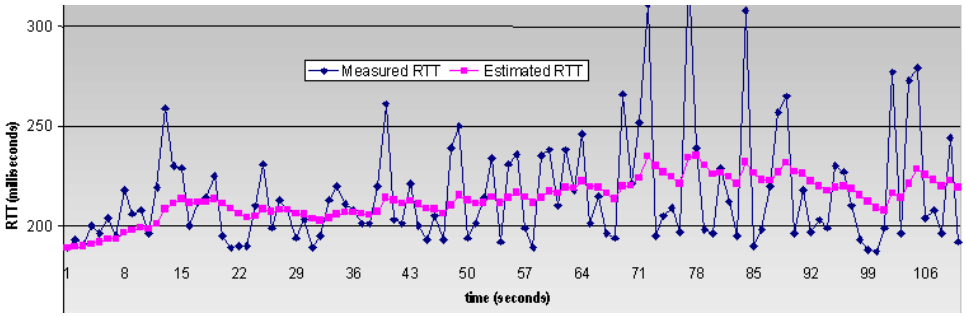
Lower transmission rate to prevent buffer errors

Congestion Control

Average transmission rate

$$R = \left\langle \frac{\text{packets sent}}{\text{time to ACK}} \right\rangle = \left\langle \frac{\text{cwnd}}{\text{RTT}} \right\rangle$$

R ↓ if cwnd ↓ or RTT ↑



Isolated buffer error

T_{error} < time between packets

RTT ↑ ⇒ R ↓ for T_{error}

Buffer corrects itself

No need to change cwnd

Congestion

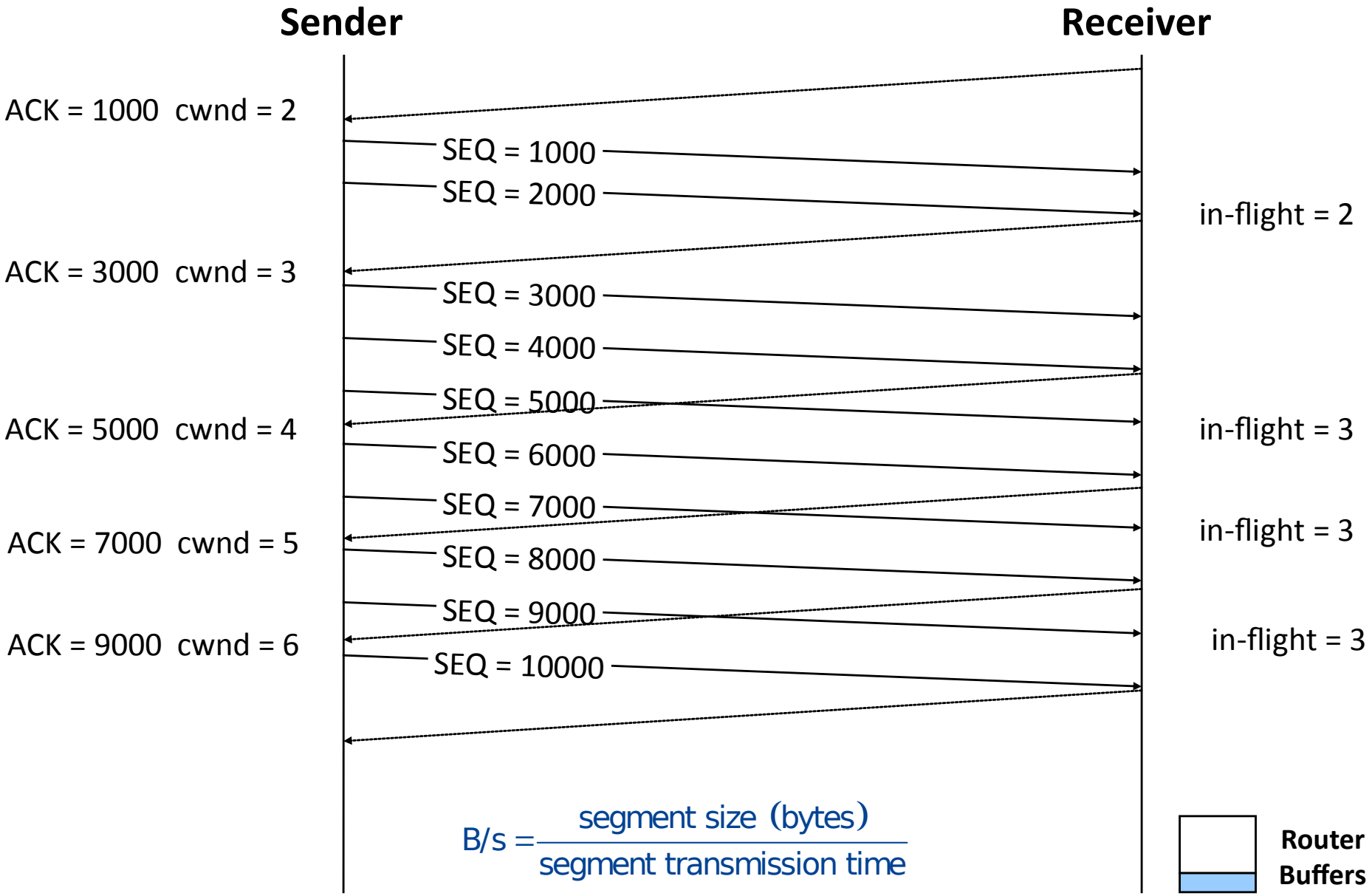
T_{error} > time between packets

RTT ↑ ⇒ R ↓

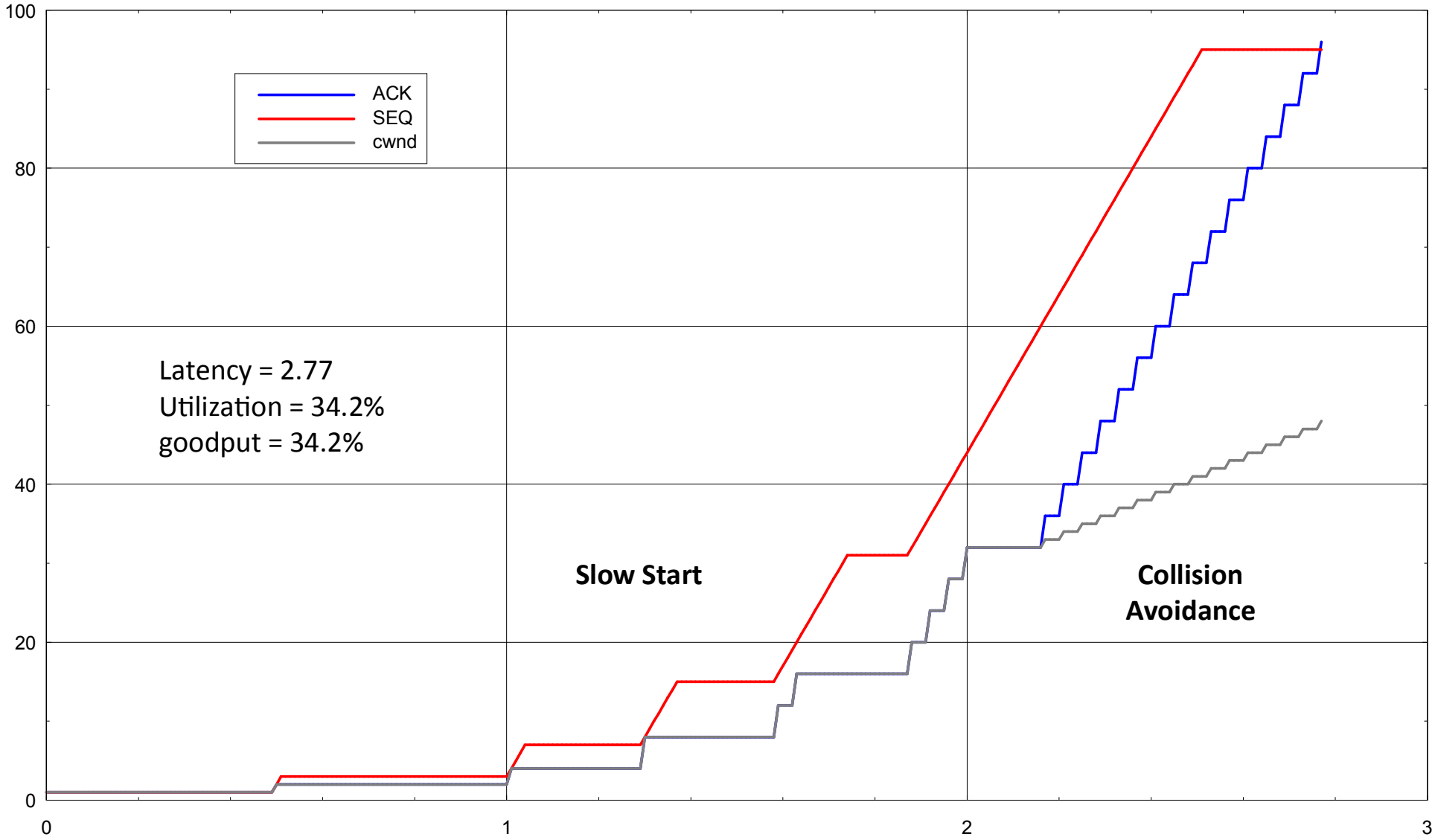
Buffer still too full

Senders: cwnd ↓ ⇒ R ↓

Classic TCP-Reno — No Congestion in Collision Avoidance

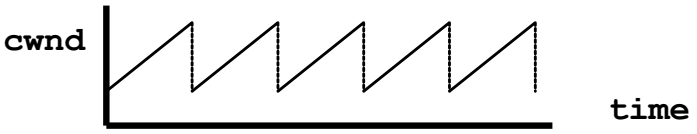


Error-Free Transmission



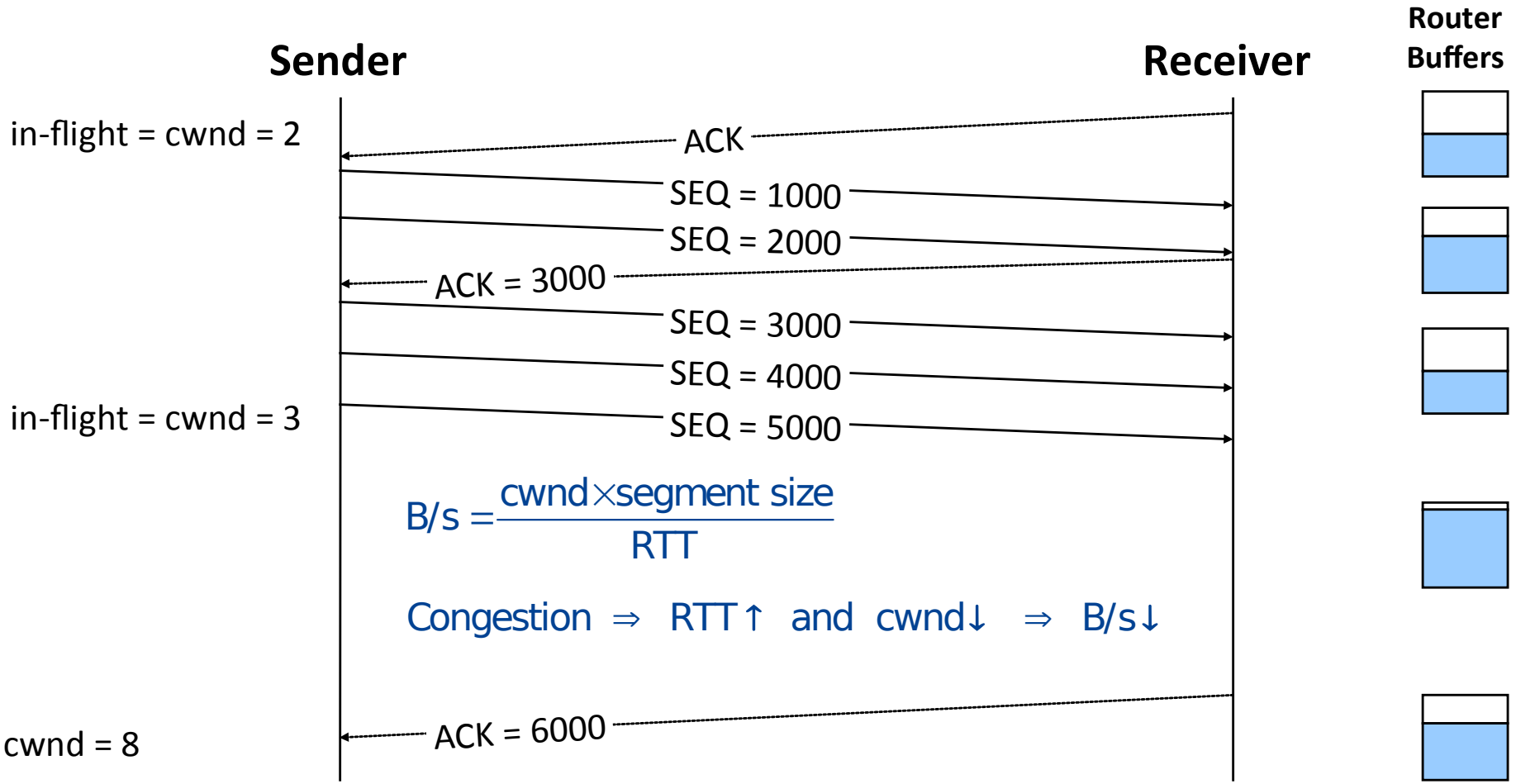
Classic TCP-Reno — Steady State in Collision Avoidance

Self-Clocking

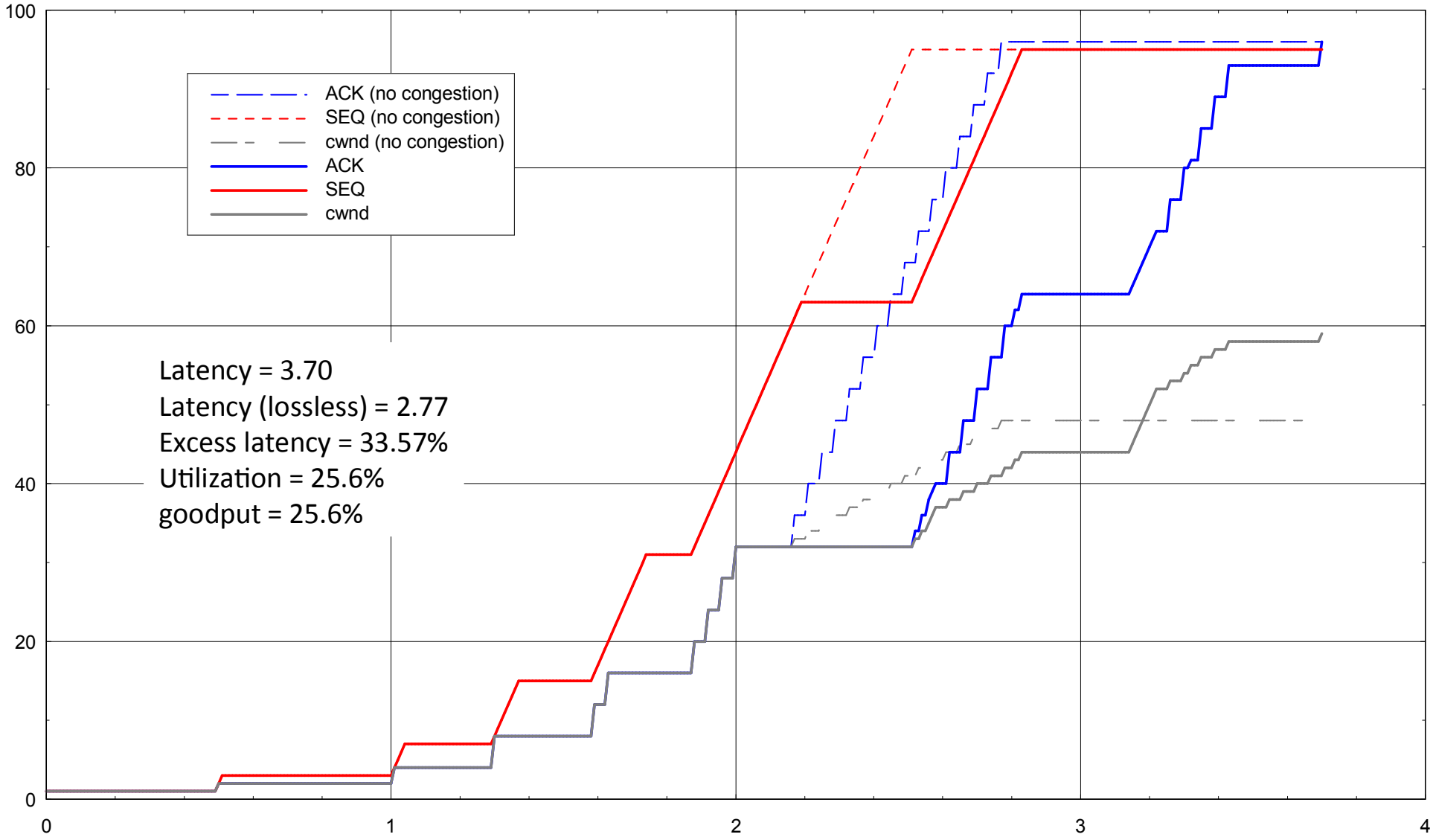


$$\texttt{in-flight} \approx \texttt{cwnd}$$

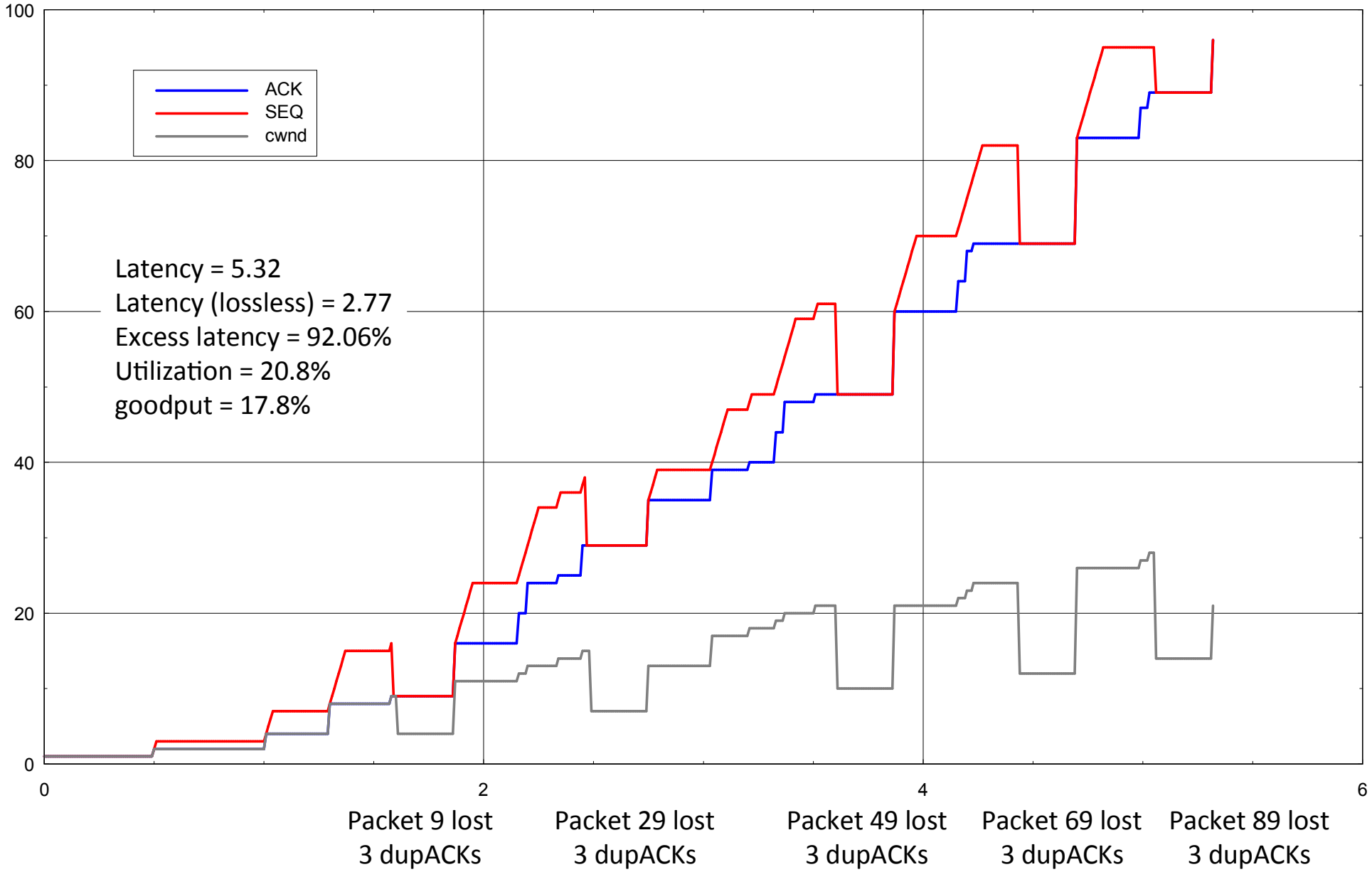
$$\texttt{n new ACKs} \Rightarrow \left\{ \begin{array}{c} \texttt{in-flight} \leftarrow \texttt{in-flight - n} \\ \texttt{cwnw} \leftarrow \texttt{cwnd + 1} \end{array} \right\} \Rightarrow \texttt{n+1 new segments}$$



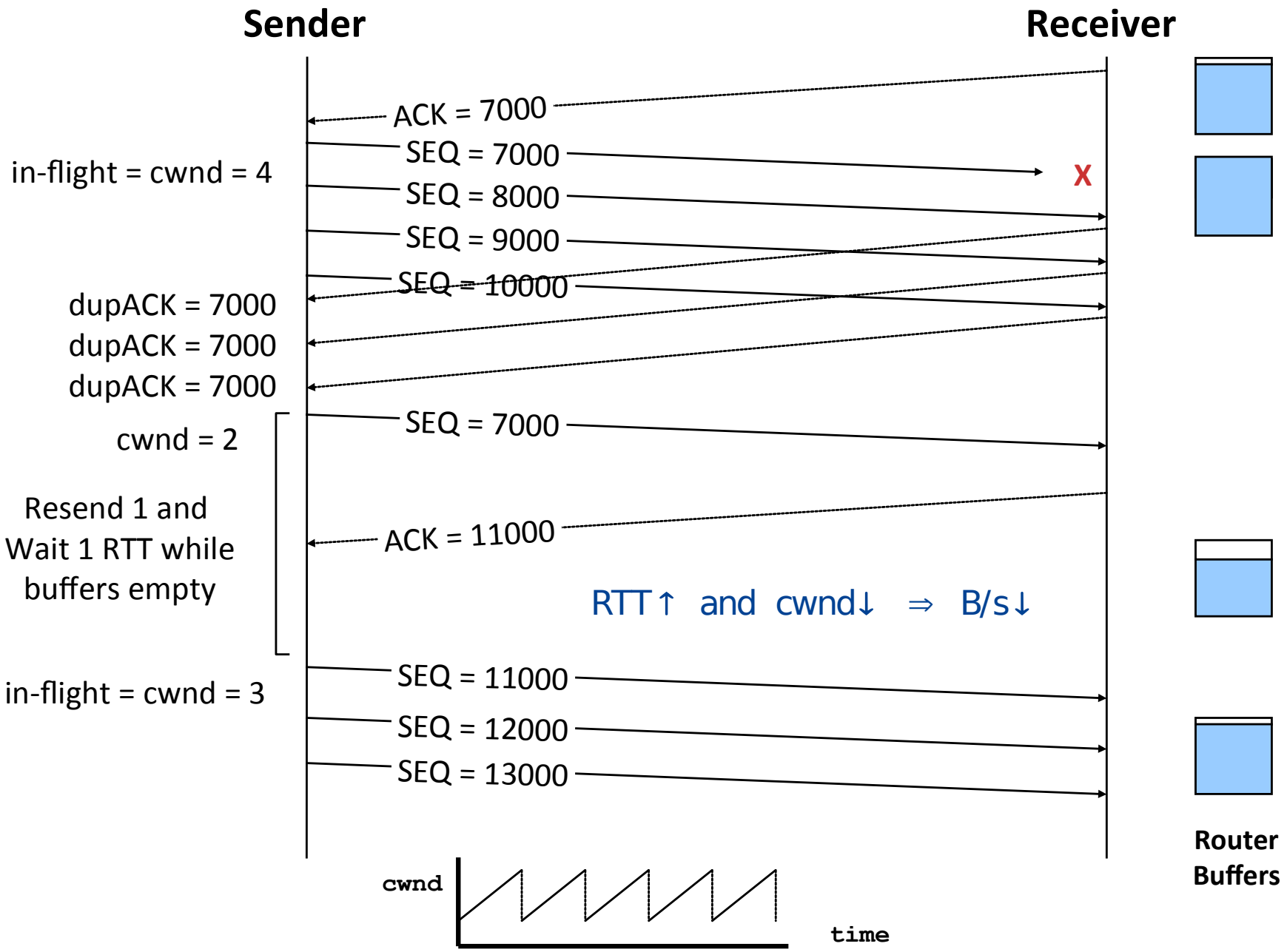
Congestion —→ Longer RTT



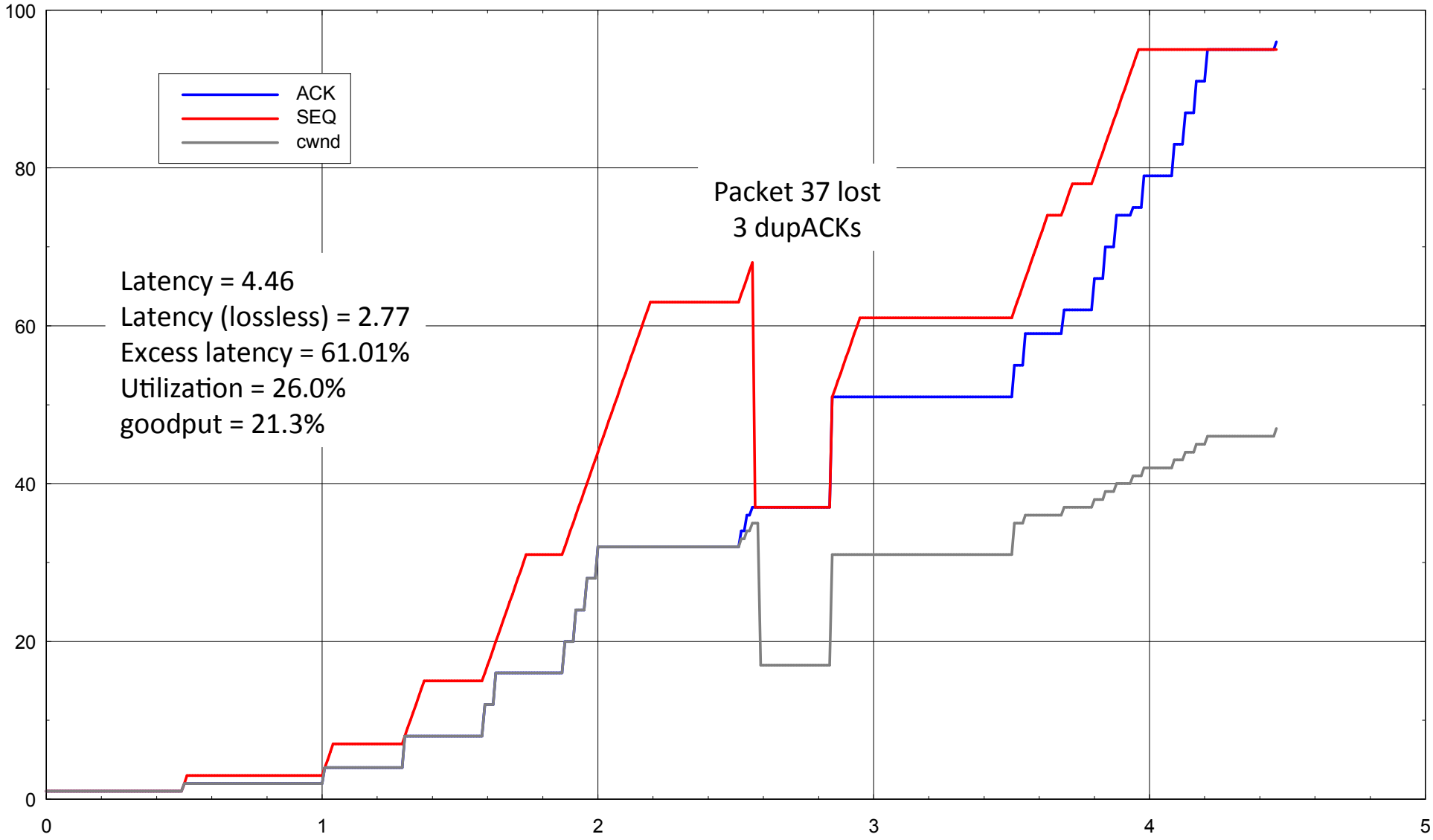
Steady State



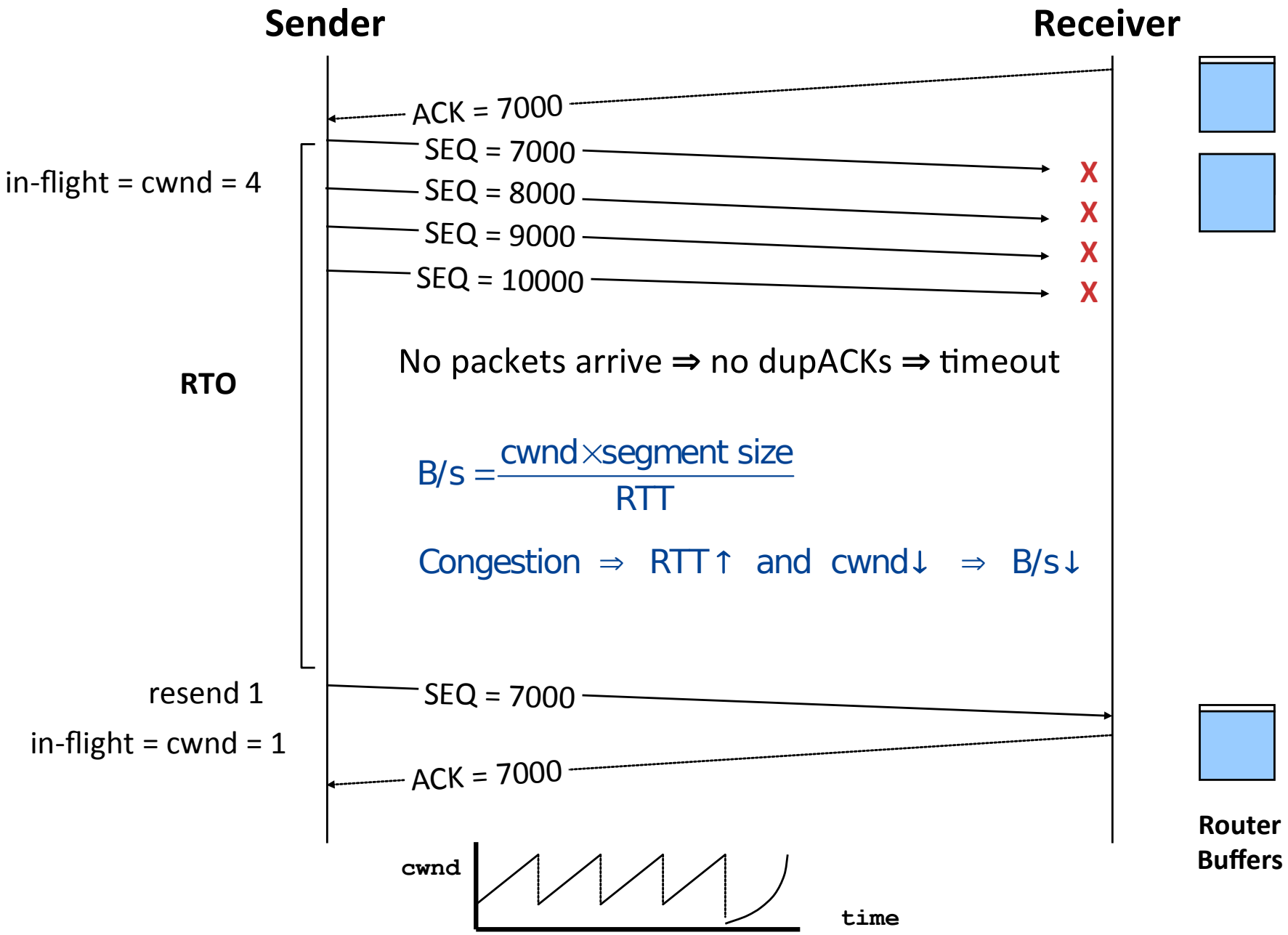
Classic TCP-Reno — 1 Lost Packet



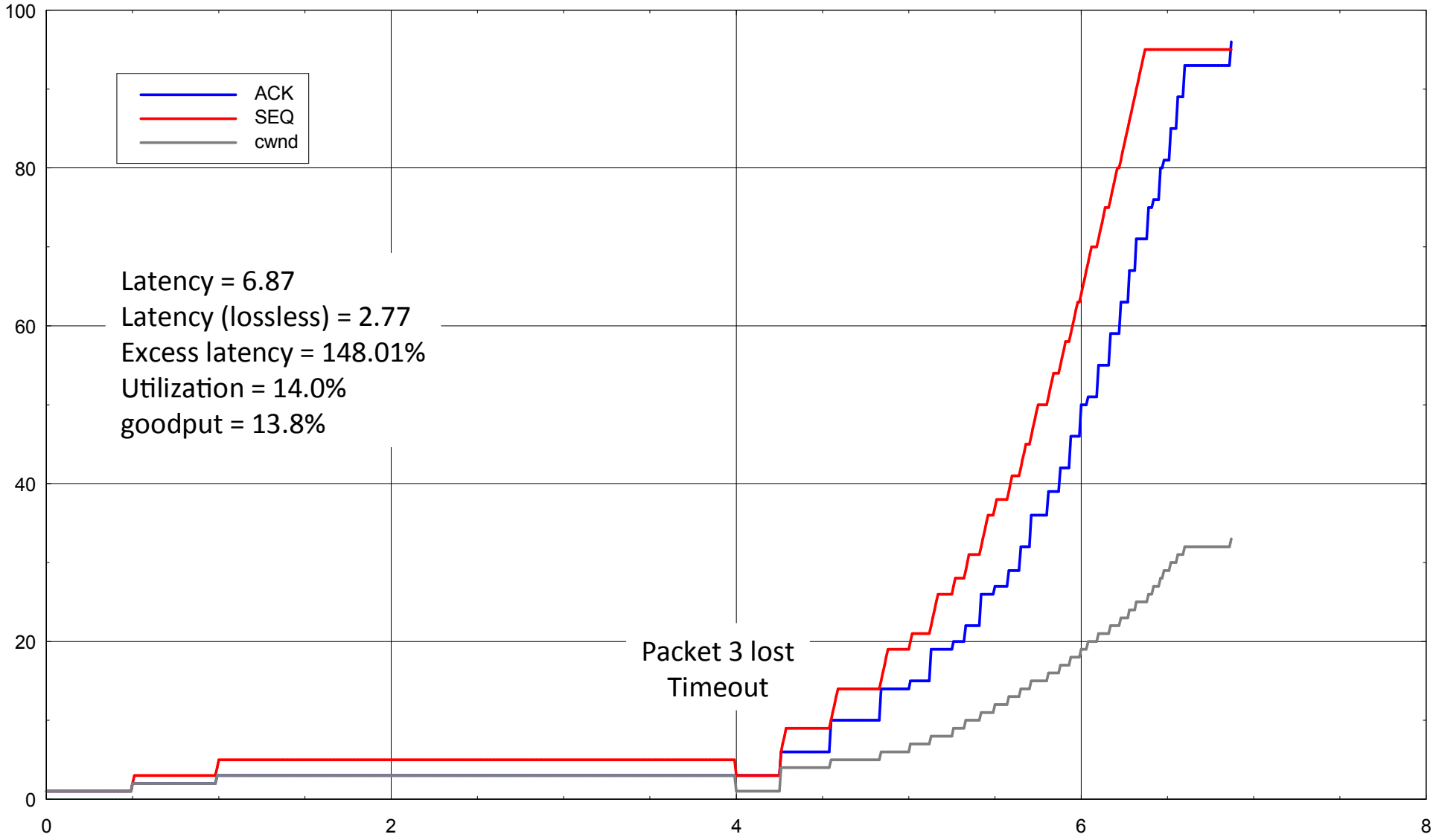
1 Lost Packet



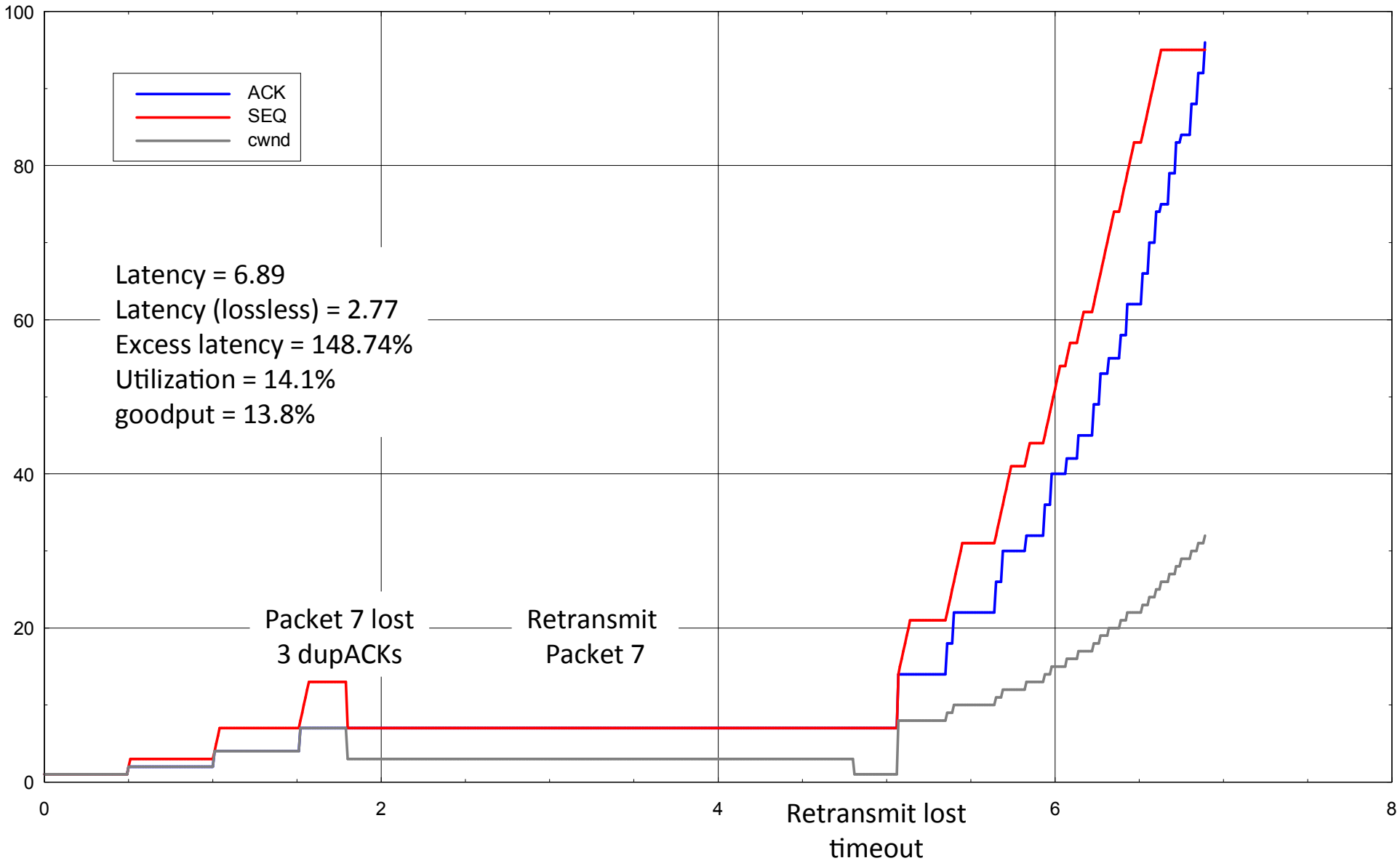
Classic TCP-Reno — Serious Congestion



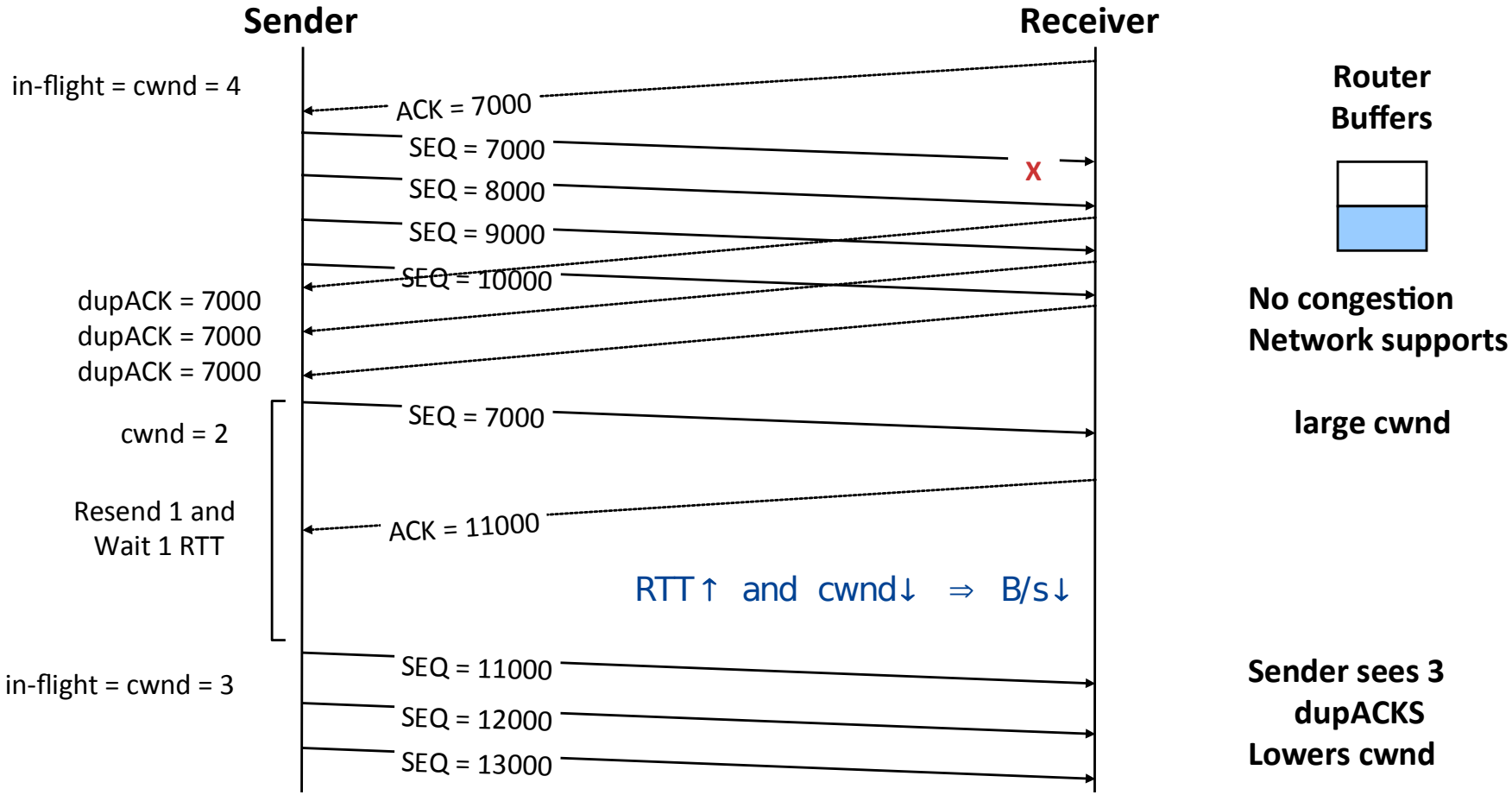
1 Lost Packet — Early



Lost Packet + Lost Retransmit



Classic TCP-Reno — 1 Packet Error

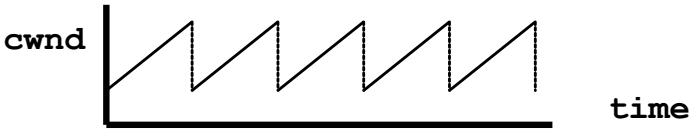


Reno assumes lost packet ⇒ congestion

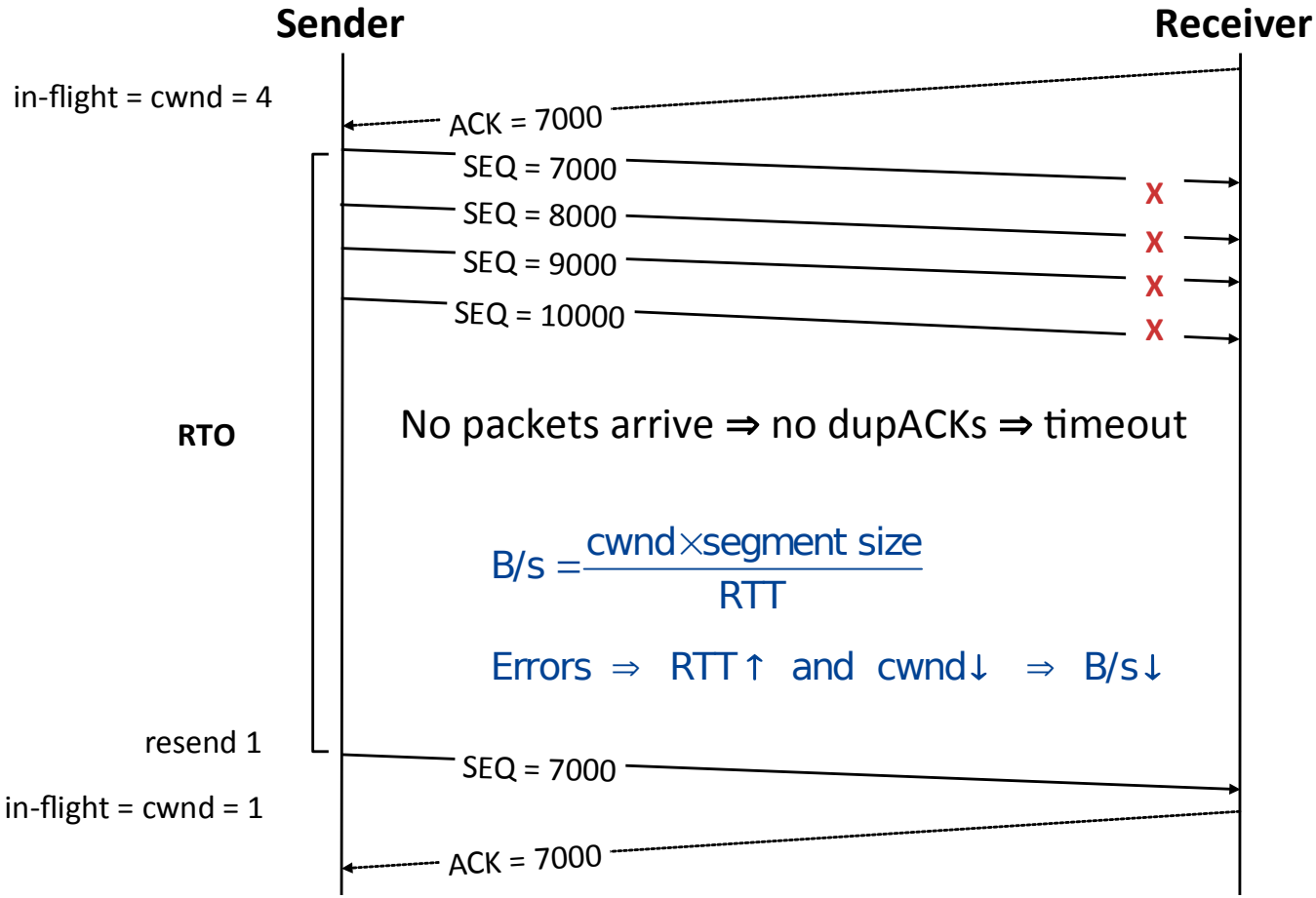
Low packet error on wired Ethernet — BER ≈ 10^{-5}

Moderate response

Lose 1 RTT + time to recover cwnd



Classic TCP-Reno — Multiple Packet Errors



Router Buffers



No congestion
Network supports

large cwnd

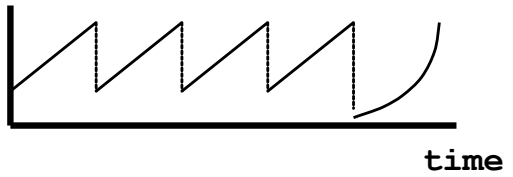
Sender timeout
Lowers cwnd

Reno assumes multiple lost packets ⇒ serious congestion

P(k consecutive packet errors on Ethernet) ≈ $(10^{-5})^k$

Extreme response

Lose RTO + time to recover cwnd in slow start



The Trouble with Wireless — 1

Variations in transmission medium

Multiple correlated packet losses + bit errors

TCP interprets as serious congestion — timeouts + slow start

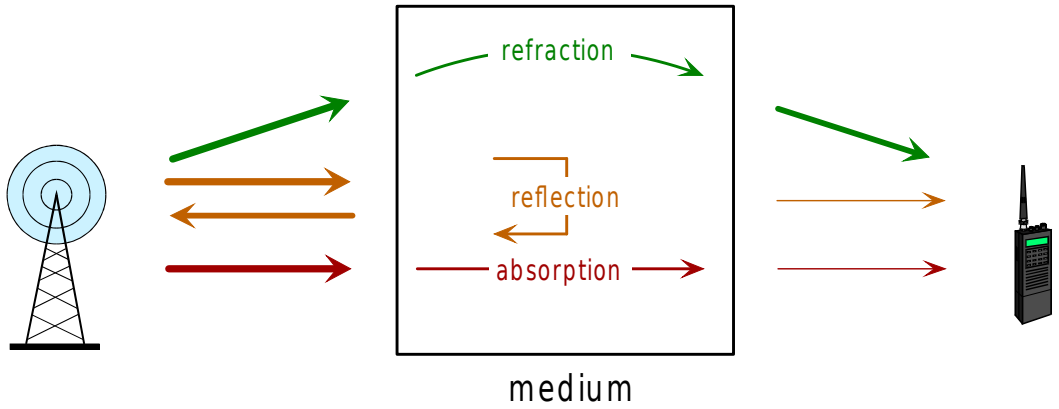
Fading channels

Refraction

Reflection

Absorption

Multipath



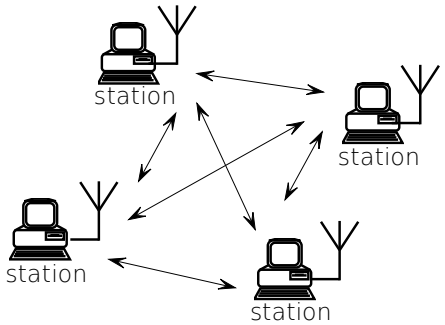
EMI (electromagnetic interference)

Other users

Other radio equipment

Other radiating equipment



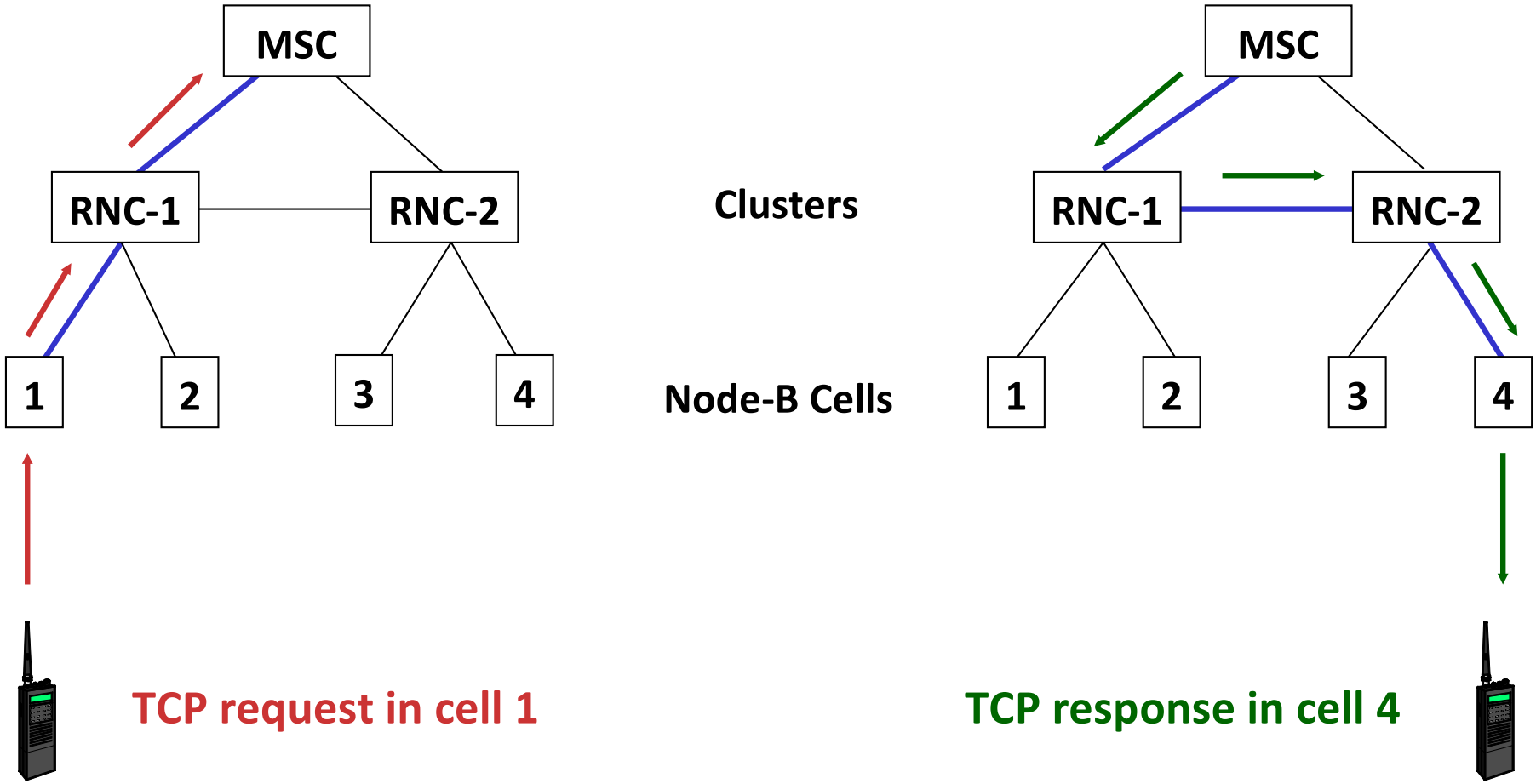


The Trouble with Wireless — 2

Mobility drops

User moves between wireless domains

Adds delays + buffers + dupACKs + timeouts



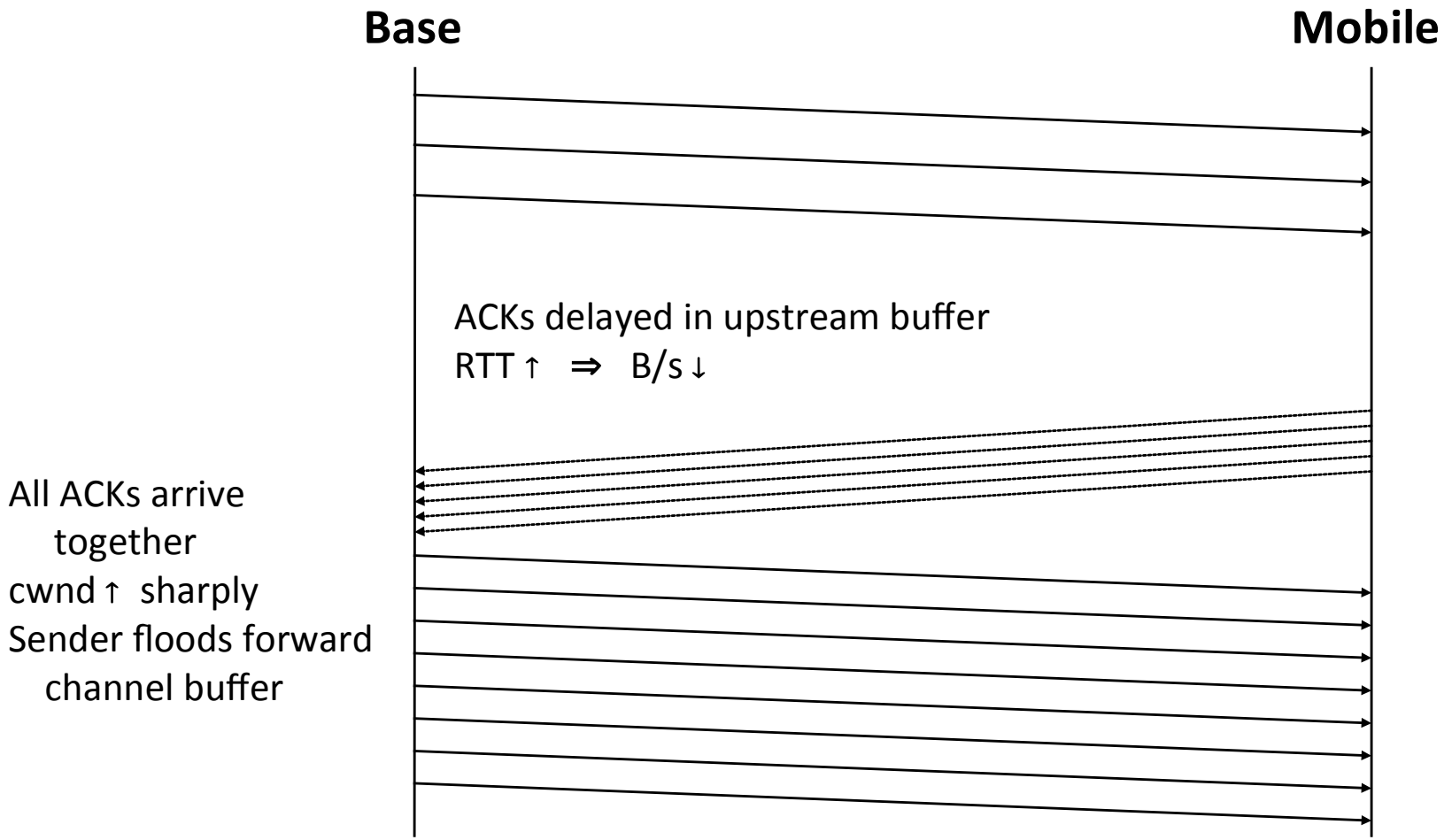
The Trouble with Wireless — 3

Link asymmetry

Upstream channel slower than downstream channel

Larger buffer ⇒ longer buffer delay ⇒ lower average B/s

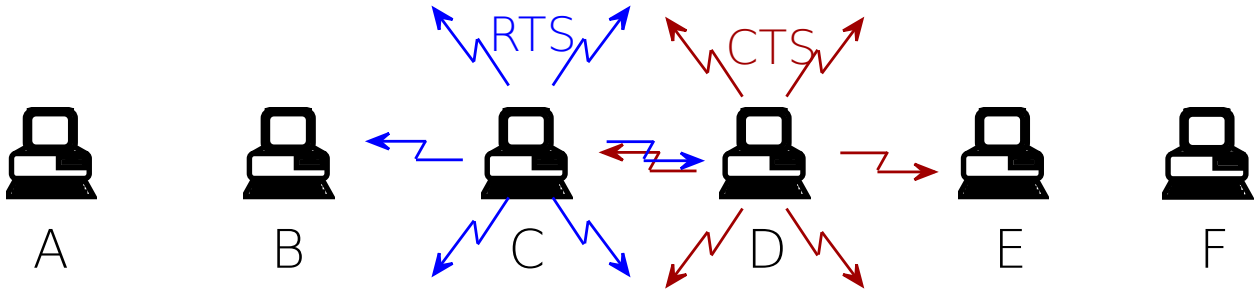
ACK compression

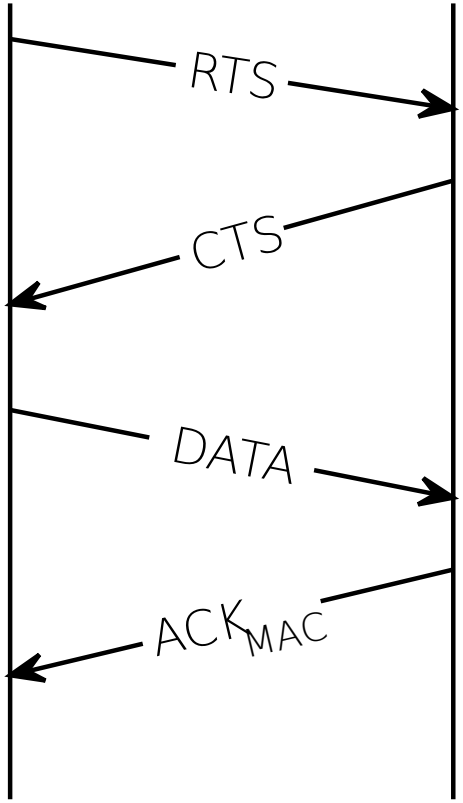


The Trouble with Wireless — 4

MACA in WiFi

Required to prevent hidden node problem





MACA overhead

RTS+CTS+ACK_{MAC} time ≈ TCP ACK time

TCP WiFi ACK delay = 2 × wired TCP ACK delay ⇒ RTT ↑

Timeouts ⇒ cwnd ↓

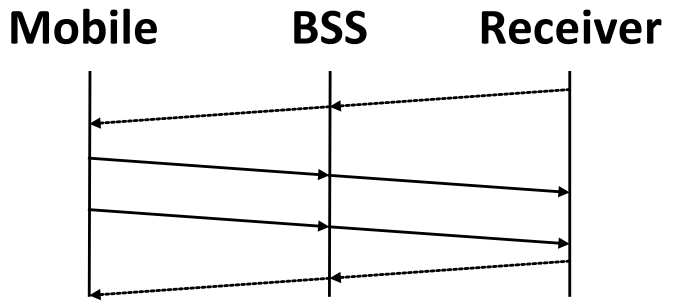
$$B/s = \frac{cwnd \times segment\ size}{RTT}$$

RTT ↑ and cwnd ↓ ⇒ B/s ↓

Typology of Approaches to Wireless TCP

Split mode

- MS / BSS — specialized TCP
- BSS / server — standard TCP
- Proposed in 1990s



End-to-end

- Refinements to TCP-Reno
  - Better handling of non-congestion packet loss
  - Example — TCP → TCP-Tahoe → TCP-Reno
- Infrastructure-awareness
  - Use IP and infrastructure layer information at TCP layer
  - Example — ECN provides IP router information to TCP

Reactive congestion control

- React to congestion loss
- Example — Reno cuts cwnd after 3 dupACKs

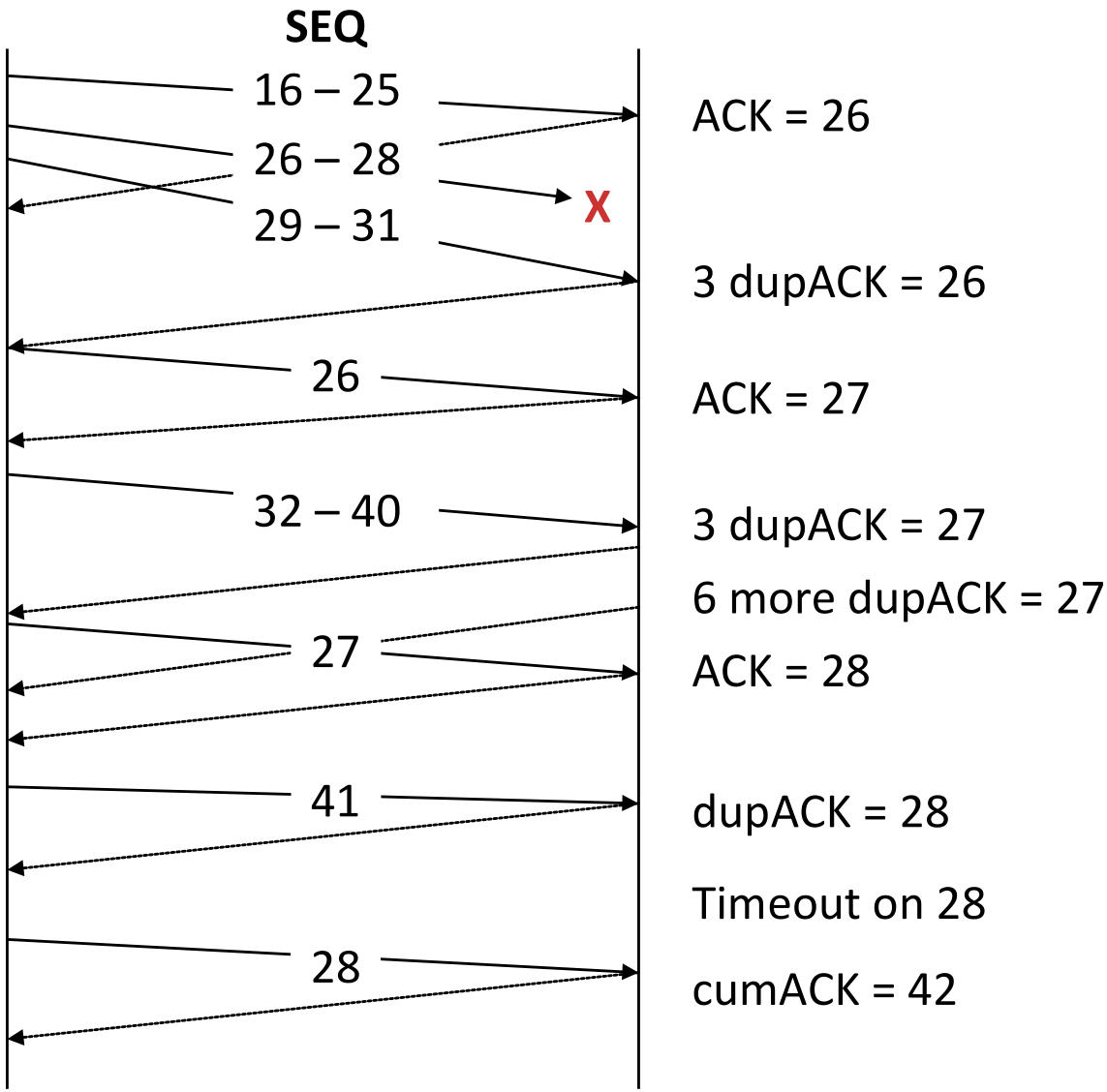
Proactive congestion control

- Predict available bandwidth from ACK rates

Multiple Losses in Reno

Example — 3 consecutive lost packets of 16

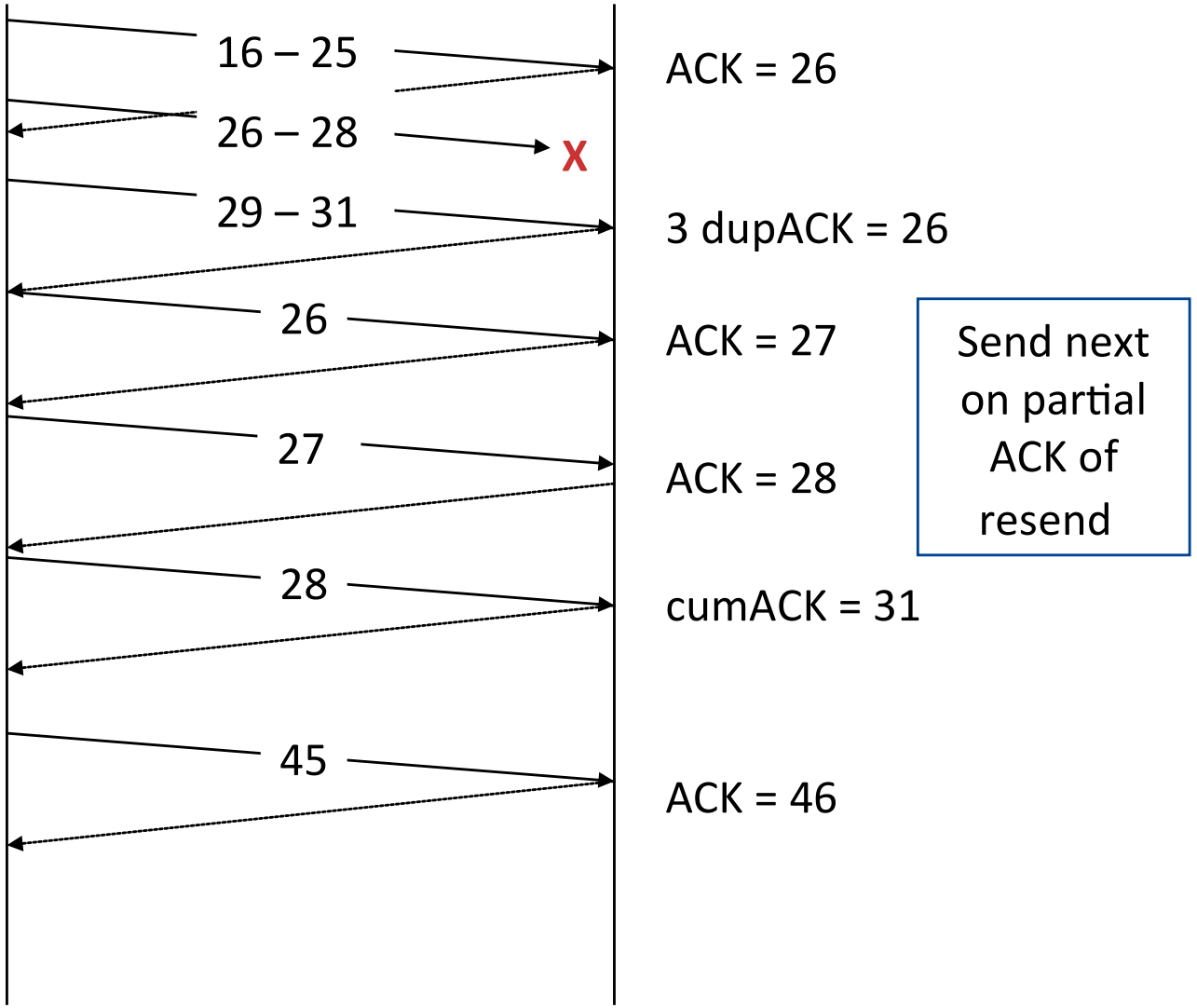
cwnd	in-flight
16	16
26	6
13	6
14	5
14	14
7	14
13	14
14	13
14	14
1	14
15	0



New Reno — RFC 6582

Maintain fast recovery until all in-flight ACKed

cwnd	in-flight
16	16
26	6
13	6
14	5
14	4
15	0
15	15
16	



Send next on partial ACK of resend

TCP-NCL (Non-Congestive Loss)

Heuristic

Timeout could be error

Retransmission timeout probably congestion

Enhancement to Reno

Treat first timeout as error

Treat long timeout as congestion

Two timeouts

Retransmission Decision (RD)

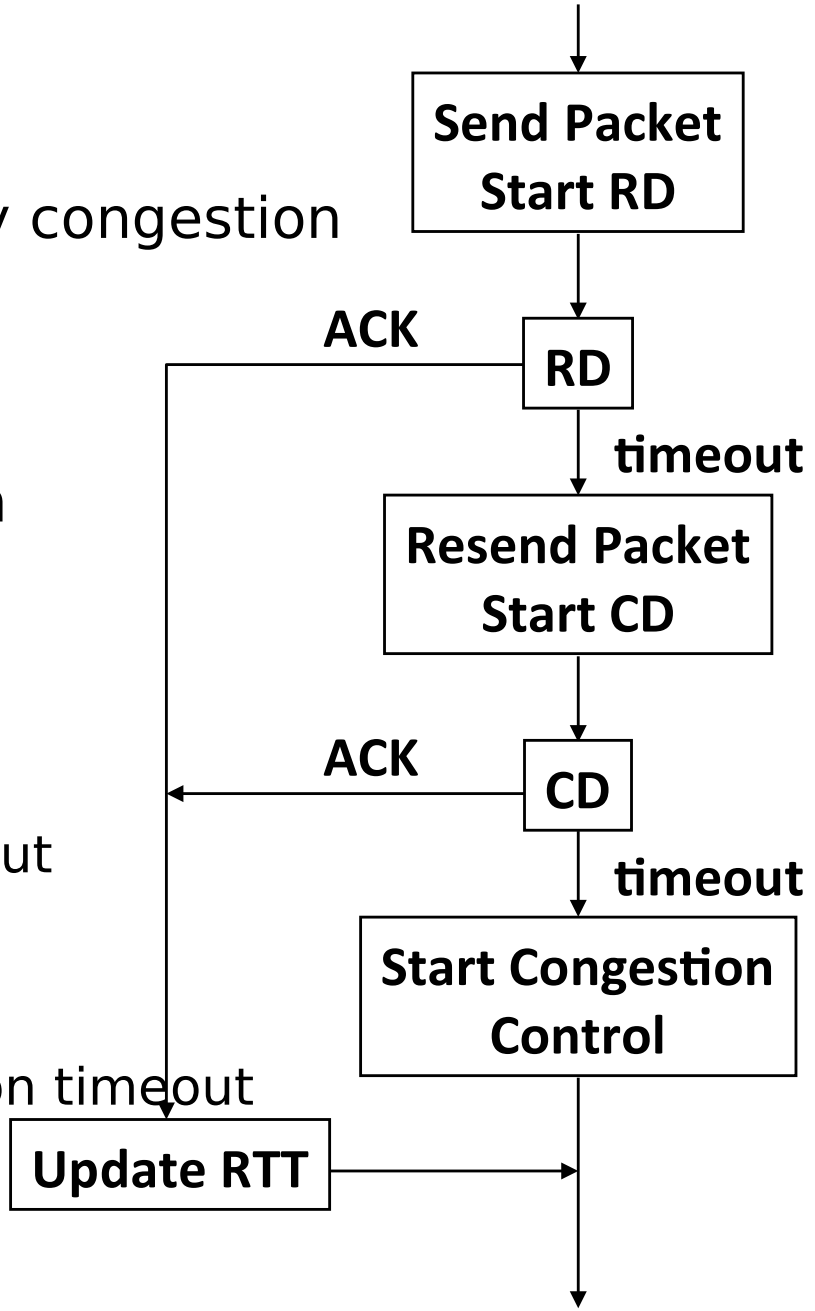
Cancel on ACK

Retransmit on timeout

Congestion Decision (CD)

Cancel on ACK

Congestion control on timeout



C. Lai, K. Leung, V.O.K. Li, "Enhancing Wireless TCP: A Serialized-Timer Approach", IEEE INFOCOM, 2010.

TCP Vegas

Standard Reno slow start

L. Brakmo and L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet," IEEE JSAC, vol. 13, no. 8, Oct. 1995, pp. 1465-80.

`cwnd = 1`

`cwnd ← cwnd + size of data ACKed`

Base expectation

`expected throughput = cwnd / RTT`$_{\texttt{min}}$

Define thresholds α, β

Enhanced Reno congestion avoidance

Measure **`RTT`** for distinguished (special) segment of some size

`actual throughput = size of ACK received / RTT`

`diff = expected throughput - actual throughput`

if (**`diff`** $< \alpha$) then **`cwnd++`** on each ACK

if (**`diff`** $> \beta$) then **`cwnd--`** on each ACK

dupACKs

Retransmit on 1 **`dupACK`** if **`RTT > RTT`**$_{\texttt{Vegas-timeout}}$

TCP — Modified Fast Retransmission (MFR)

Reno

Transmit cwnd packets

Packet n lost → k dupACKs → retransmit on k ≥ 3 dupACKs

Wait for ACK of n ⇒ no dupACKs if retransmit lost ⇒ timeout

Enhancement to Reno fast retransmit

k dupACKs ⇒ k packets arrived

k dupACKs + 1 lost packet = k+1 packets accounted

cwnd – (k+1) unaccounted

If (k > 3) cwnd = cwnd + (k – 3) → send more packets

Cumulative ACK ⇒ retransmitted packet + more arrived

If (dupACKs – 3 > unaccounted)

New packets arrived but retransmitted n lost

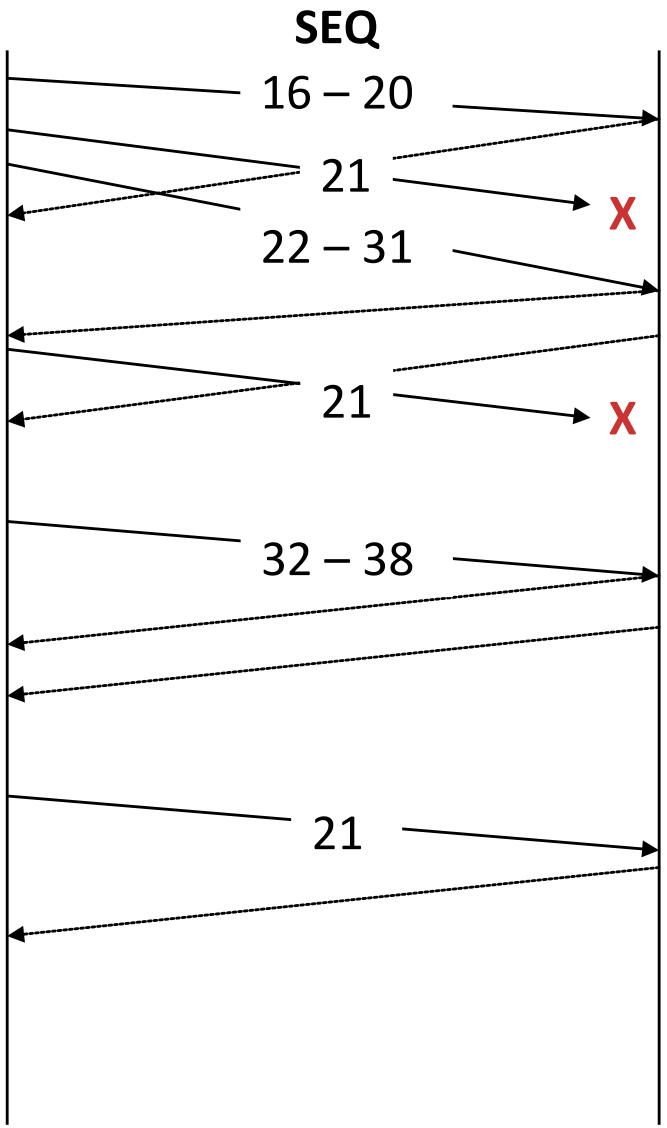
Retransmit n again — no timeout

S. Prasanthi, S. Chung, "An Efficient Algorithm for the Performance of TCP over Multi-hop Wireless Mesh Networks", Seventh International Conference on Information Technology, pp. 816 – 821, 2010.

Modified Fast Retransmission — Example

Prevents timeout on repeated loss

cwnd	in-flight
21	16
26	11
13	11
18	11
18	18
9	18
18	18
18	18
19	0



ACK = 21

3 dupACK = 21
5 more dupACK = 21 (8 of 10)

16 sent – 5 ACK – 8 dupACK
– 1 lost = 2 ???

3 dupACK = 21 (late or new?)
6 more dupACK = 21

6 more dupACKS > 2 ???
Resend 21*

cumACK = 39

* If more dupACKS = 2 either:
21 lost again
Late dupACKs from 22 – 31

Modified Congestion Control (MCC)

Reno congestion control

ssthresh = cwnd

cwnd = cwnd / 2

Approach: Oops! Went to far. Cut back half.

Bandwidth Estimate (BWE)

$$\text{BWE} = \text{weighted average of } \frac{\text{bytes ACKed}}{\Delta\text{RTT for bytes_to_ACK}}$$

$$\text{BWE_s[k]} = \frac{\text{packets ACKed} \times \text{bytes per packet}}{\text{now - last_ACK_time}}$$

$$\text{BWE[k]} = (1-\beta) \times \frac{\text{BWE_s[k] + BWE_s[k-1]}}{2} + \beta \times \text{BWE[k-1]}$$

L. Yongmei, J. Zhigang, Z. Ximan, "A New Protocol to Improve Wireless TCP Performance and Its Implementation", 5th International Conference on Wireless Communications, Networking and Mobile Computing, 2009.

Modified congestion control

3 dupACKs

ssthresh = (BWE * ΔRTTmin) / segment size

cwnd = ssthresh if (cwnd > ssthresh)

Timeout

cwnd = 1

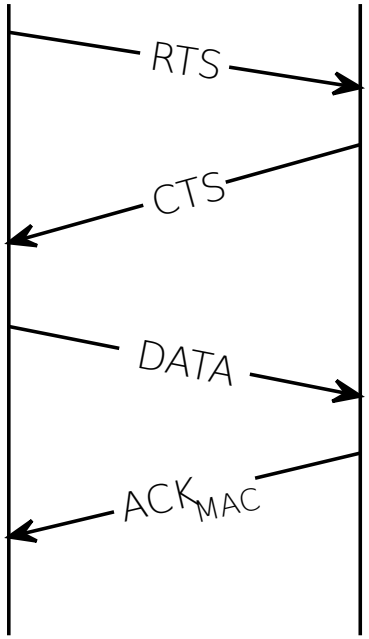
ssthresh = max{(BWE*ΔRTTmin) / segment size, 2}

Proposed Solutions for WiFi Overhead

Problem: MACA overhead

Each TCP ACK ⇒ RTS+CTS+ACK_{MAC}

RTS+CTS+ACK_{MAC} time ≈ TCP ACK time



Solution: reduce control traffic

Increase TCP ACK delay

Buffer more TCP ACKs in receiver

ACK 4 TCP segments instead of 2

Reduce ACK traffic by 50%

BWE not changed

Bits ACKed ↑ and RTT ↑

Problem: longer ACK delay ⇒ lower utilization

Sender waits longer for ACKs

Solution: infrastructure awareness

Sender writes cwnd in TCP option field

Immediate ACK if unACKed data in receiver ACK buffer ≈ cwnd

Freeze-TCP

Handles mobility drops

- Handoff disconnect / reconnect
- Temporary fading by obstacles

Mobile station monitors radio signal strength

- Predicts impending disconnections
- Freeze-TCP receiver proactively sets window = 0
- TCP sender enters persist mode

On reconnect

- Receiver sends multiple ACKs of last received packet
- Prevents exponential back off of cwnd in sender

Infrastructure awareness

- TCP layer exposed to details handled by lower layers
- Roaming / handoff at layers 2 / 3
- Radio signal quality at layers 1 and 2

TCP-Jersey

Available Bandwidth Estimation (ABE)

Similar to BWE expressed in segments

Congestion Warning (CW)

Requires explicit congestion notification (ECN) support in routers

IP routers mark packets on congested link

TCP receiver returns ECN warning to TCP sender

CW without loss

cwnd = ABE

3 dupACKs

Retransmit

If (CW in dupACK segment) cwnd = ABE

Else no change to cwnd

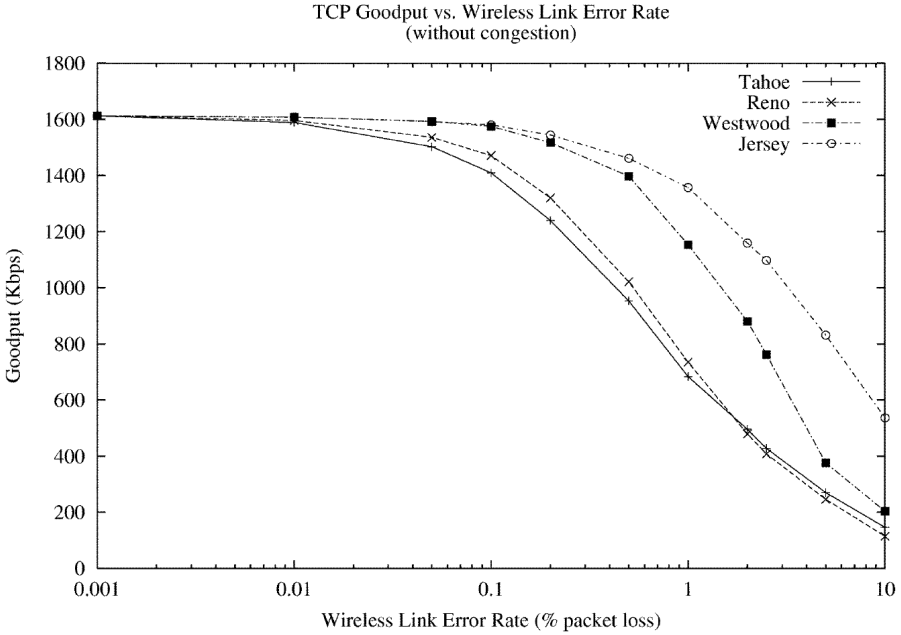
Timeout

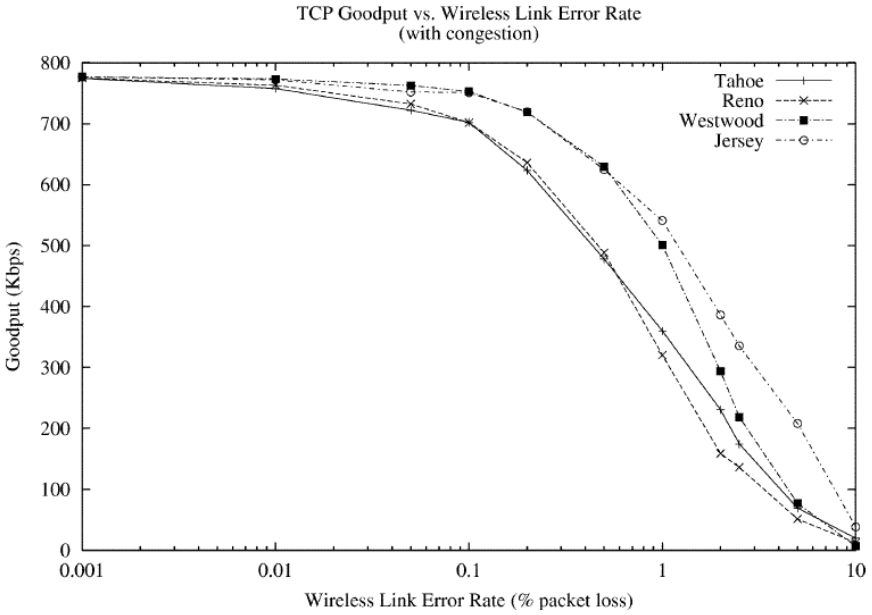
Retransmit

If (CW in dupACK segment) cwnd = 1

Else no change to cwnd

TCP-Jersey Results





TCP Westwood

Modifies TCP sender

Not dependent on negotiation with TCP receiver

Not dependent on ECN support

Estimates available bandwidth

Counts ACKs and dupACKs as successful traffic

On packet loss set cwnd = available bandwidth

Improves on Reno cwnd = cwnd / 2

Performance almost as good as Jersey

Westwood Packet Counting

Reno enhancement

Standard slow start

Standard congestion avoidance

Calculate bandwidth estimate (BWE) on each ACK

$$BWE_k = \frac{d_k}{t_k - t_{k-1}} = \frac{\text{number of bytes ACKed at time } t_k}{\text{me of ACK - time of previous ACK}}$$

Bandwidth sample

$$BWE = \alpha_k BWE_k + (1 - \alpha_k)\frac{BWE_k + BWE_{k-1}}{2}$$

Bandwidth estimate

Handling isolated error

Out-of-order packets $\longrightarrow$ dupACKs

dupACK $\Rightarrow$ some packets **ARRIVED** at receiver

Count dupACKs in bandwidth estimate

On ACK of lost packet — treat dupACK as already counted

Serious congestion

Causes timeouts instead of dupACKs

Westwood Packet Counting Example

Packet transmission times

1	2	3	4	5	6	7	8	5	9
t_0	t_1		t_2				t_3	t_6	

Time	ACK	Arrived at Receiver	Packets for BWE	Counted dupACK	BWE
t_1	2	1	1	0	$1 / (t_1 - t_0)$
t_2	4	2 + 3	2	0	$2 / (t_2 - t_1)$
t_3	5	4	1	0	$1 / (t_3 - t_2)$
t_4	5	6	1	1	$1 / (t_4 - t_3)$
t_5	5	7	1	2	$1 / (t_5 - t_4)$
t_6	5	8	1	3	$1 / (t_6 - t_5)$
t_7	10	5 + 9	**2**	0	$2 / (t_7 - t_6)$

ACK jumps to 10 from 5 but 3 ACKs (6, 7, 8) already counted as dupACKs

Westwood ACK Counter (for BWE)

`newACK = ACK – prevACK ;`	Packets ACKed by new ACK
`// if (newACK = 1) do nothing`	No error condition — report 1 ACK

`if (newACK = 0)`
`  count++ ;`
`  newACK = 1 ;`
	ACK "stuck" on old value (dupACK)
	Increment dupACK counter
	Count 1 dupACK

`if (newACK > 1)`	ACK advances

`  if (count >= newACK)`
`  count = count – newACK + 1 ;`
`  newACK = 1 ;`
	Not all arrived packets ACKed
	Remove ACKed from dupACK count
	Count as additional dupACK

`  else if (count < newACK)`
`  newACK = newACK - count ;`
`  count = 0;`
	All packets ACKed in order
	Report ACKed – counted (new ACKs)
	Zero dupACK counter

`prevACK = ACK ;` `return(newACK);`	

Westwood Scenario 1 — No Errors

Receive 1 — 2,3 — 4,5,6,7 — 8,9,10,11,12,13,14,15

ACK 2 — 4 — 8 — 12 — 16 −→ newACK = 1 — 2 — 4 — 4 — 4

`newACK = ACK - prevACK ;`	1	2	4	4	4

`if (newACK = 0)`
`  count++ ;`
`  newACK = 1 ;`
	—	—	—	—	—

`if (newACK > 1)`					

`  if (count >= newACK)`
`  count = count - newACK + 1 ;`
`  newACK = 1 ;`
	—	—	—	—	—

`  else if (count < newACK)`
`  newACK = newACK - count ;`
`  count = 0;`
		2
		0
			4
			0
				4
				0
					4
					0

`prevACK = ACK ;`	2	4	8	12	16
`return(newACK);` Total = 15	1	2	4	4	4

Westwood Scenario 2 — 1 Packet Out-of-Order

Receive 1 — 2,3 — 5,4,6,7 — 8,9,10,11,12,13,14,15

ACK 2 — 4 — 4 — 6 — 10 — 14 — 16

`newACK = ACK - prevACK ;`	1	2	0	2	4	4	2

`if (newACK = 0)`
`  count++ ;`
`  newACK = 1 ;`
	—	—	1
			1
				—	—	—	—

`if (newACK > 1)`							

`  if (count >= newACK)`
`  count = count - newACK + 1;`
`  newACK = 1 ;`
	—	—	—	—	—	—	—

`  else if (count < newACK)`
`  newACK = newACK - count ;`
`  count = 0;`
		2
		0
				1
				0
					4
					0
						4
						0
							2
							0

`prevACK = ACK ;`	2	4	4	6	10	14	16
`return(newACK);` Total = 15	1	2	1	1	4	4	2

Westwood Scenario 3 — 1 Packet Very Out-of-Order

Receive 1 — 2,3 — 5,6,7,4 — 8,9,10,11,12,13,14,15

ACK 2 — 4 — 4 — 4 — 4 — 8 — 12 — 16

`newACK = ACK - prevACK ;`	1	2	0	0	0	4	4	4

`if (newACK = 0)`
`  count++ ;`
`  newACK = 1 ;`
	—	—	1
			1
				2
				1
					3
					1
						—	—	—

`if (newACK > 1)`								

`  if (count >= newACK)`
`  count = count - newACK + 1 ;`
`  newACK = 1 ;`
	—	—	—	—	—	—	—	—

`  else if (count < newACK)`
`  newACK = newACK - count ;`
`  count = 0;`
		2
		0
						1
						0
							4
							0
								4
								0

`prevACK = ACK ;`	2	4	4	4	4	8	12	16
`return(newACK);` Total = 15	1	2	1	1	1	1	4	4

Westwood Scenario 4 — Variation of Scenario 3

Receive 1 — 2,3 — 5,6,7,4 — 8,9,10,11,12,13,14,15

ACK 2 — 4 — 4 — 4 — 4 — 5 — 9 — 13 — 16

`newACK = ACK - prevACK ;`	1	2	0	0	0	1	4	4	3

`if (newACK = 0)`
`  count++ ;`
`  newACK = 1 ;`
	—	—	1
			1
				2
				1
					3
					1
						—	—	—	—

`if (newACK > 1)`									

`  if (count >= newACK)`
`  count = count - newACK + 1 ;`
`  newACK = 1 ;`
	—	—	—	—	—	—	—	—	—

`  else if (count < newACK)`
`  newACK = newACK - count ;`
`  count = 0;`
		2
		0
							1
							0
								4
								0
									3
									0

`prevACK = ACK ;`	2	4	4	4	4	5	9	13	16
`return(newACK);` Total = 15	1	2	1	1	1	1	1	4	3

Westwood Scenario 5 — Another Variation of Scenario 3

Receive 1 — 2,3 — 5,6,7,4 — 8,9,10,11,12,13,14,15

ACK 2 — 4 — 4 — 4 — 4 — 6 — 10 — 14 — 16

`newACK = ACK - prevACK ;`	1	2	0	0	0	2	4	4	2

`if (newACK = 0)`
`  count++ ;`
`  newACK = 1 ;`
	—	—	1
			1
				2
				1
					3
					1
						—	—	—	—

`if (newACK > 1)`									

`  if (count >= newACK)`
`  count = count - newACK + 1 ;`
`  newACK = 1 ;`
	—	—	—	—	—	2
						1
							—	—	—

`  else if (count < newACK)`
`  newACK = newACK - count ;`
`  count = 0;`
		2
		0
						—	2
							0
								4
								0
									2
									0

`prevACK = ACK ;`	2	4	4	4	4	5	10	14	16
`return(newACK);` Total = 15	1	2	1	1	1	1	2	4	2

Westwood Scenario 6 — 2 Packets Out-of-Order

Receive 9,10,11,13,14,15,8,12 (after ACK = 8)

ACK 8 — 8 — 8 — 8 — 8 — 8 — 12 — 16

`newACK = ACK - prevACK ;`	0	0	0	0	0	0	4	4

`if (newACK = 0)`
`  count++ ;`
`  newACK = 1 ;`
	1
	1
		2
		1
			3
			1
				4
				1
					5
					1
						6
						1
							—	—

`if (newACK > 1)`	—	—	—	—	—	—		

`  if (count >= newACK)`
`  count = count - newACK + 1 ;`
`  newACK = 1 ;`
							3
							1
								—

`  else if (count < newACK)`
`  newACK = newACK - count ;`
`  count = 0;`
							—	1
								0

`prevACK = ACK ;`	7	7	7	7	7	7	11	15
`return(newACK);` Total = 15	1	1	1	1	1	1	1	1

Westwind Congestion Control

Reno slow start

```
On (ACK && cwnd < ssthresh)
    cwnd ← cwnd + size of data ACKed
```

Reno congestion avoidance

```
On (ACK && cwnd > ssthresh)
    cwnd ← cwnd + 1
```

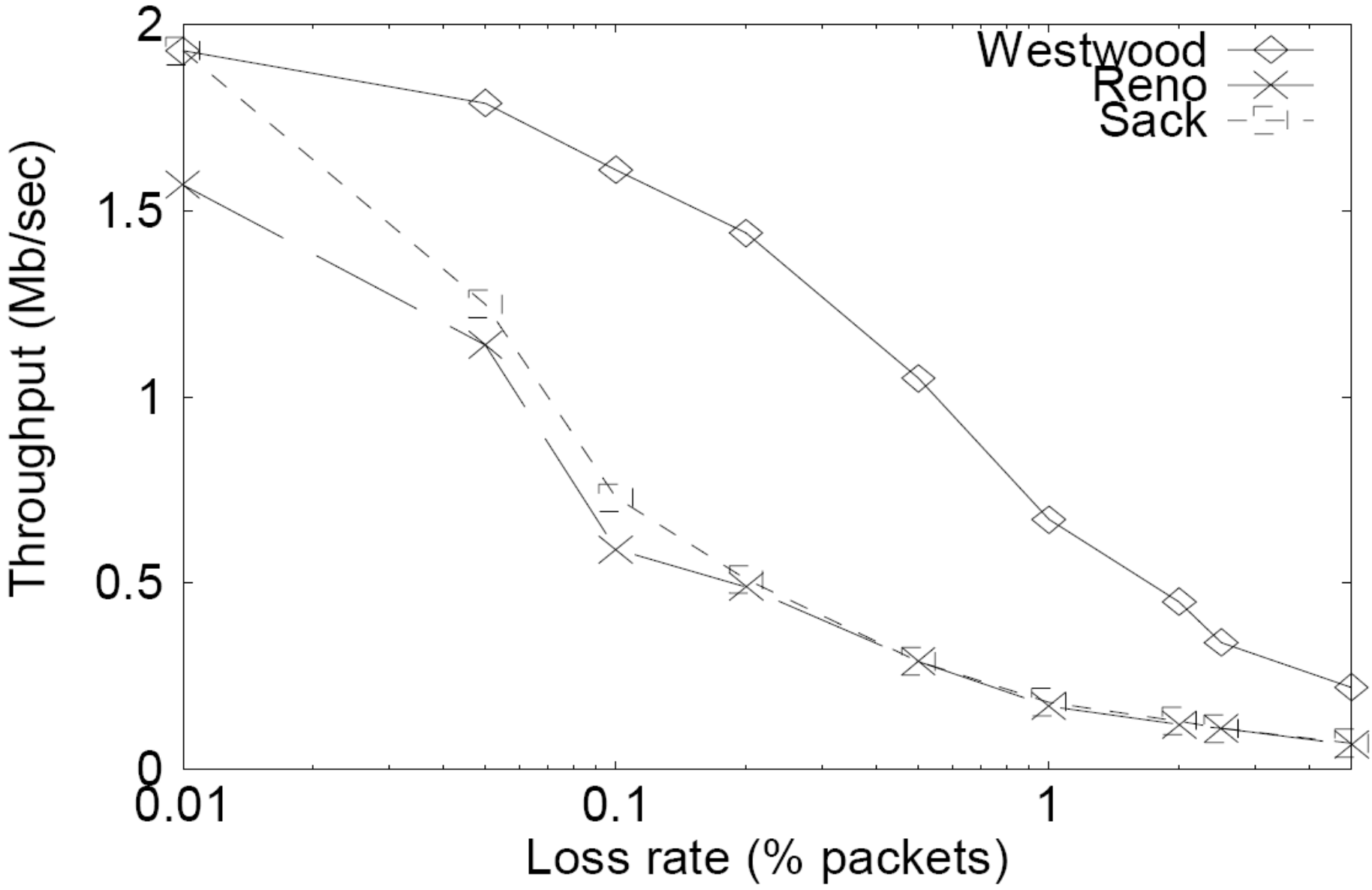
Modified fast recovery

```
On n dupACKs
    ssthresh = BWE * RTT_min / segment_size
    if (cwnd > ssthresh) cwnd = ssthresh
```

Modified timeout

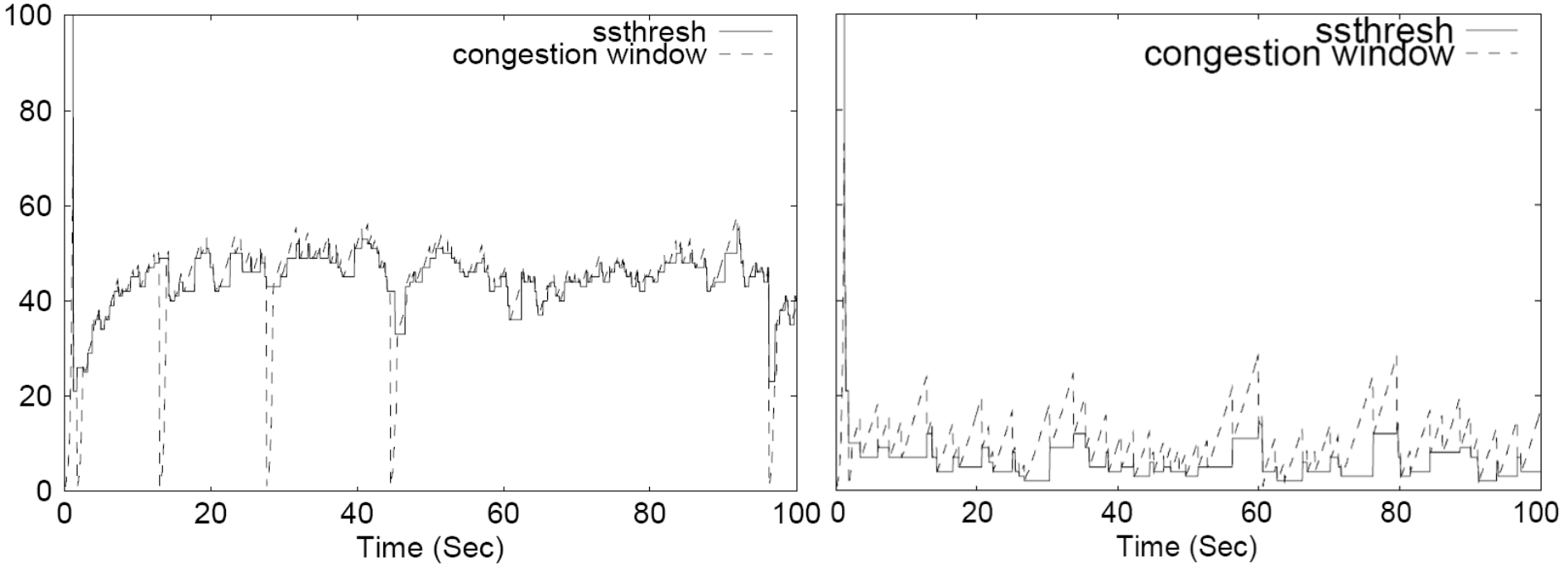
```
ssthresh = BWE * RTT_min / segment_size
if (ssthresh < 2) ssthresh = 2
cwnd = 1
```

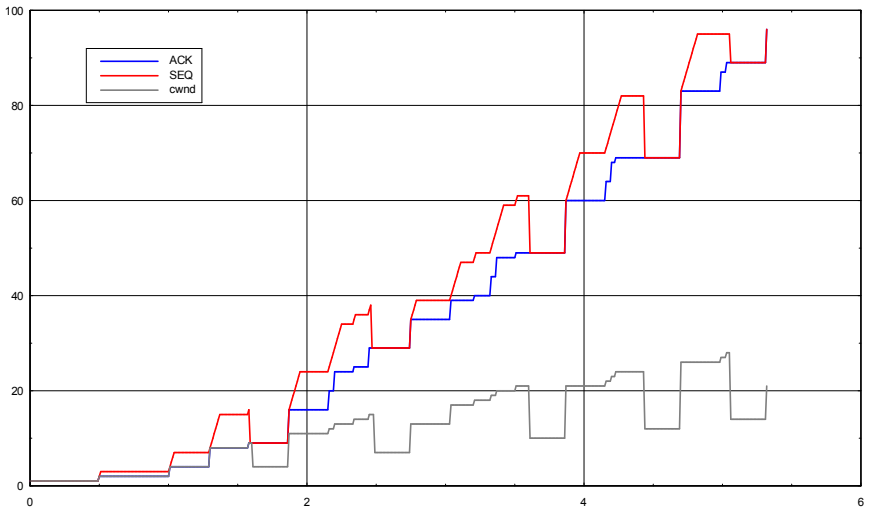
Westwood Performance — Throughput



Westwood — cwnd and ssthresh

Westwood versus Reno





Steady state cwnd (slide 10)