

Computer Networking

Introduction

Prof. Andrzej Duda
duda@imag.fr

`http://duda.imag.fr`

Course goals

- Understand TCP/IP and networking concepts
- Approach
 - bottom-up, descriptive, use Internet as an example
 - wrap up with the application layer seen during the 1st year course
- Organization
 - 27 h course
 - demos, exercises
 - slides are not exhaustive - you must take notes and ask questions!
 - bonus questions: 5 good answers get 1 point (limited to 5 per person)
- Exam
 - closed-book: no personal notes, textbook, etc., are allowed
 - we provide a summary of required factual knowledge
- Your team
 - Andrzej Duda (in English), Olivier Alphant (en français)

Networking lab

- Important part of the course (separate grade)
 - perform required operations, write lab reports, final exam
 - cannot be repeated
 - grade < 8, you repeat your year!
- Goals
 - acquire practical knowledge
 - plug cables, configure hosts and routers, monitor, measure, program network applications
- Rooms D200 and D201:
 - 80 PCs with multiple network interfaces
 - network equipment: hubs, switches, routers
 - isolated from the rest of the network
- Your team
 - Olivier Alphand, Sébastien Viardot, TAs

Contents

- Introduction
 - architecture, performance
- Data Link
 - PPP, LAN (Ethernet, 802.11)
- Network layer
 - IP, ATM
 - Routing
- Transport
 - reliable transfer protocols
 - TCP, UDP, sockets
 - congestion control

Course support

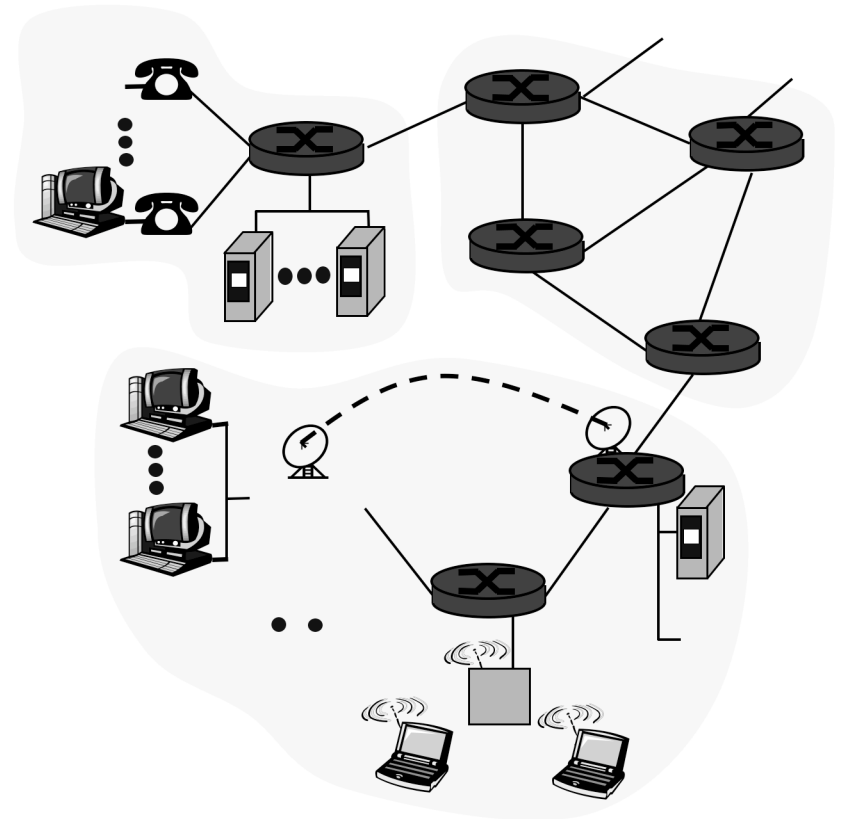
- Web site
 - <http://duda.imag.fr/2at>
- J. Kurose, K. Ross "Computer Networking", 4th edition, Addison Wesley, 2007
- J. Kurose, K. Ross, "Analyse structurée des réseaux. Des applications de l'internet aux infrastructures des télécommunications." Pearson Education France, 2003
- Others
 - L. Toutain "Réseaux locaux et Internet", 3me édition, Hermes, 2003
 - W. R. Stevens "TCP/IP illustrated, Volume I", Addison Wesley (Very detailed, experimental hands-on description of TCP/IP)

Overview

- Network architectures
 - recall on the Internet
 - protocol architectures
 - how entities cooperate?
 - interconnection structure
 - which entities are connected?
 - related protocols
 - how and where different functionalities are implemented?
- Performance
 - transmission
 - propagation
 - bandwidth-delay product
 - queueing delay

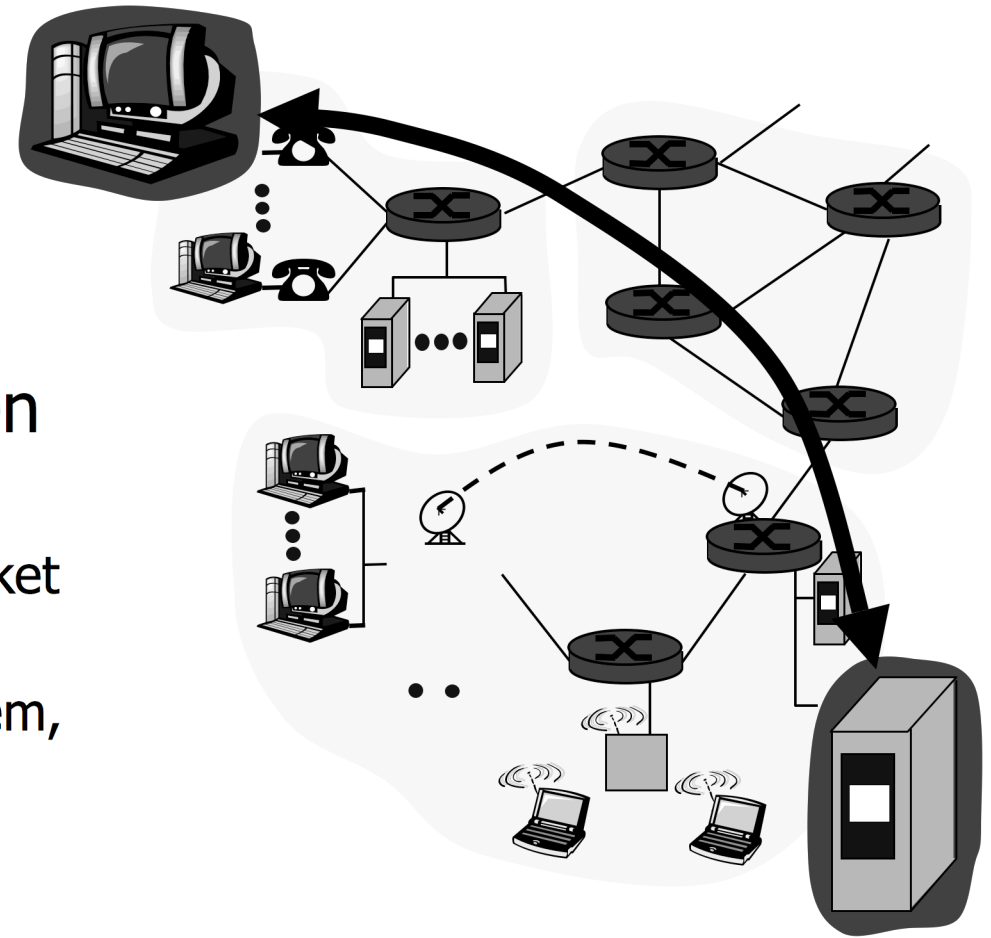
Inside the Internet

- Between end systems
 - TCP protocol for reliable transmission
- Inside the network core
 - IP protocol: forwarding packets between routers
- Between routers or between end system and router
 - high speed link: ATM, POS (Packet over SONET), satellite links
 - access network: Ethernet, modem, xDSL, HFC



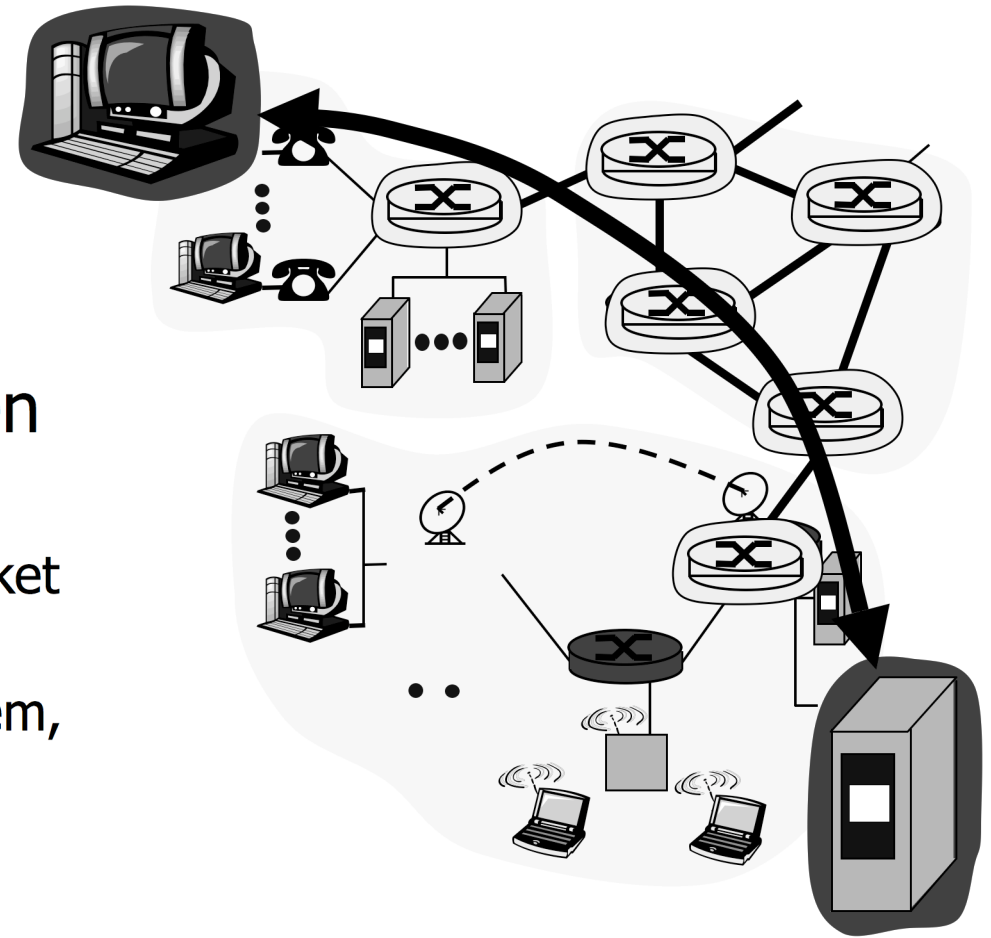
Inside the Internet

- Between end systems
 - TCP protocol for reliable transmission
- Inside the network core
 - IP protocol: forwarding packets between routers
- Between routers or between end system and router
 - high speed link: ATM, POS (Packet over SONET), satellite links
 - access network: Ethernet, modem, xDSL, HFC



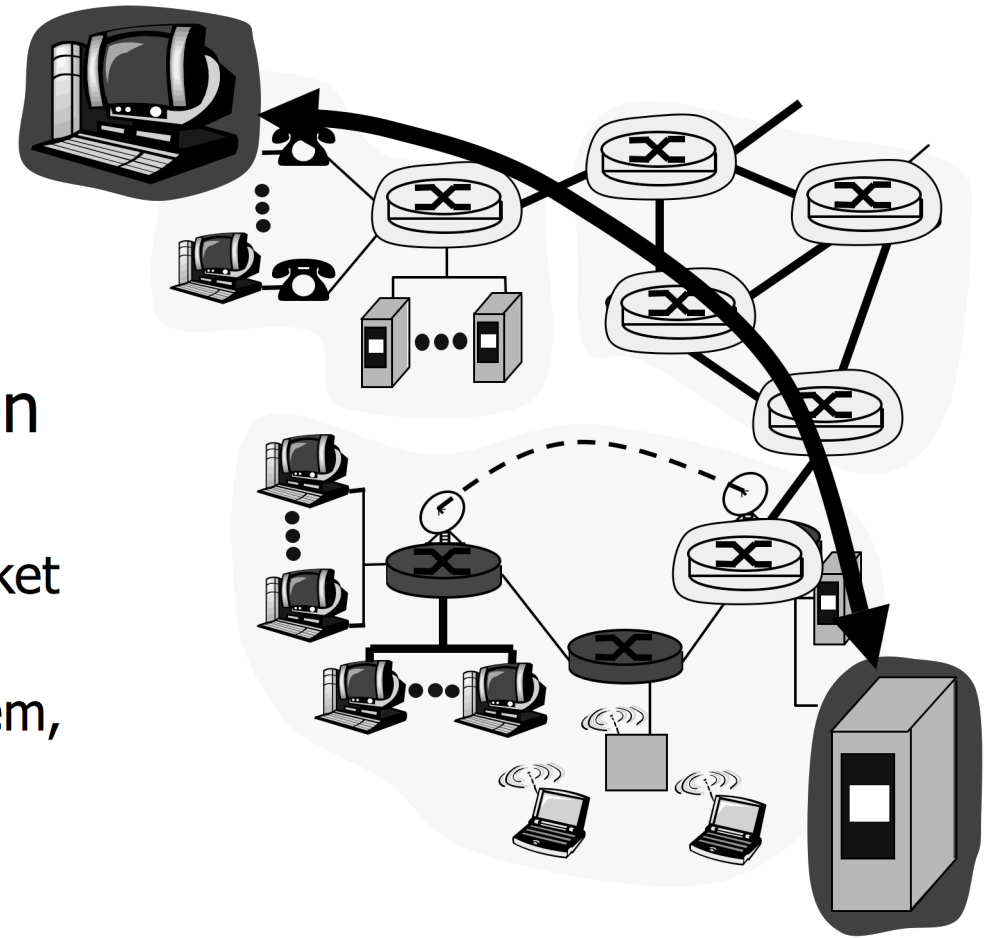
Inside the Internet

- Between end systems
 - TCP protocol for reliable transmission
- Inside the network core
 - IP protocol: forwarding packets between routers
- Between routers or between end system and router
 - high speed link: ATM, POS (Packet over SONET), satellite links
 - access network: Ethernet, modem, xDSL, HFC



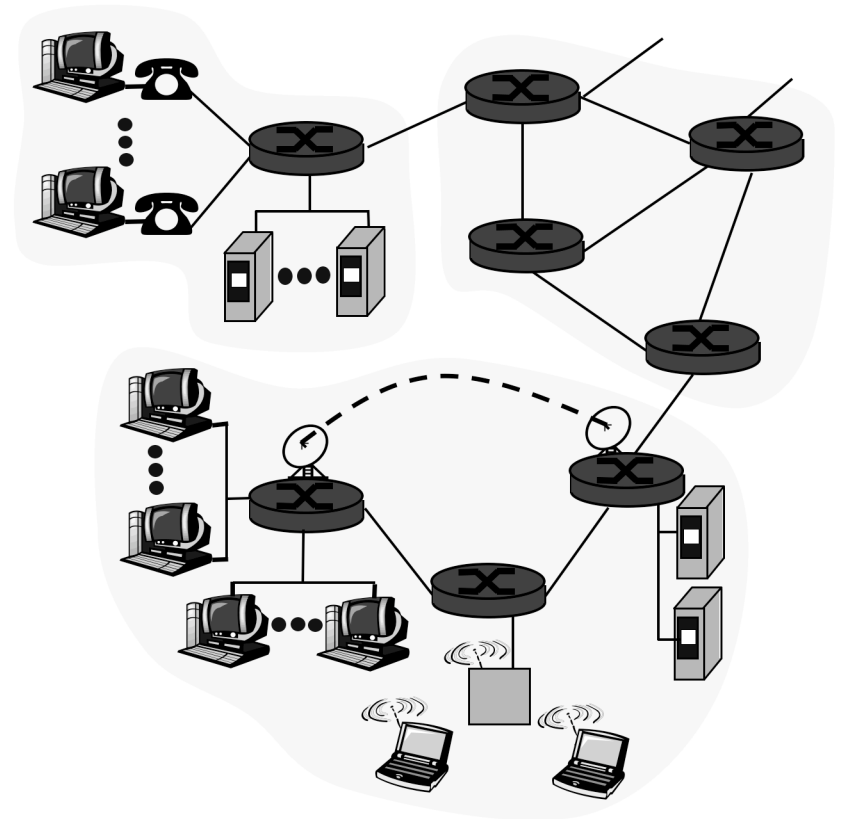
Inside the Internet

- Between end systems
 - TCP protocol for reliable transmission
- Inside the network core
 - IP protocol: forwarding packets between routers
- Between routers or between end system and router
 - high speed link: ATM, POS (Packet over SONET), satellite links
 - access network: Ethernet, modem, xDSL, HFC



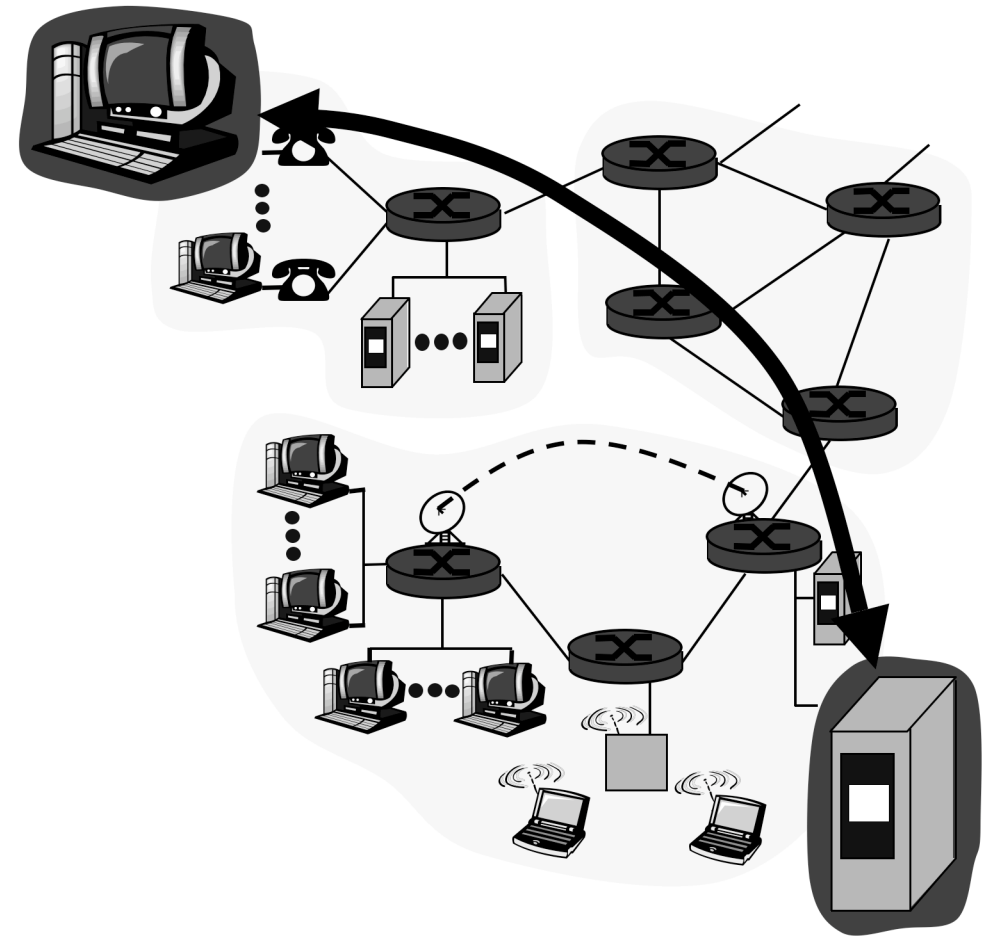
Network structure

- network edge:
applications and hosts
- network core:
 - routers
 - network of networks
- access networks,
physical media:
communication links



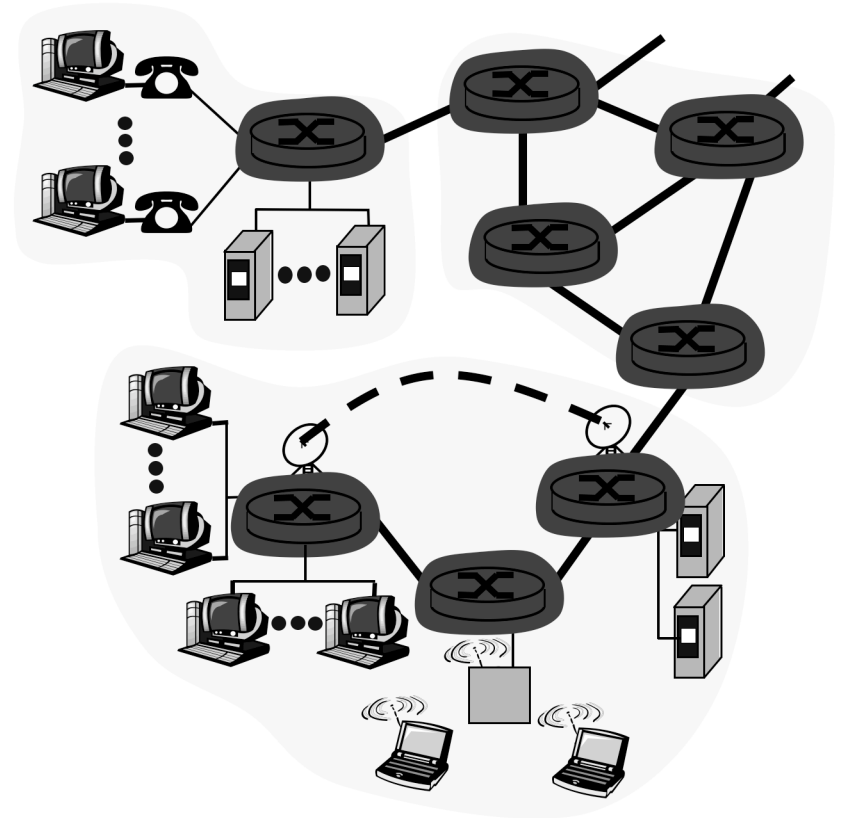
The network edge:

- end systems (hosts):
 - run application programs
 - e.g., WWW, email
 - at “edge of network”
- client/server model
 - client host requests, receives service from server
 - e.g., WWW client (browser)/server, email client/server
- peer-peer model:
 - symmetric host interaction
 - e.g. teleconferencing



The Network Core

- mesh of interconnected routers
- *the* fundamental question: how is data transferred through net?
 - circuit switching: dedicated circuit per call: telephone nets
 - packet-switching: data sent thru net in discrete "chunks" (IP)



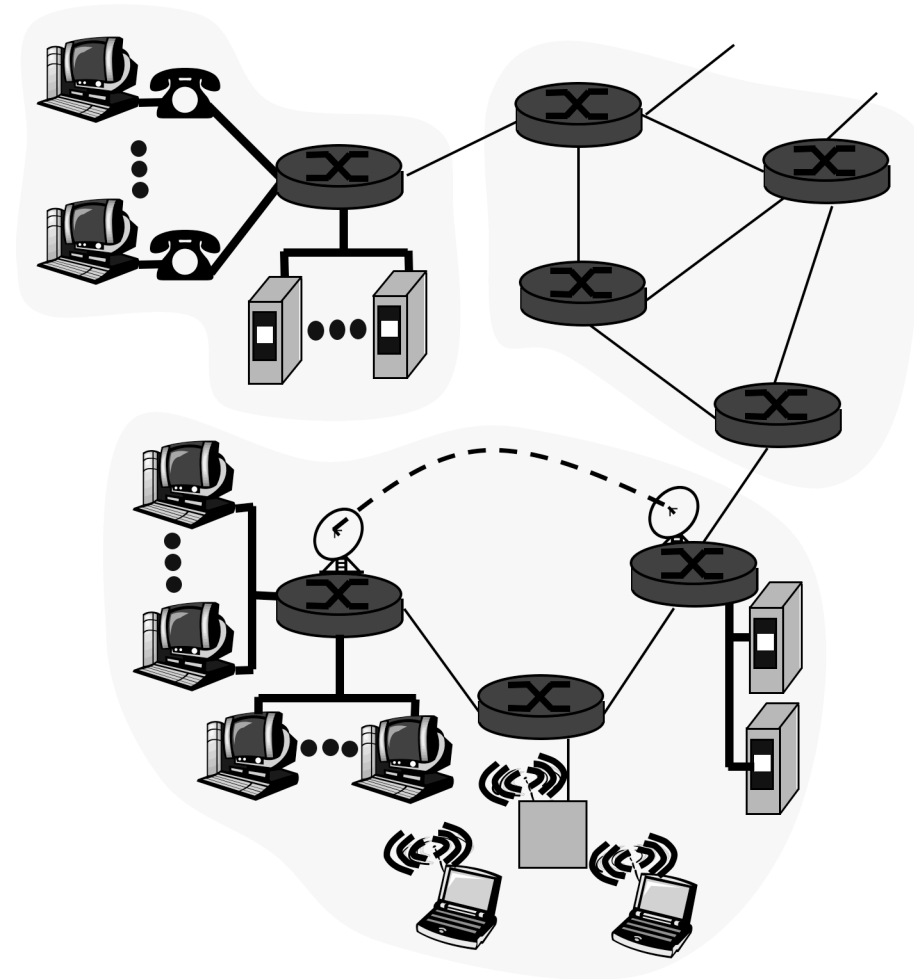
Access networks and physical media

How to connect end systems to edge router?

- residential access nets
- institutional access networks (school, company)
- mobile access networks

Characteristics:

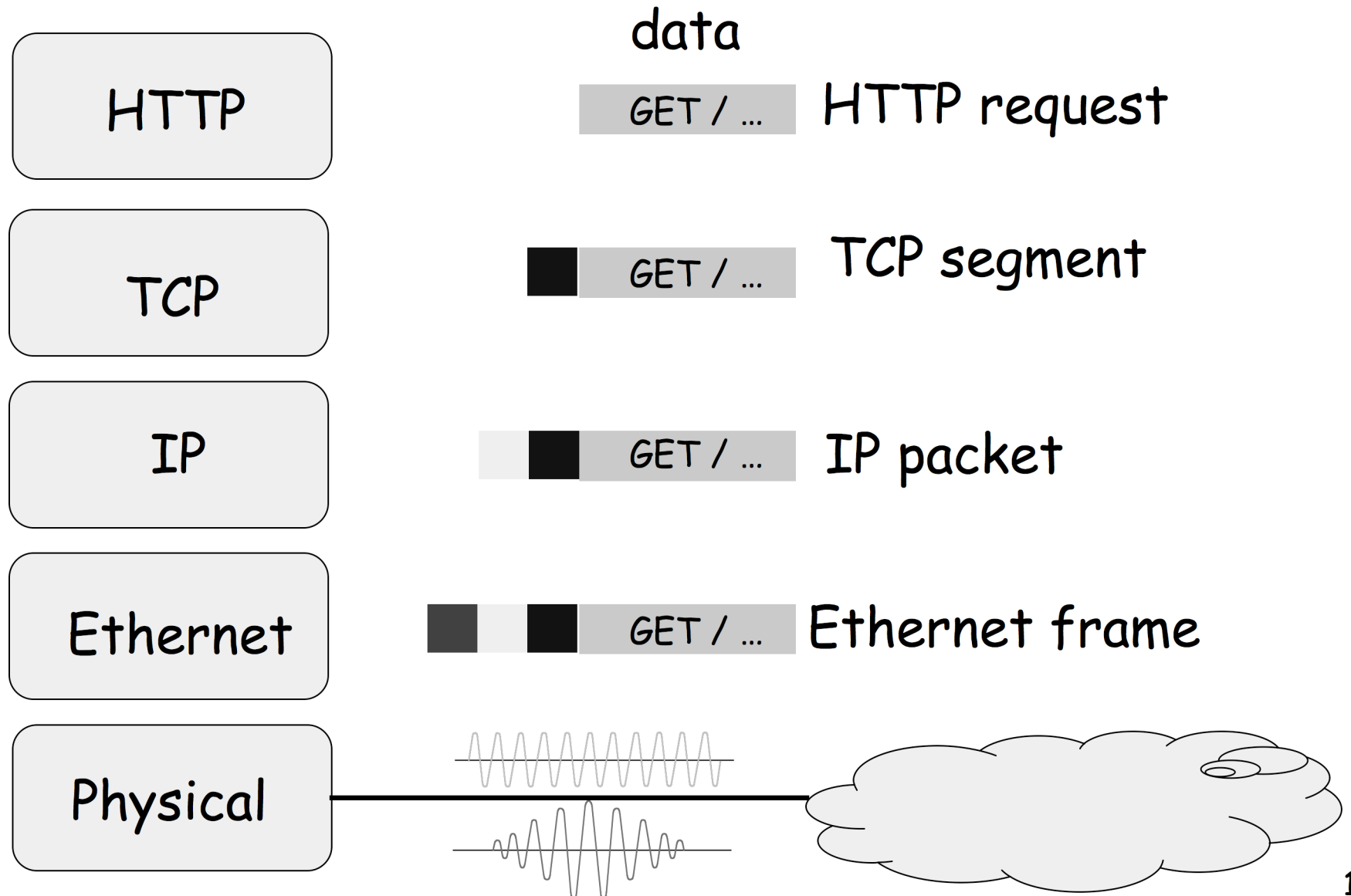
- bandwidth (bits per second) of access network
- shared or dedicated



Internet design principles

- Cerf and Kahn's internetworking principles:
 - minimalism, autonomy - no internal changes required to interconnect networks
 - best effort service model
 - stateless routers
 - decentralized control
- Small number of layers
 - compromise between performance and flexibility
 - thin layers encourage flexibility, but increases overhead
- Define today's Internet architecture

TCP/IP Architecture



Application Layer

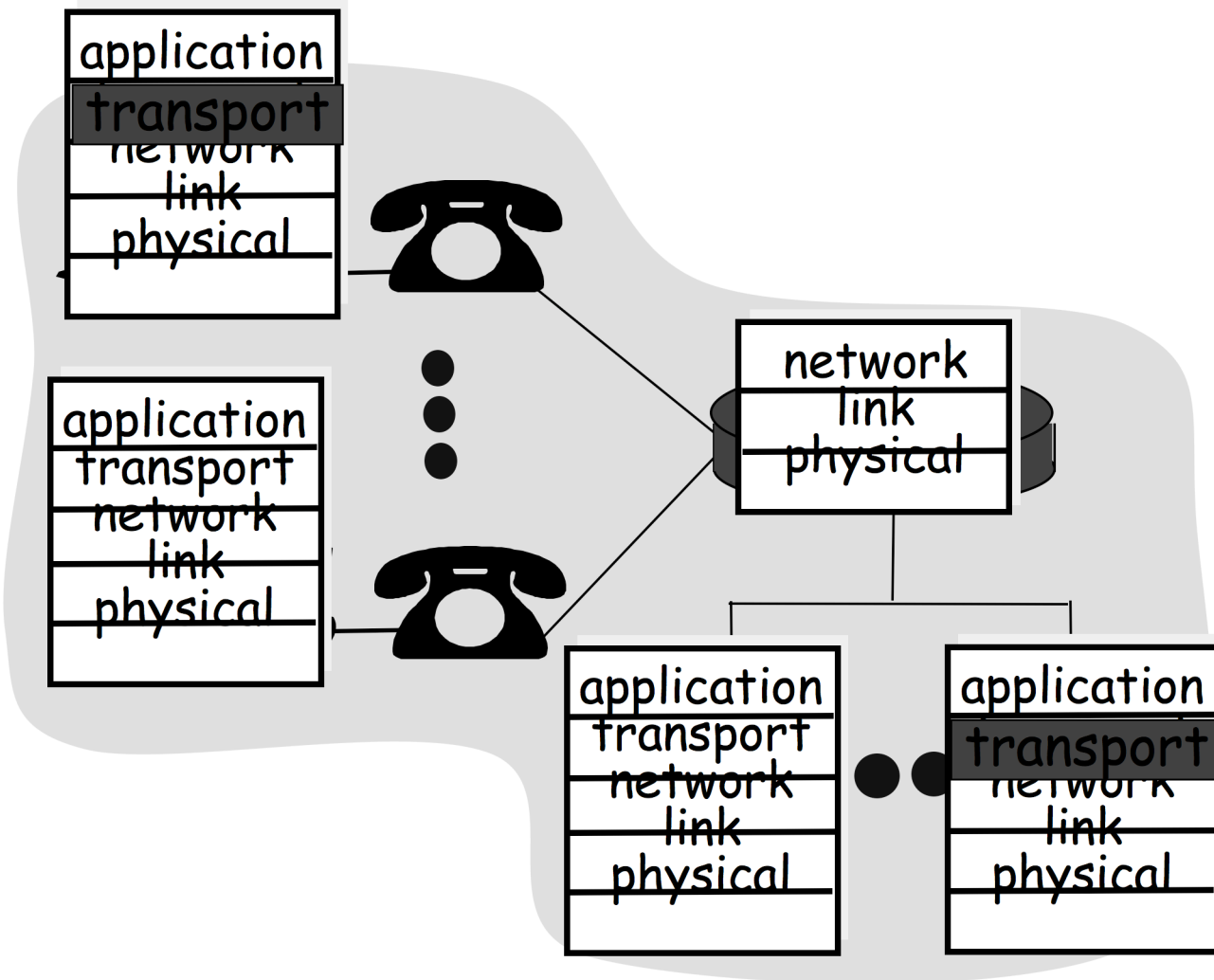
- Application layer supports network application
 - applications that are distributed over the network
 - applications that communicates through the network
- Many known protocols
 - FTP: file transfer
 - SMTP: email protocol
 - HTTP: web protocol
- An application uses UDP or TCP, it is a designer's choice
- Interface with the transport layer
 - use for example the **socket** API: a library of C functions
 - **socket** also means (IP address, port number)

Transport Layer

- Why a transport layer ?
 - **transport layer** = makes network service available to programs
 - is end-to-end only, not in routers
- In TCP/IP there are two transport protocols
 - UDP (user datagram protocol)
 - unreliable
 - offers a datagram service to the application (unit of information is a message)
 - TCP (transmission control protocol)
 - reliable
 - offers a stream service (unit of information is a byte)

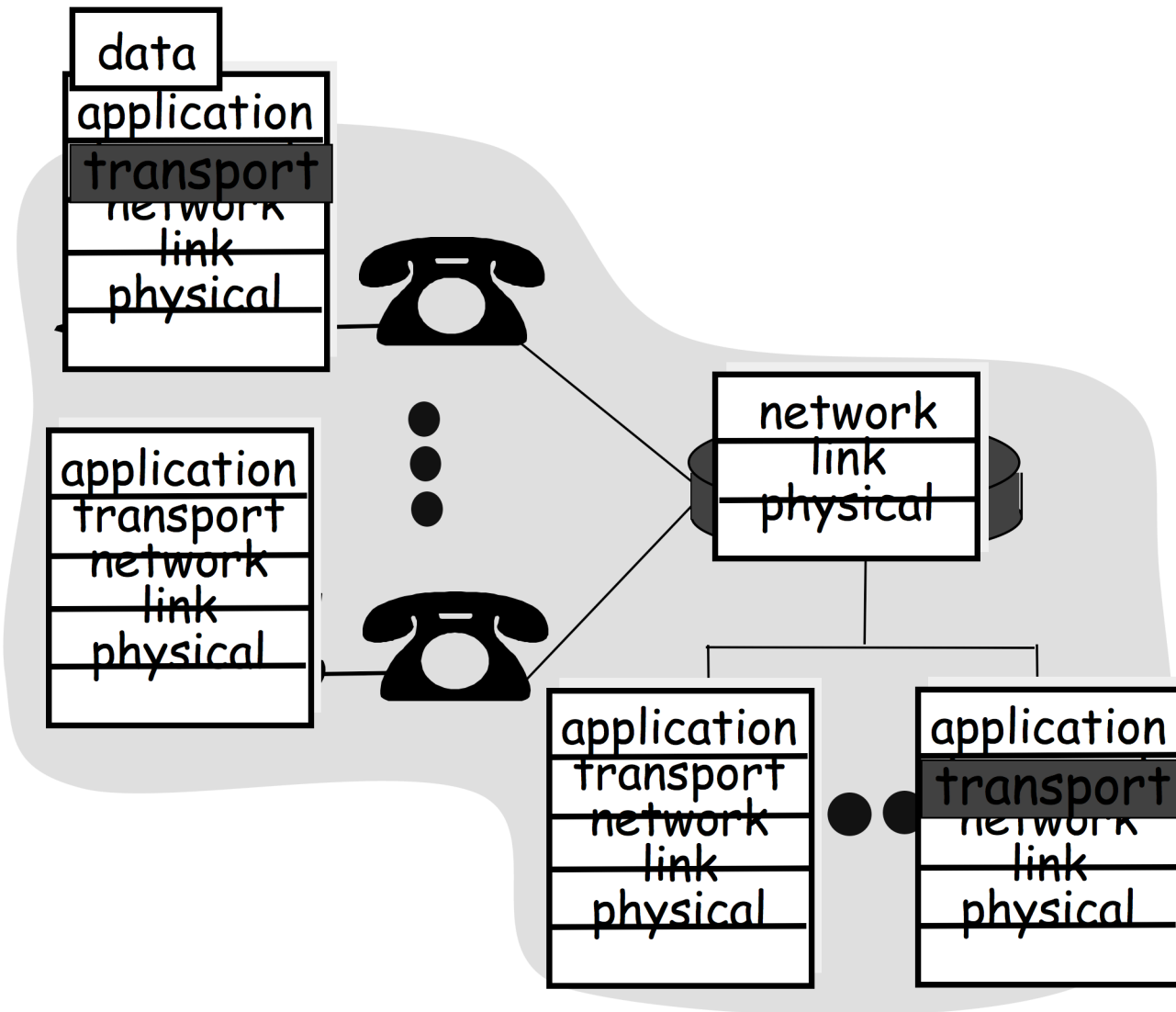
Layering: *logical* communication

- E.g.: transport
- take data from app
- add addressing, reliability check info to form “datagram”
- send datagram to peer
- wait for peer to ack receipt
- analogy: post office



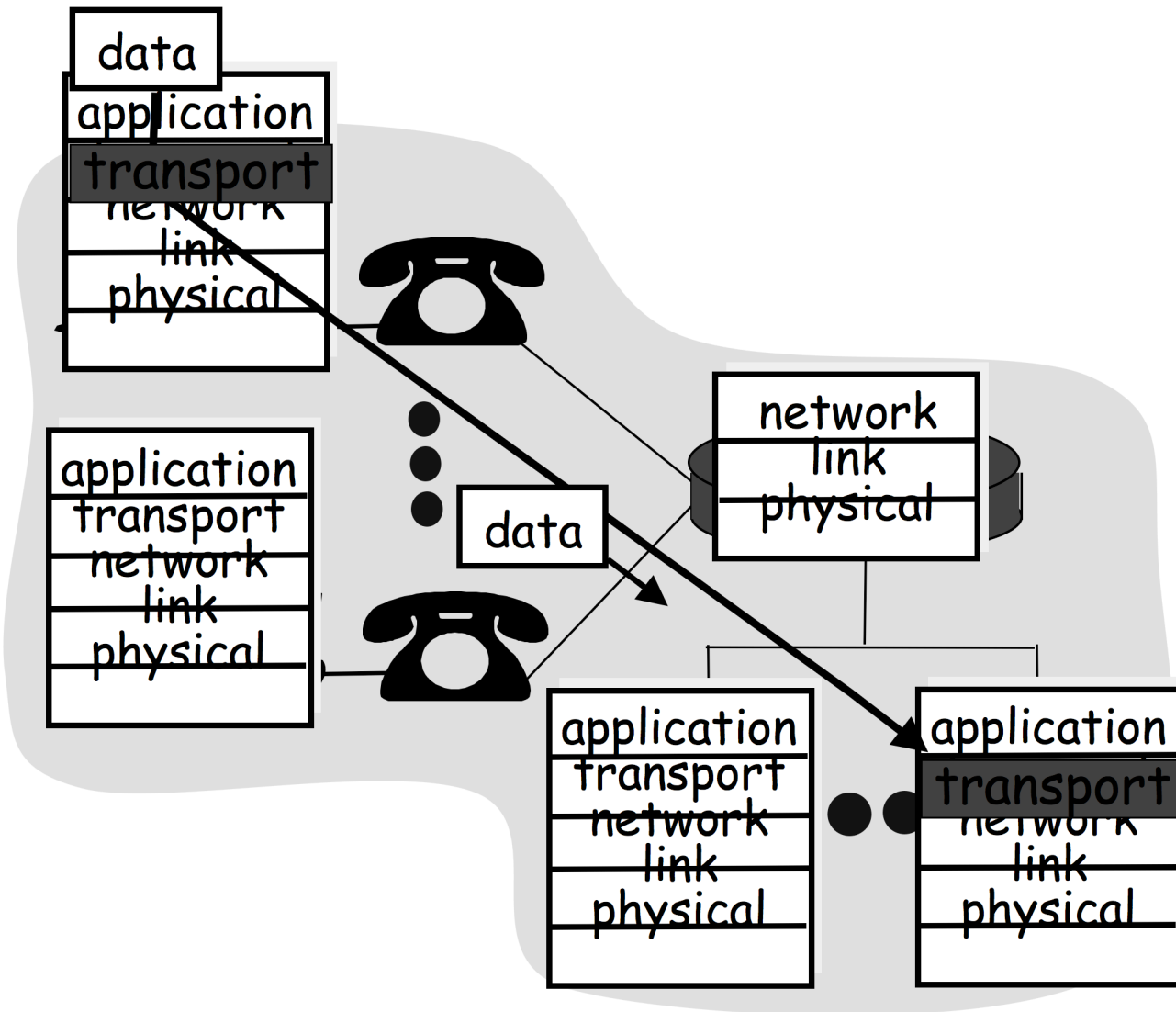
Layering: *logical* communication

- E.g.: transport
- take data from app
- add addressing, reliability check info to form “datagram”
- send datagram to peer
- wait for peer to ack receipt
- analogy: post office



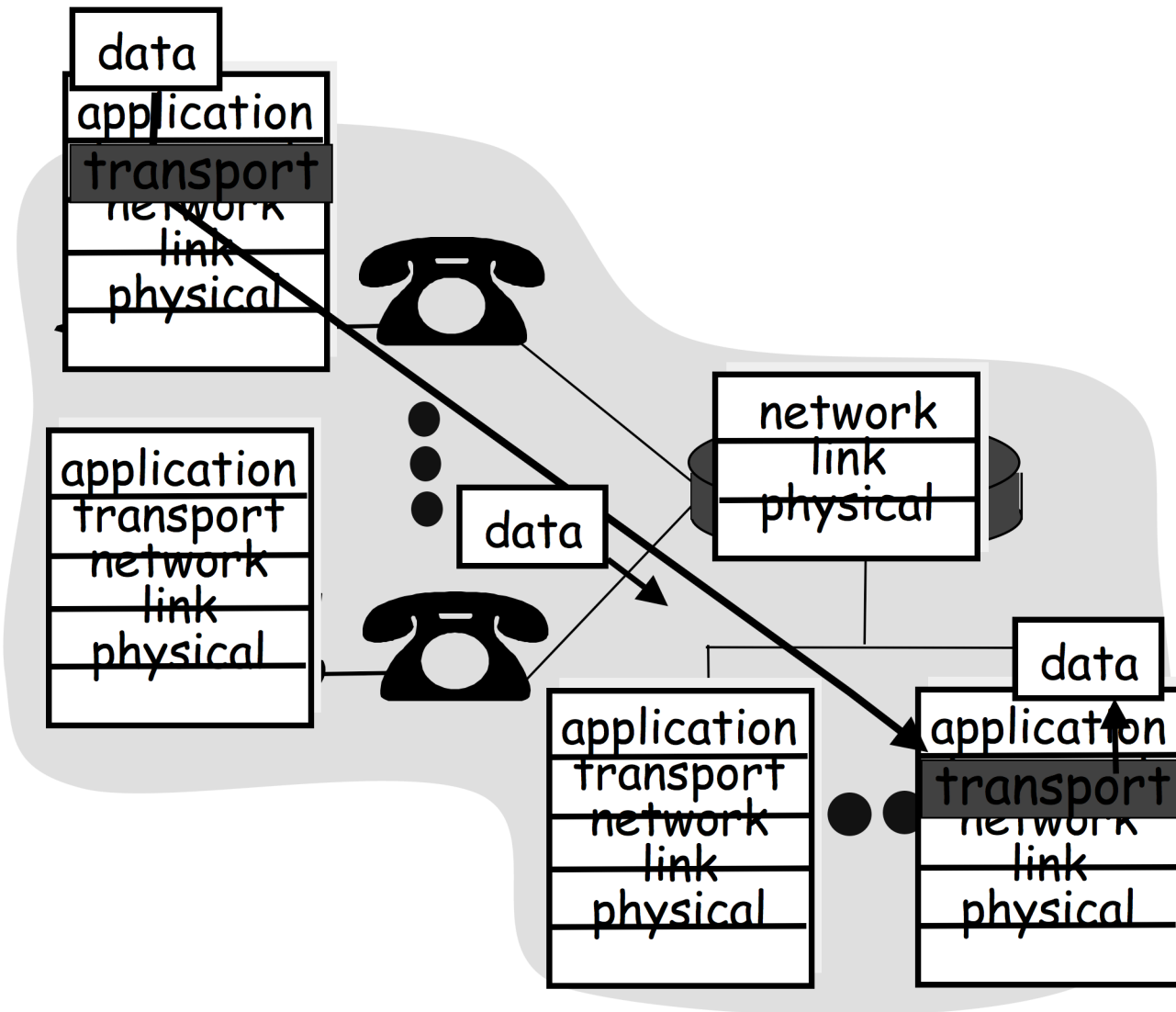
Layering: *logical* communication

- E.g.: transport
- take data from app
- add addressing, reliability check info to form “datagram”
- send datagram to peer
- wait for peer to ack receipt
- analogy: post office



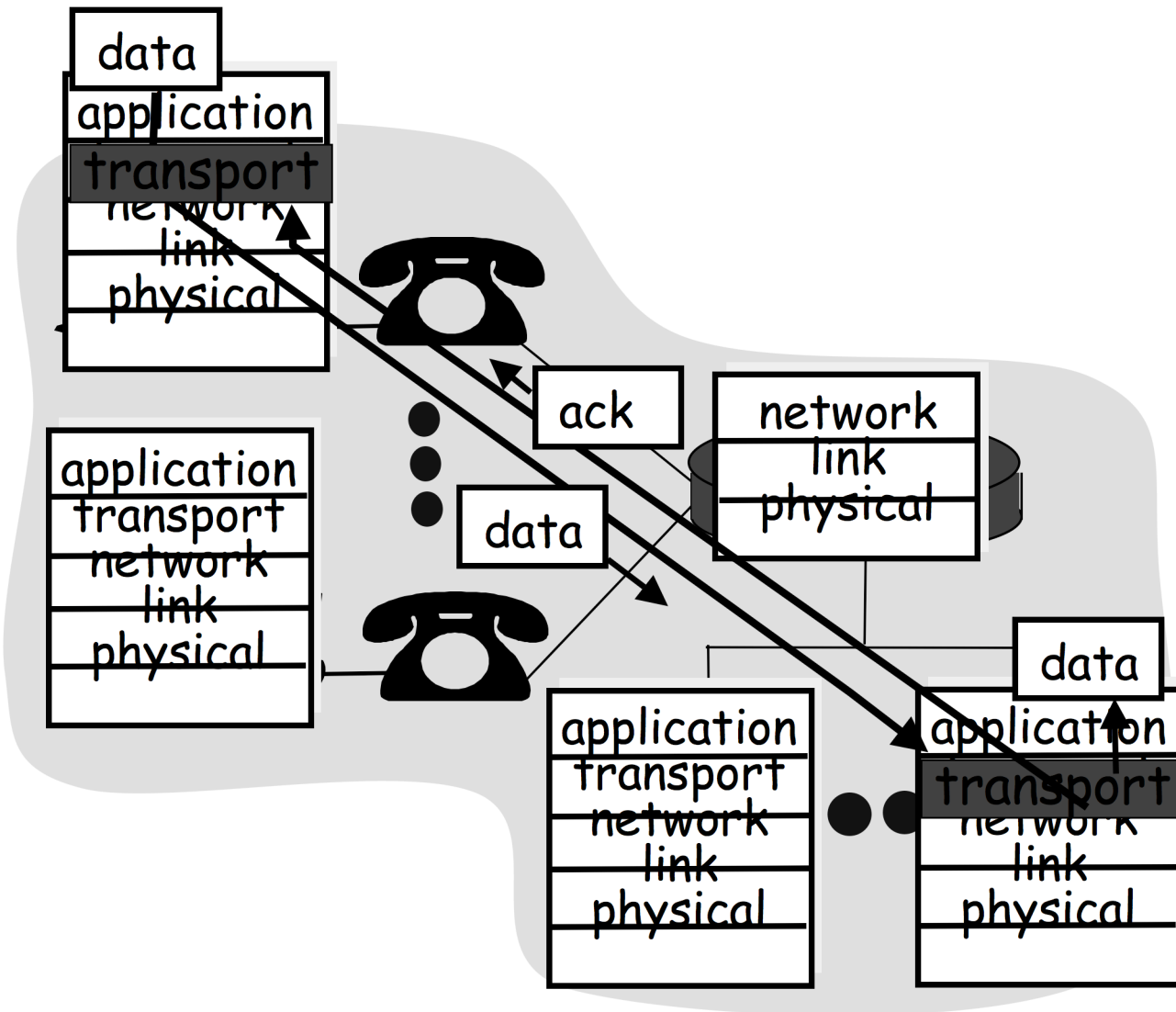
Layering: *logical* communication

- E.g.: transport
- take data from app
- add addressing, reliability check info to form “datagram”
- send datagram to peer
- wait for peer to ack receipt
- analogy: post office

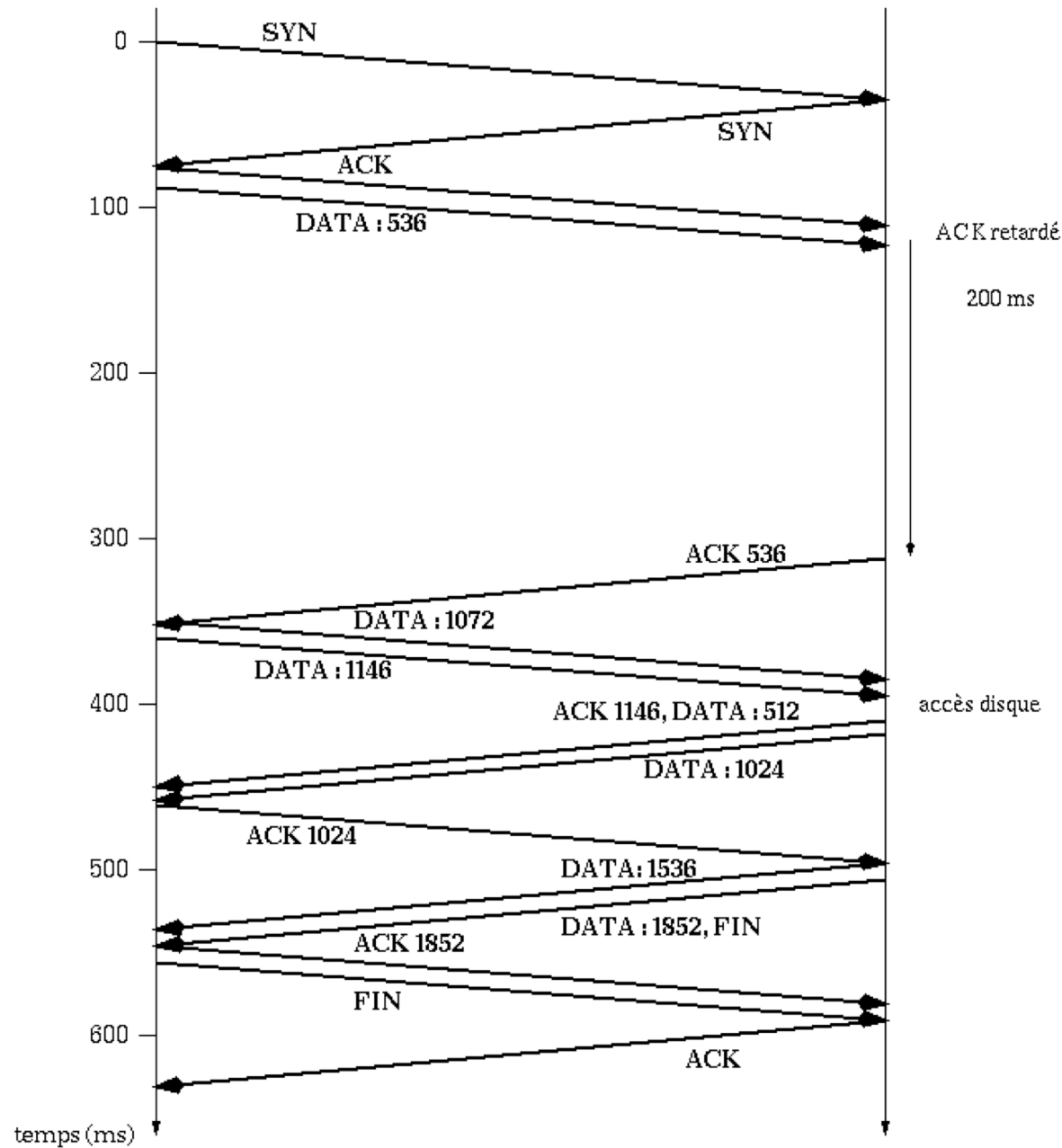


Layering: *logical* communication

- E.g.: transport
- take data from app
- add addressing, reliability check info to form “datagram”
- send datagram to peer
- wait for peer to ack receipt
- analogy: post office



TCP



HTTP Request : 1146 octets

HTTP Reply : 1852 octets

Taille maximale de segment : 536 octets

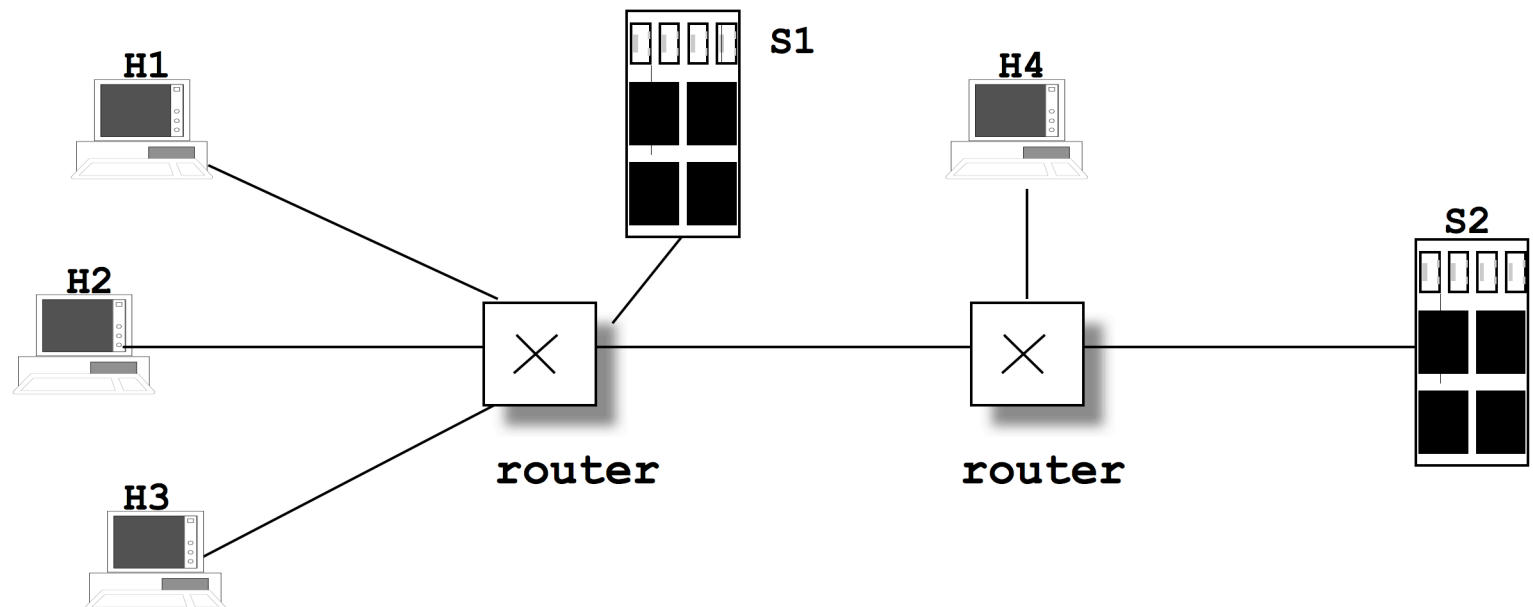
Temps aller-retour (RTT) : ~70 ms
(North Carolina → Illinois)

Problèmes :

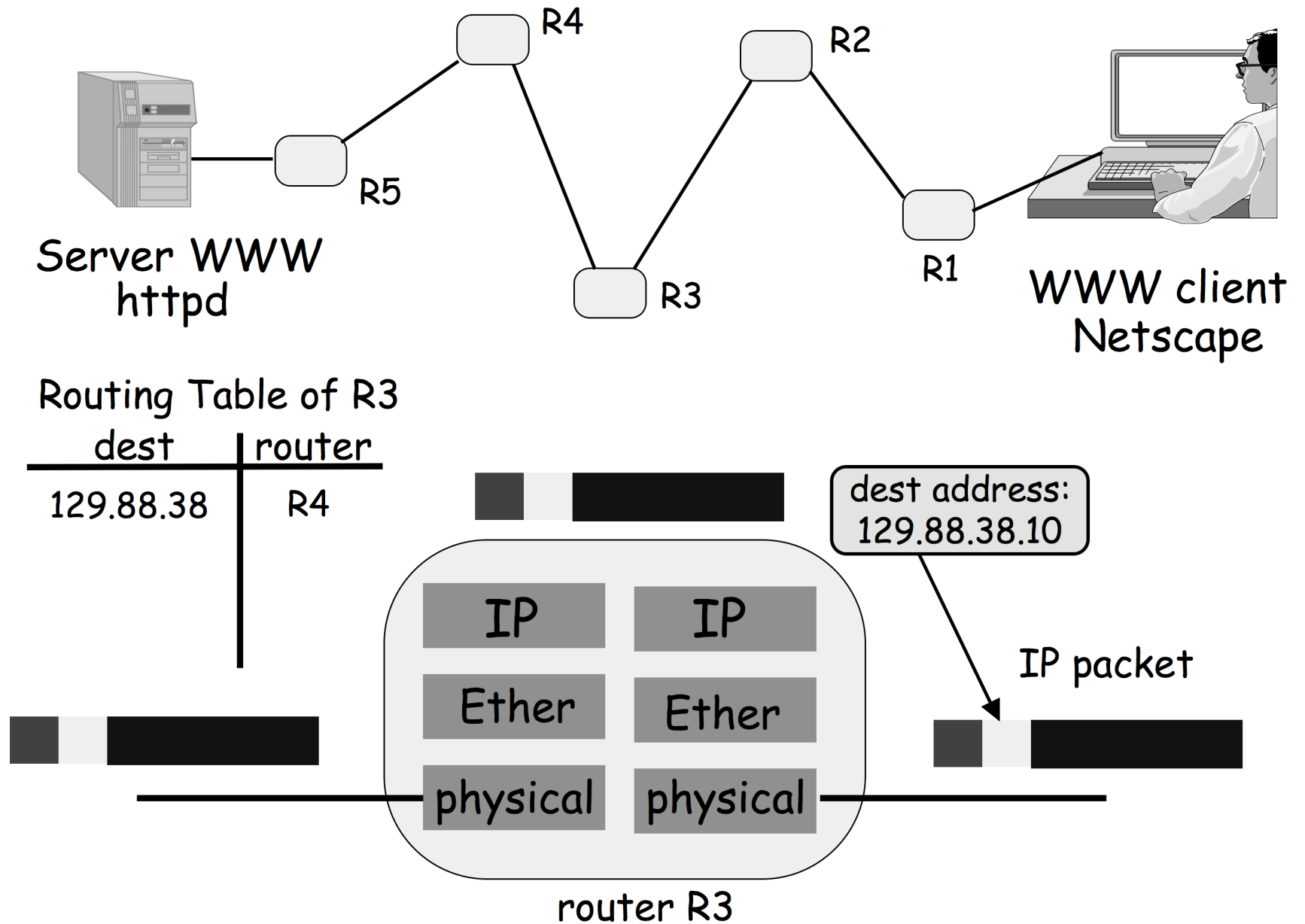
- établissement de connexion
- démarrage lent (Slow Start)
- ACK retardé
- temps aller-retour

Network Layer

- Set of functions required to transfer packets end-to-end (from host to host)
 - hosts are not directly connected - need for intermediate systems
 - examples: IP, Appletalk, IPX
- Intermediate systems
 - routers: forward packets to the final destination
 - interconnection devices

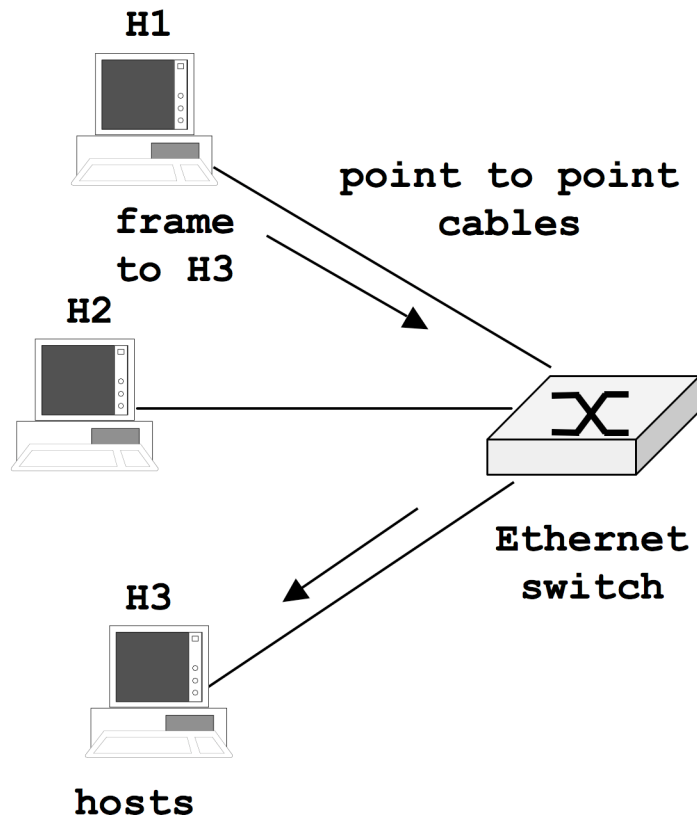


IP



Physical Layer

Data Link Layer

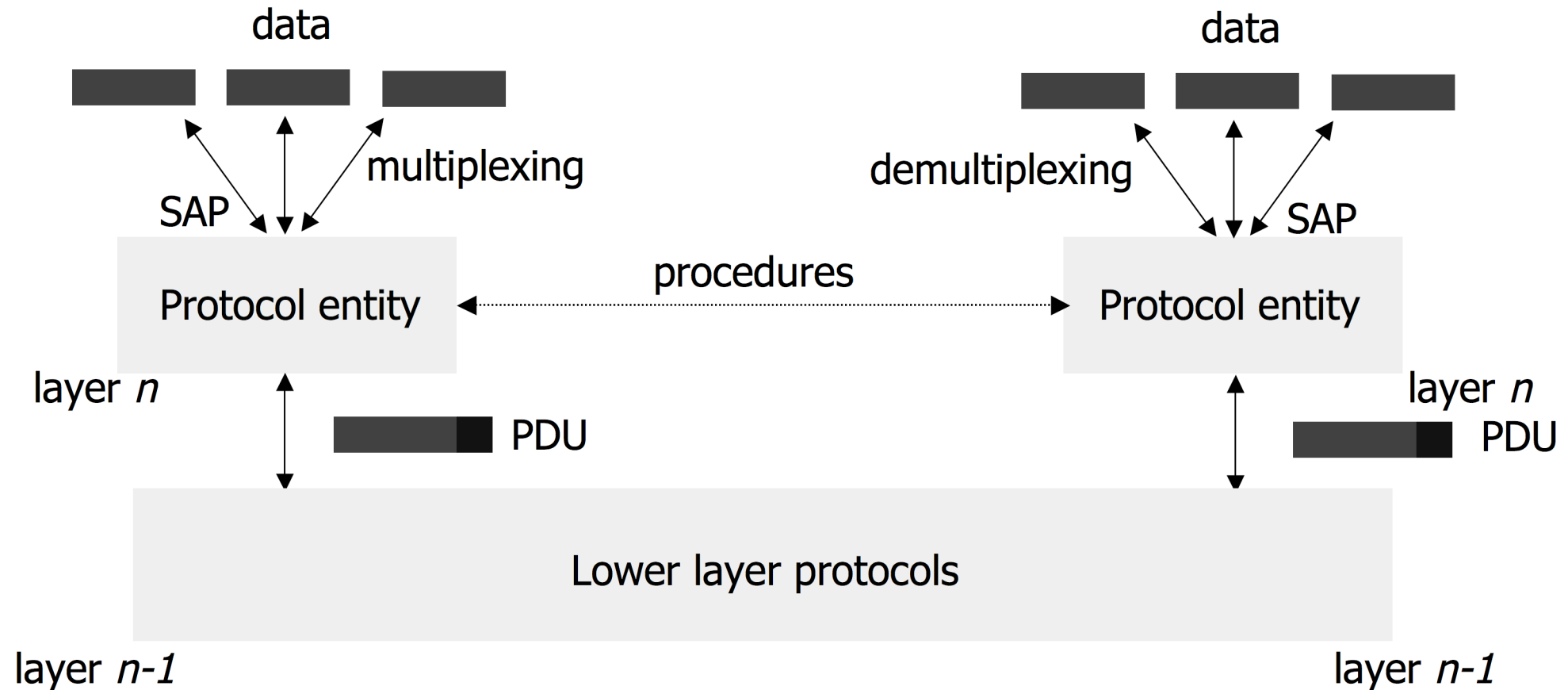


- Physical transmission = **Physical** function
 - bits $\leftrightarrow$ electrical / optical signal
 - transmit individual bits over the cable: modulation, encoding
- Frame transmission = **Data Link** function
 - bits $\leftrightarrow$ frames
 - bit error detection
 - packet boundaries
 - in some cases: error correction b retransmission (802.11)
- Modems, xDSL, LANs

Protocol architectures

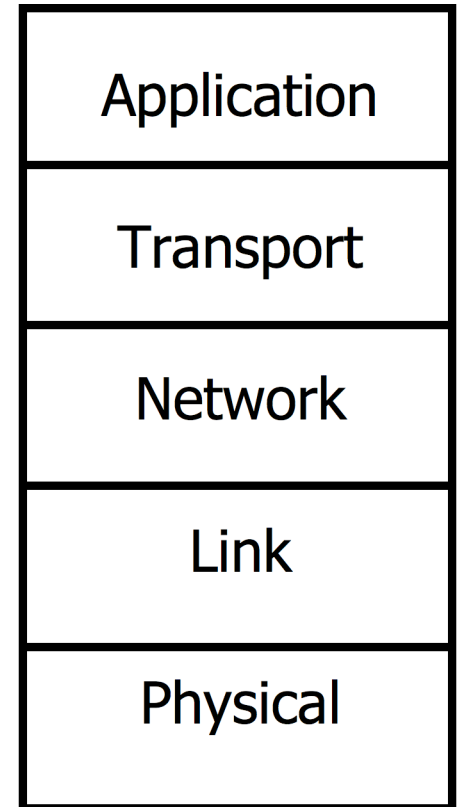
- Protocol entity
 - provides a set of services, eg.
 - *connect, send*
 - data multiplexing/demultiplexing
 - construction/analysis of PDUs
 - execution of procedures
- Protocol unit (PDU)
 - header: control functions
 - opaque data
- Procedures
 - actions to perform protocol functions: e.g. lost packet retransmission

Protocol architecture

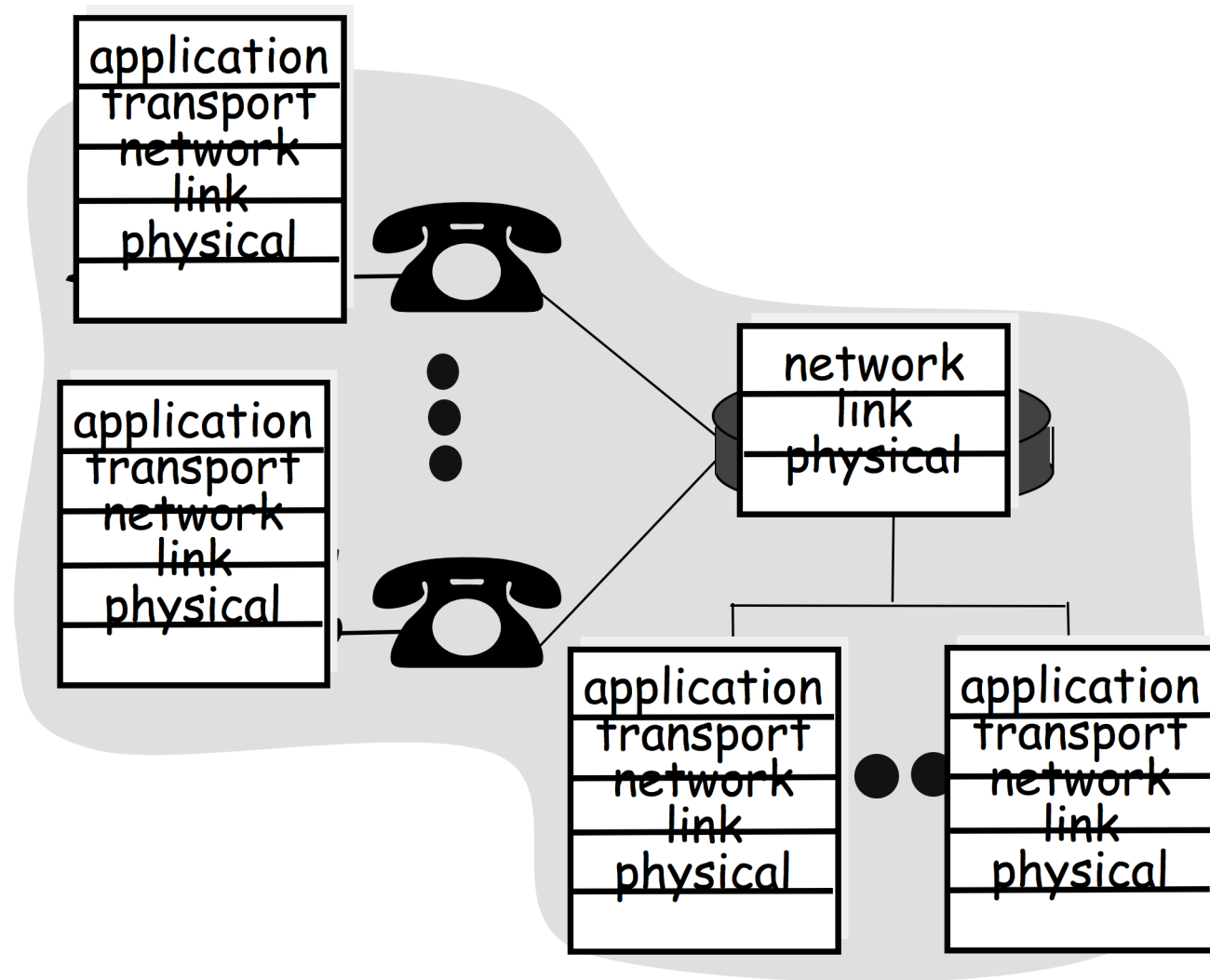


Internet protocol stack

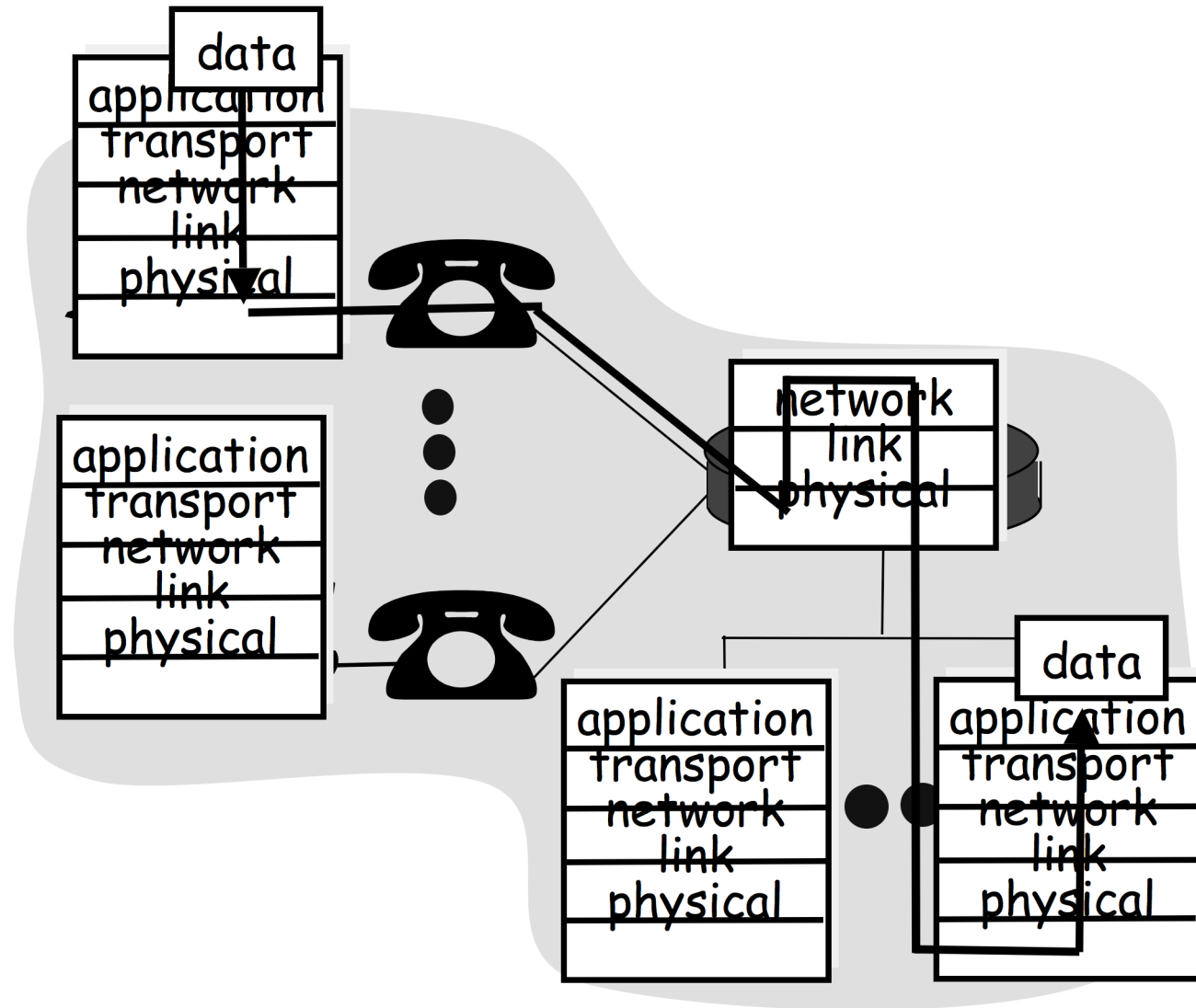
- **Application:** supporting network applications
 - FTP, SMTP, HTTP, OSPF, RIP
- **Transport:** host-host data transfer
 - TCP, UDP
- **Network:** routing of datagrams from source to destination
 - IP
- **Link:** data transfer between neighboring network elements
 - PPP, Ethernet
- **Physical:** bits “on the wire”



Layering: physical communication



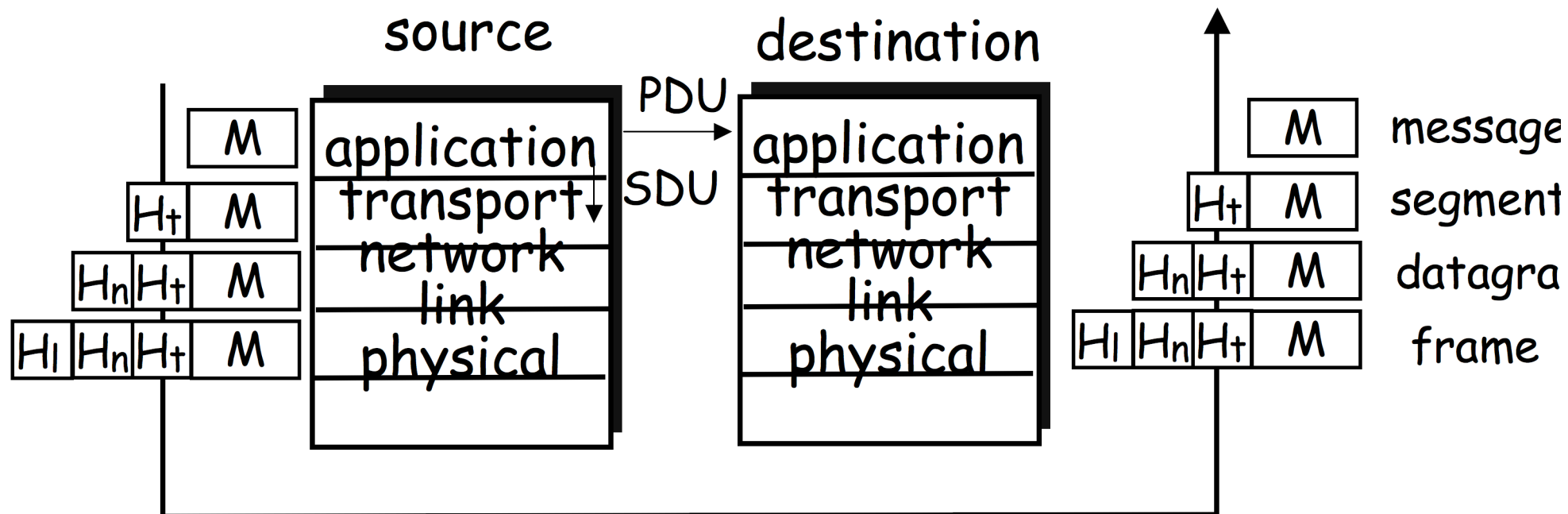
Layering: physical communication



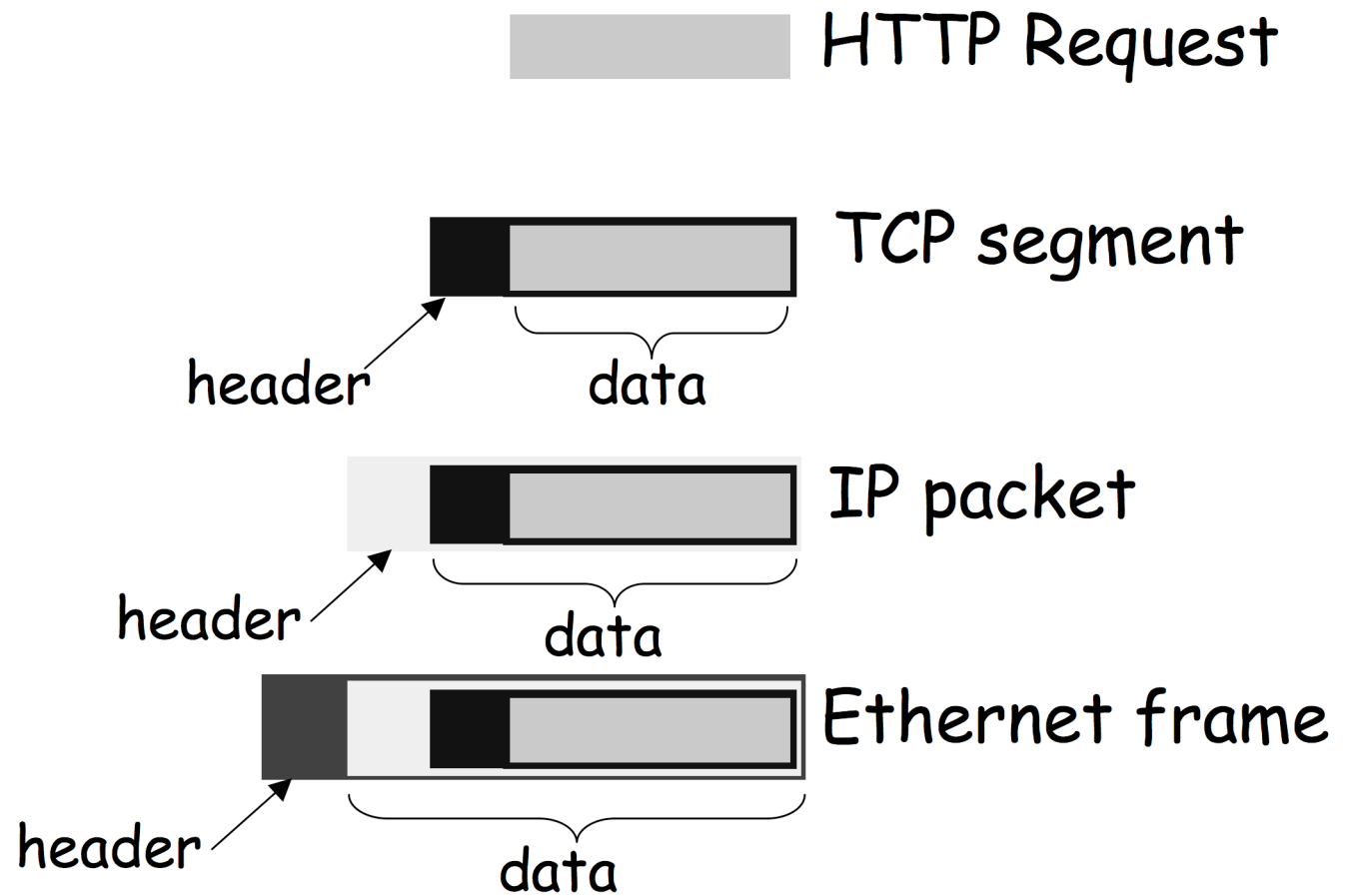
Protocol layering and data

Each layer takes data from above

- adds header information to create new data unit
- passes new data unit to layer below



Encapsulation



OSI ISO Model

Application

- Common functions

Presentation

- Interchangeable formats

Session

- Organizing dialog

Transport

- Reliable transmission

Network

- Forwarding in the network

Data link

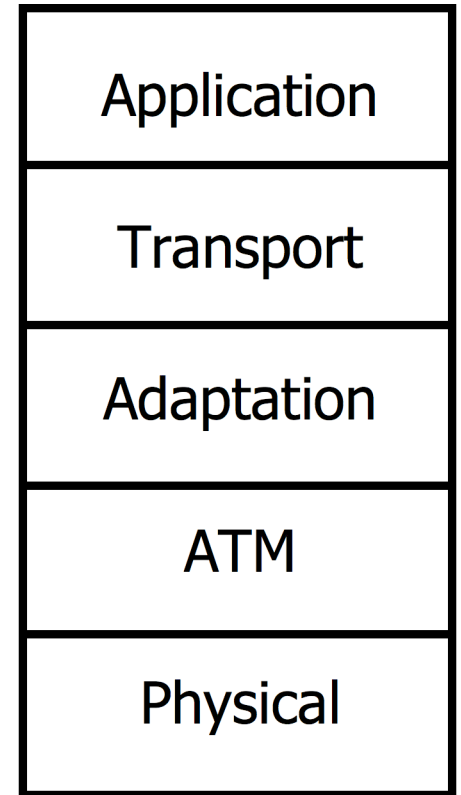
- Transmission between two nodes

Physical

- Signal transmission

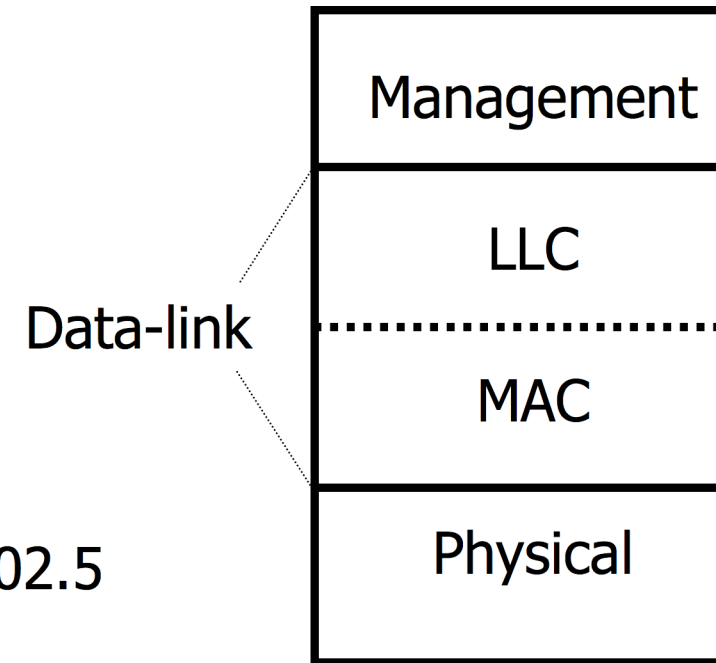
ATM protocol stack

- **Application:** native applications, other protocols
 - LAN Emulation, IP, Signaling
- **Transport:** host-host data transfer
 - SSCOP
- **Adaptation:** adapt the ATM layer to different types of applications
 - circuit emulation, real-time data
 - AAL5 suitable for IP traffic
- **ATM:** cell switching over virtual circuits
- **Physical:** bits “on the wire”, usually fiber

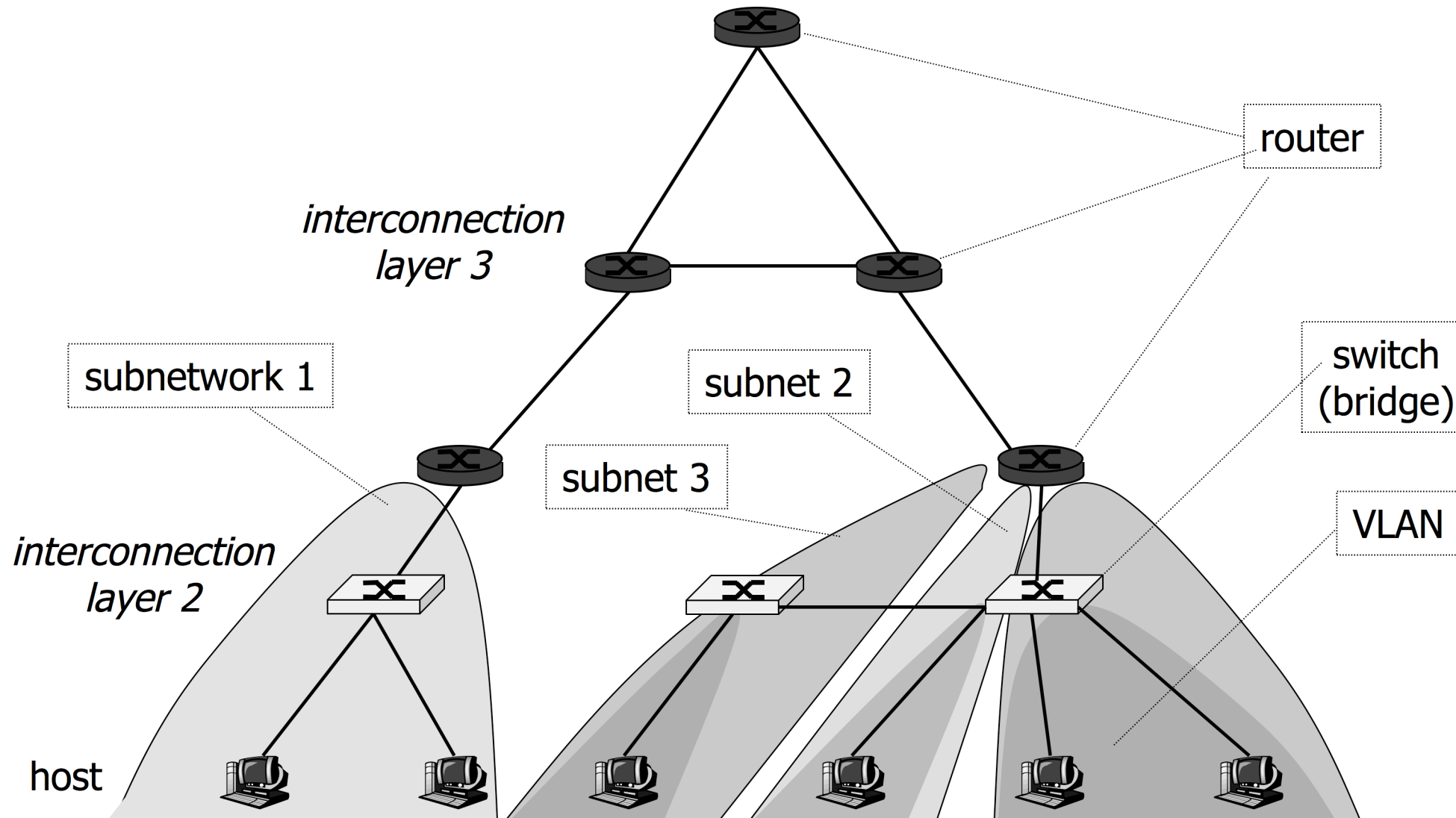


LAN stack

- **Management:** e.g. construct forwarding tables
 - SNAP: Spanning Tree protocol
- **LLC:** multiplex different protocols
 - IP, IPX, SNAP
- **MAC:** medium access
 - 802.3 (Ethernet), 802.4 (Token Ring), 802.5 (Token Bus), 802.11 (Wi-Fi)
- **Physical:** bits “on the wire”



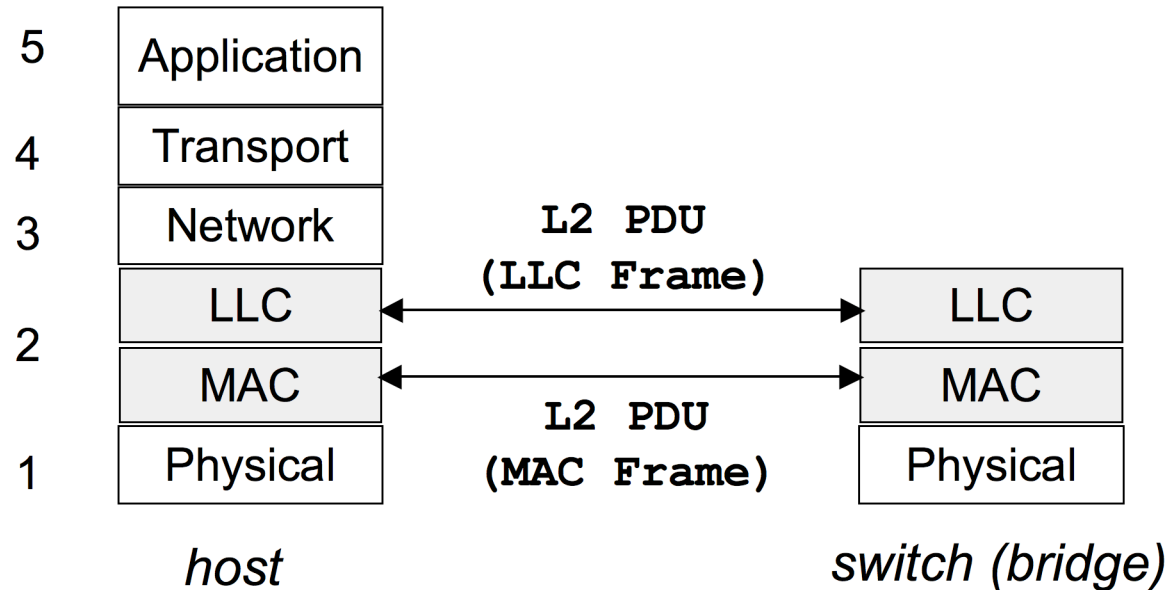
Interconnection structure - layer 3



Interconnection at layer 2

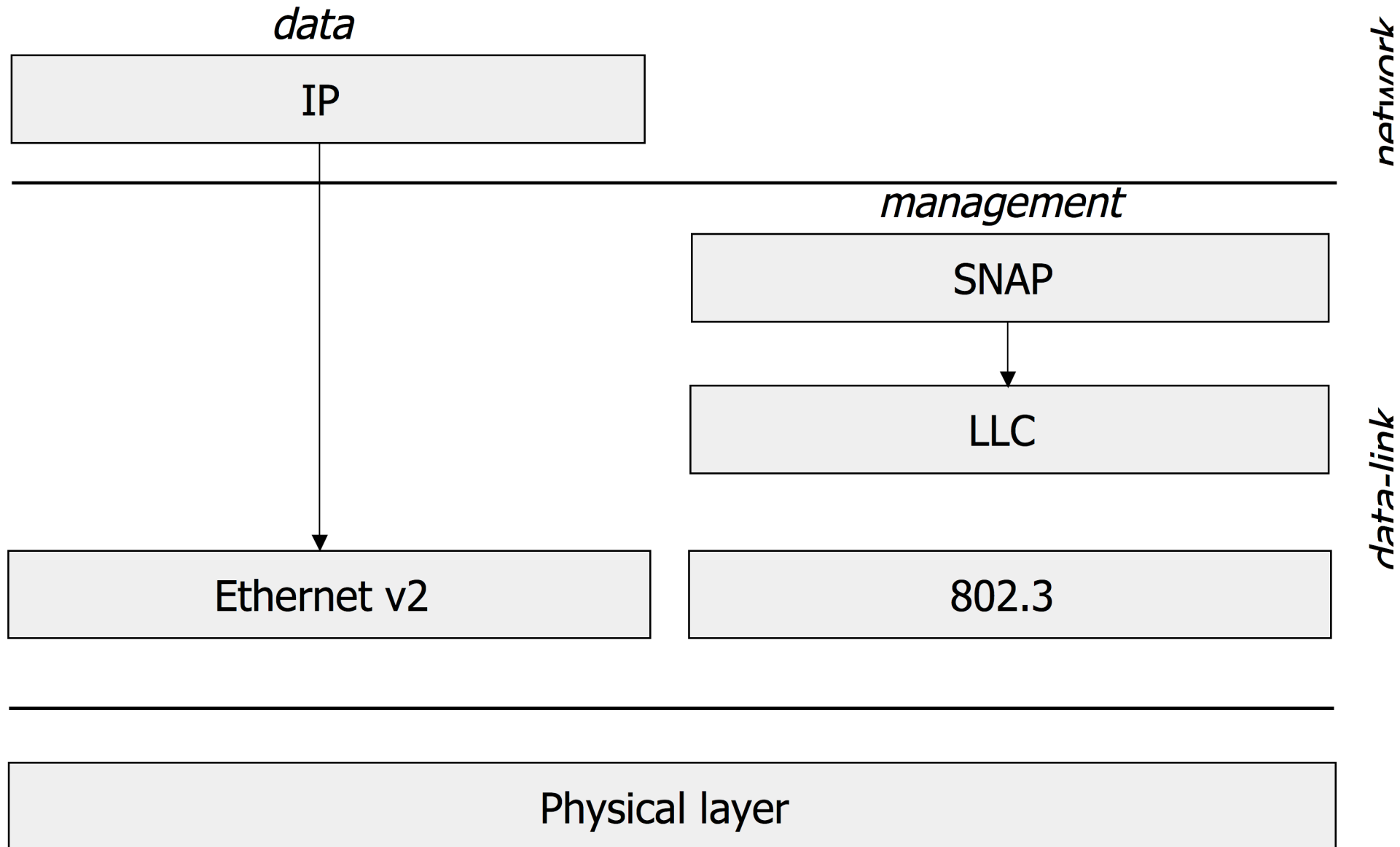
- Switches (bridges)
 - interconnect hosts
 - logically separate groups of hosts (VLANs)
 - managed by one entity
- Type of the network
 - broadcast
- Forwarding based on MAC address
 - flat address space
 - forwarding tables: one entry per host
 - works if no loops
 - careful management
 - Spanning Tree protocol
 - not scalable

Protocol architecture



- Switches are layer 2 intermediate systems
- Transparent forwarding
- Management protocols (Spanning Tree, VLAN)

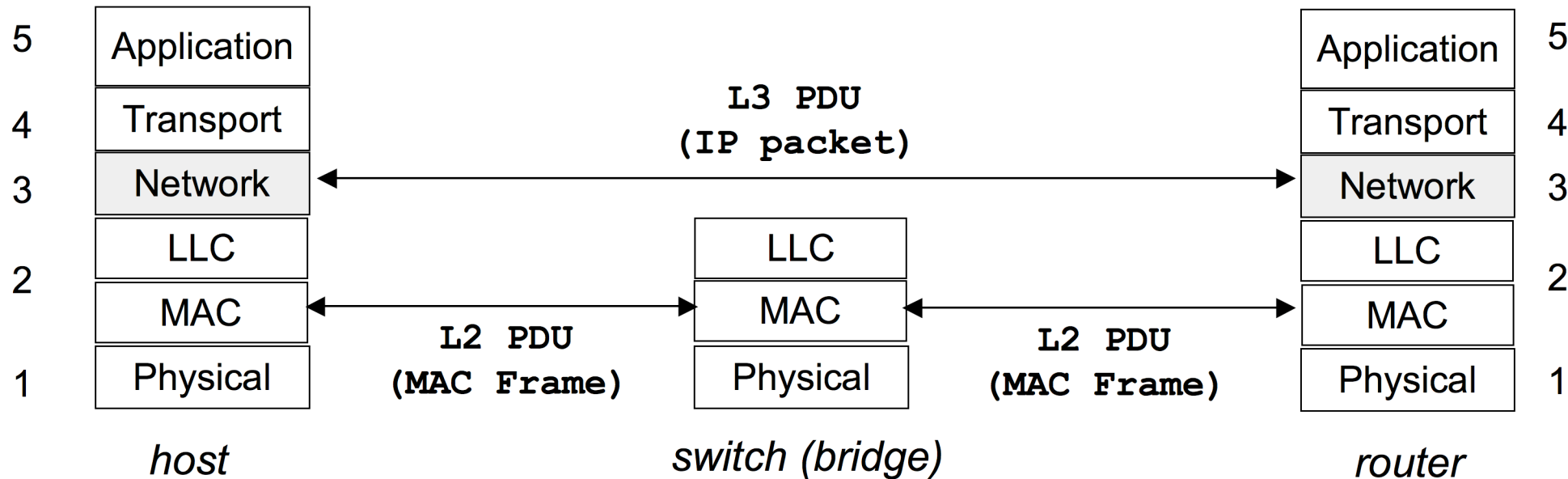
Protocols



Interconnection at layer 3

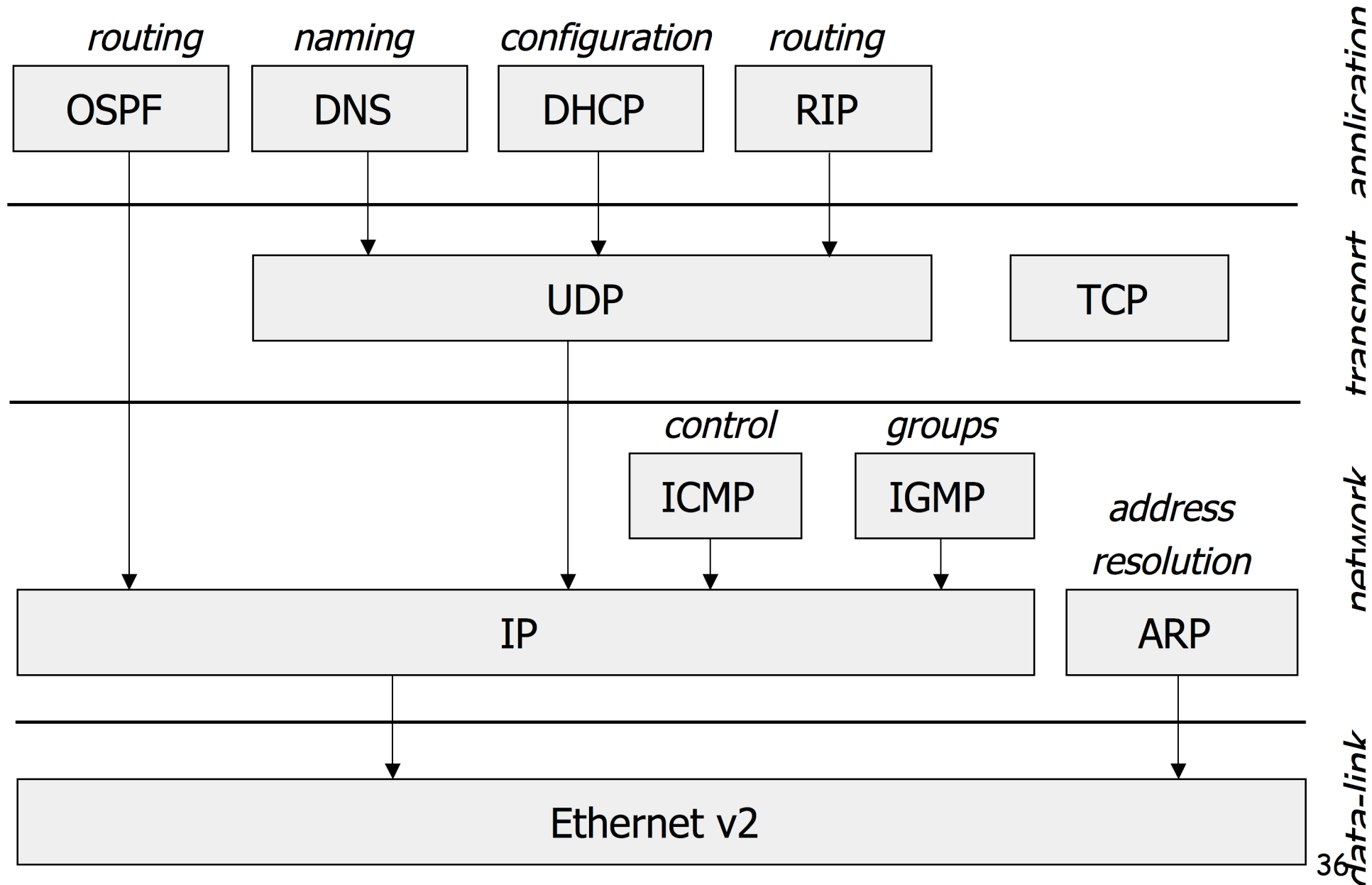
- Routers
 - interconnect subnetworks
 - logically separate groups of hosts
 - managed by one entity
- Forwarding based on IP address
 - structured address space
 - routing tables: aggregation of entries
 - works if no loops - routing protocols (IGP - Internal Routing Protocols)
 - scalable inside one administrative domain

Protocol architecture

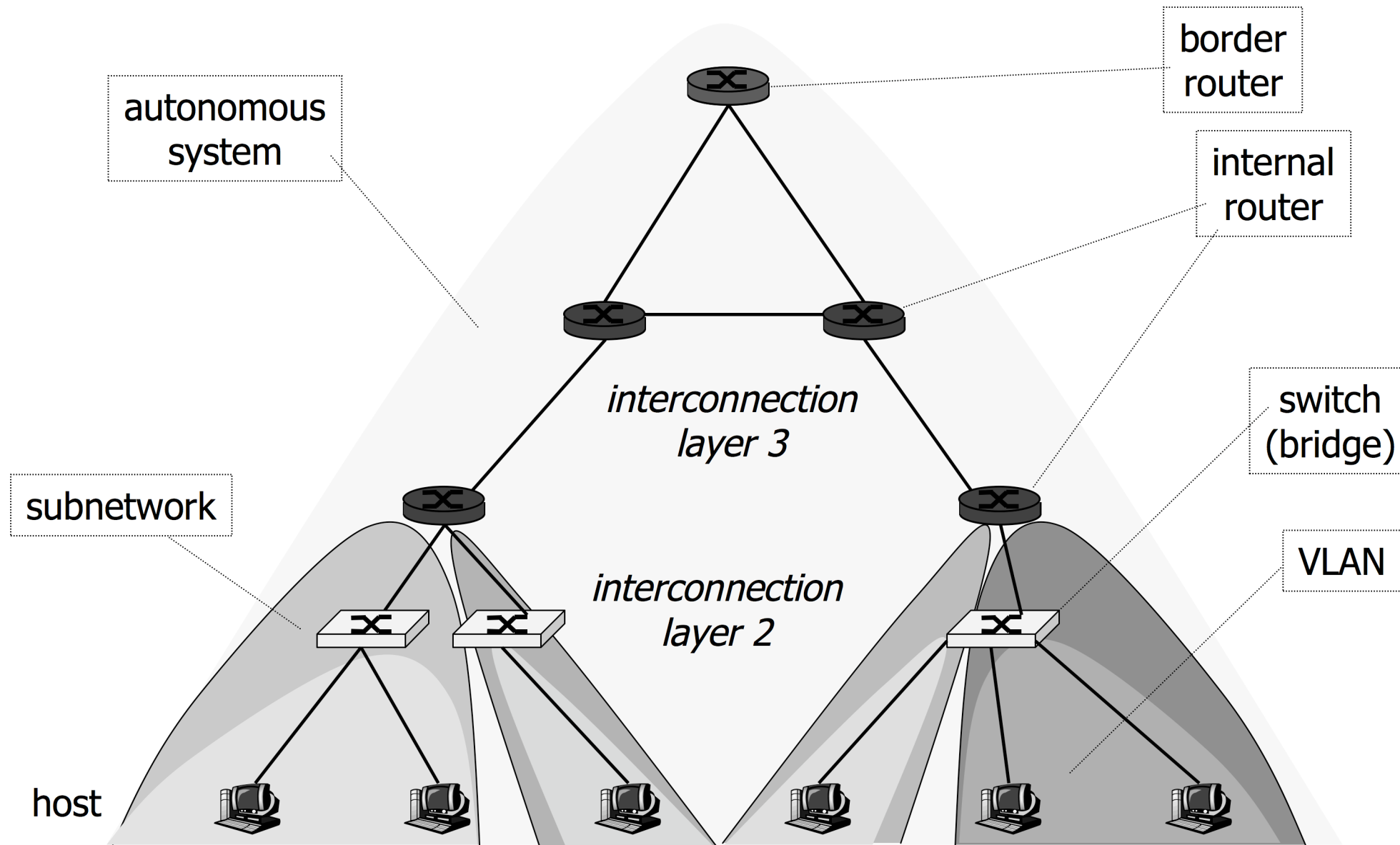


- Routers are layer 3 intermediate systems
- Explicit forwarding
 - host has to know the address of the first router
- Management protocols (control, routing, configuration)

Protocols



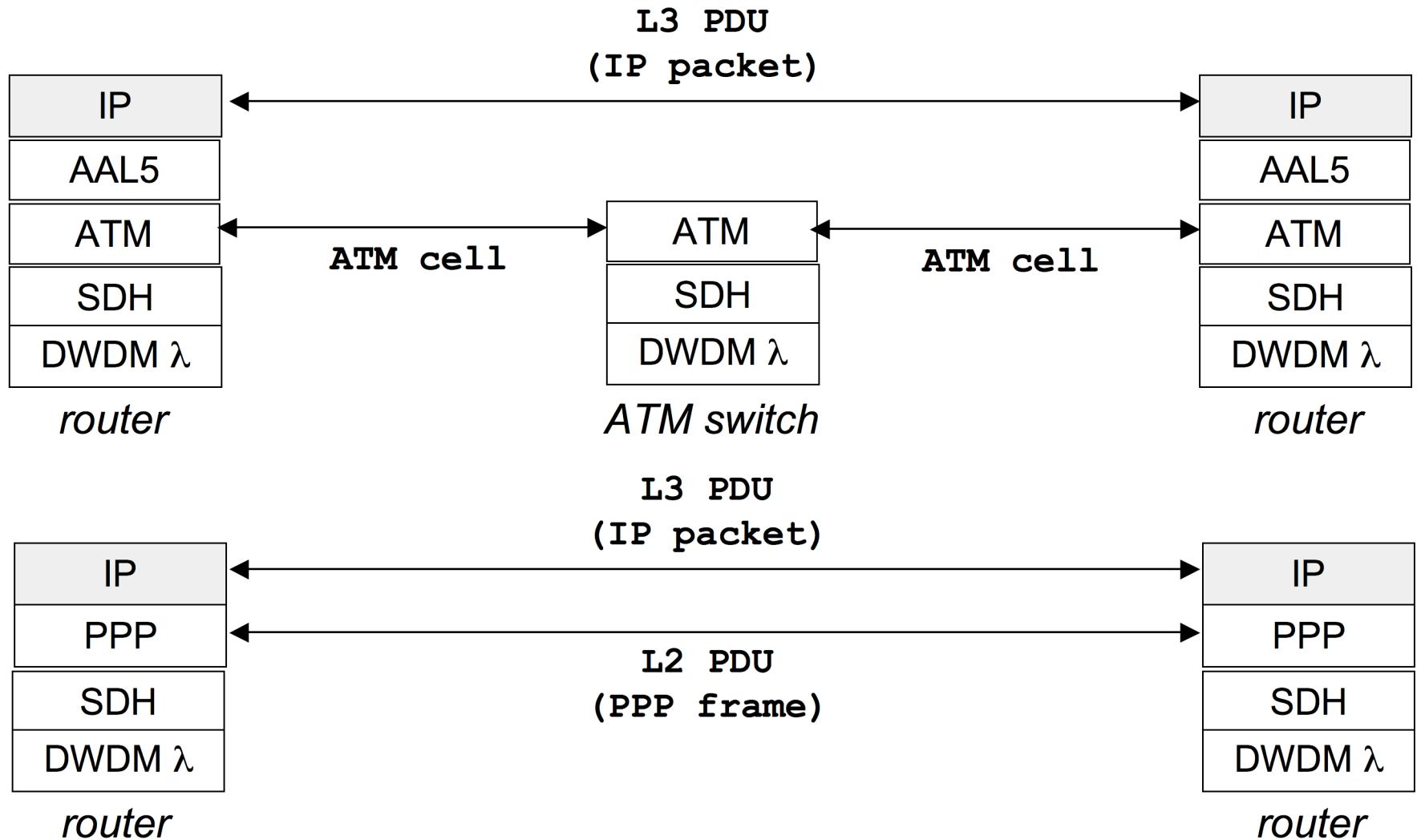
Autonomous systems



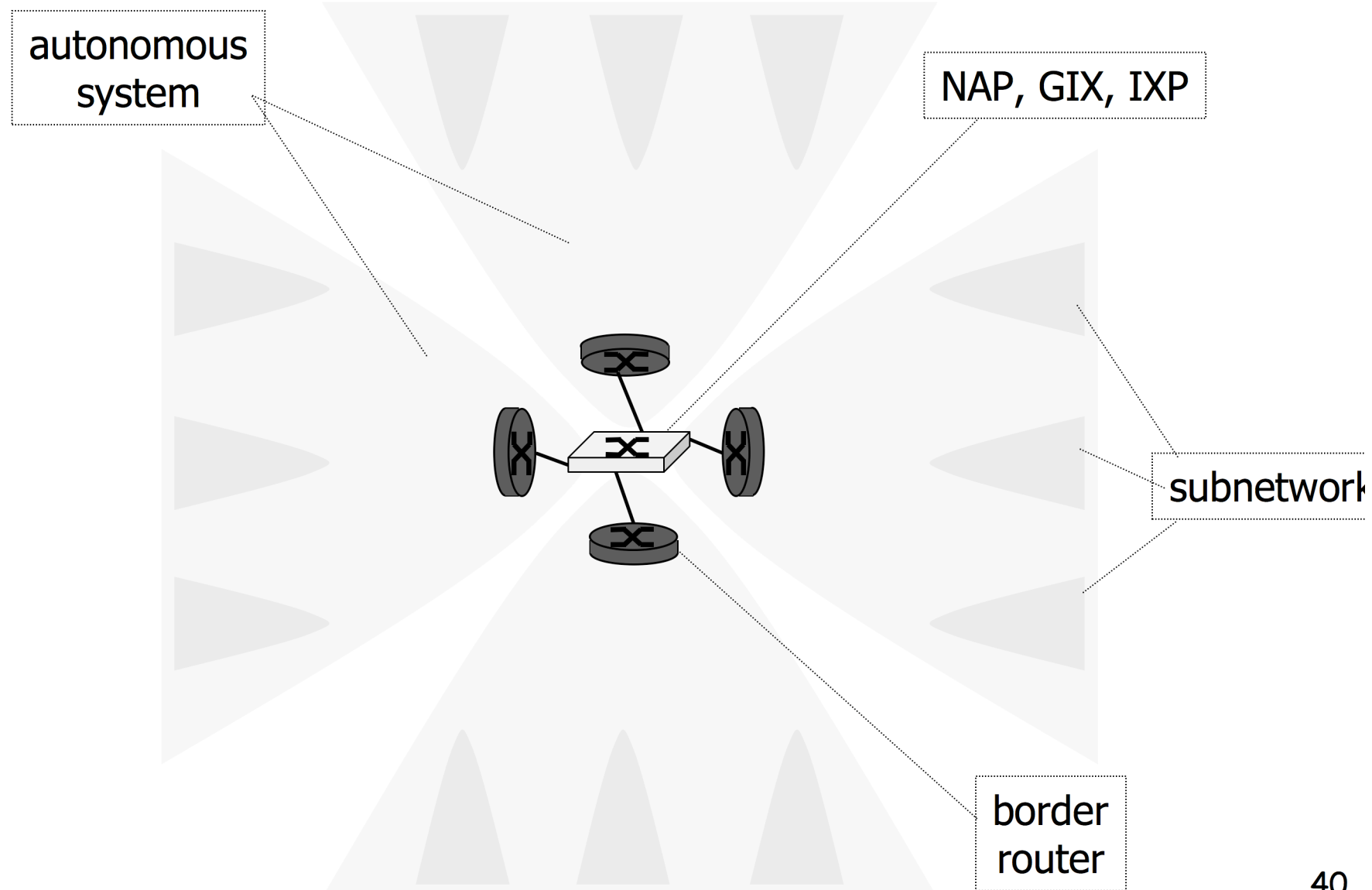
Overlaid stacks? Long-haul links

- Fiber at physical layer (SONET/SDH)
 - Dense Wave Division Multiplexing (DWDM)
 - one color of the light λ
- Different technologies
 - ATM
 - Frame Relay
 - POS (Packet over SONET/SDH)
- Type of the network
 - NBMA (Non Broadcast Multiple Access) or point-to-point
- Complex protocol hierarchies
 - IP over ATM

Protocol architecture



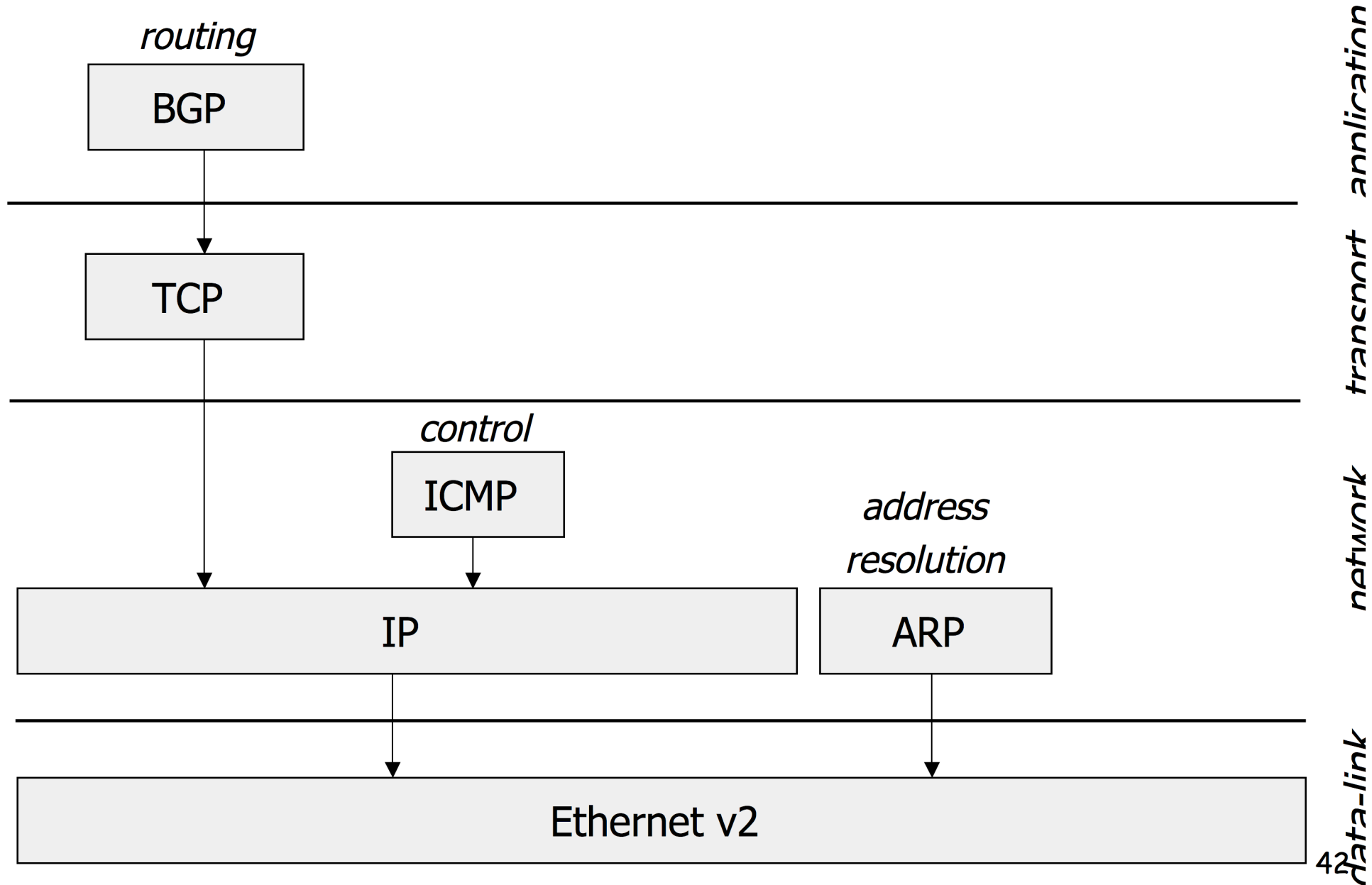
Internet



Interconnection of AS

- Border routers
 - interconnect AS
- NAP or GIX, or IXP
 - exchange of traffic - peering
- Route construction
 - based on the path through a series of AS
 - based on administrative policies
 - routing tables: aggregation of entries
 - works if no loops and at least one route - routing protocols (EGP - External Routing Protocols)

Protocols



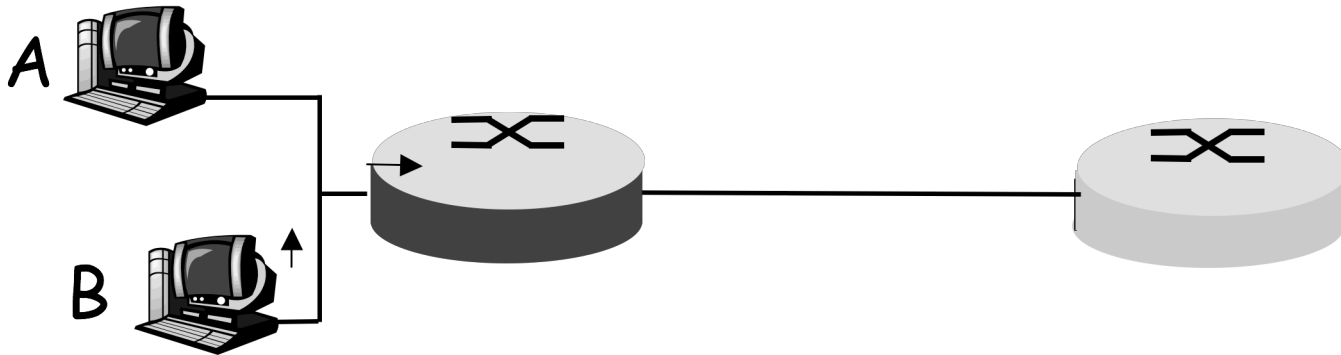
Performance

- Bit Rate (débit binaire) of a transmission system
 - bandwidth, throughput
 - number of bits transmitted per time unit
 - units: b/s or bps, kb/s = 1000 b/s, Mb/s = 10e+06 b/s, Gb/s=10e+09 b/s
 - OC3/STM1 - 155 Mb/s, OC12/STM4 - 622 Mb/s, and OC48/STM-16 - 2.5 Gb/s, OC192/STM-48 10 Gb/s
- Latency or Delay
 - time interval between the beginning of a transmission and the end of the reception
 - RTT - Round-Trip Time

Delay in packet-switched networks

packets experience delay
on end-to-end path

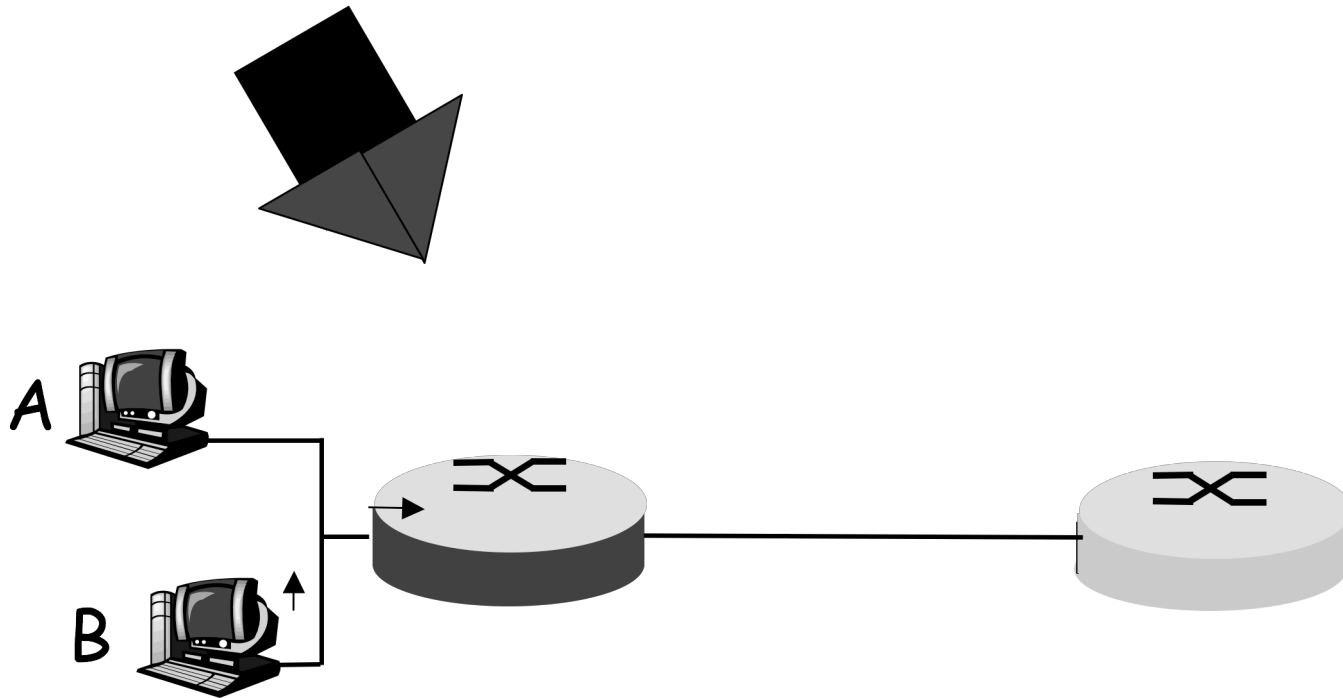
- four sources of delay at each hop



Delay in packet-switched networks

packets experience delay
on end-to-end path

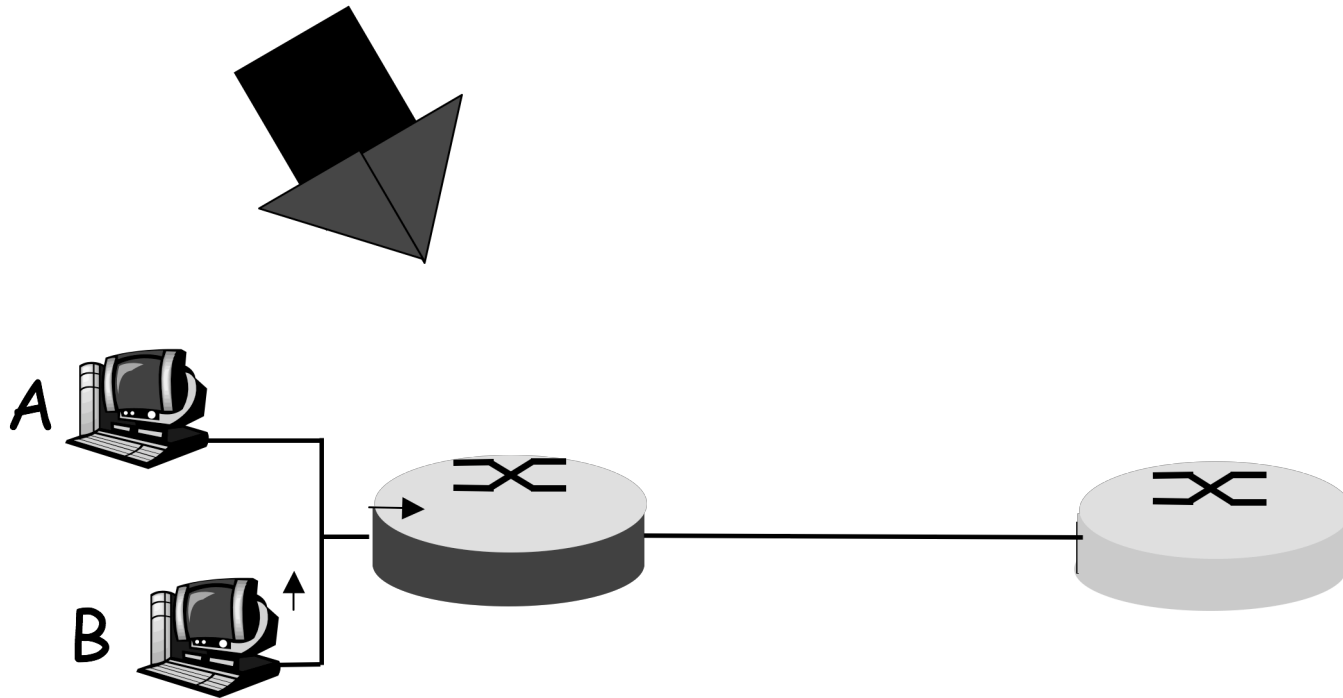
- four sources of delay at each hop



Delay in packet-switched networks

packets experience delay
on end-to-end path

- four sources of delay at each hop
- nodal processing:
 - check bit errors
 - determine output link

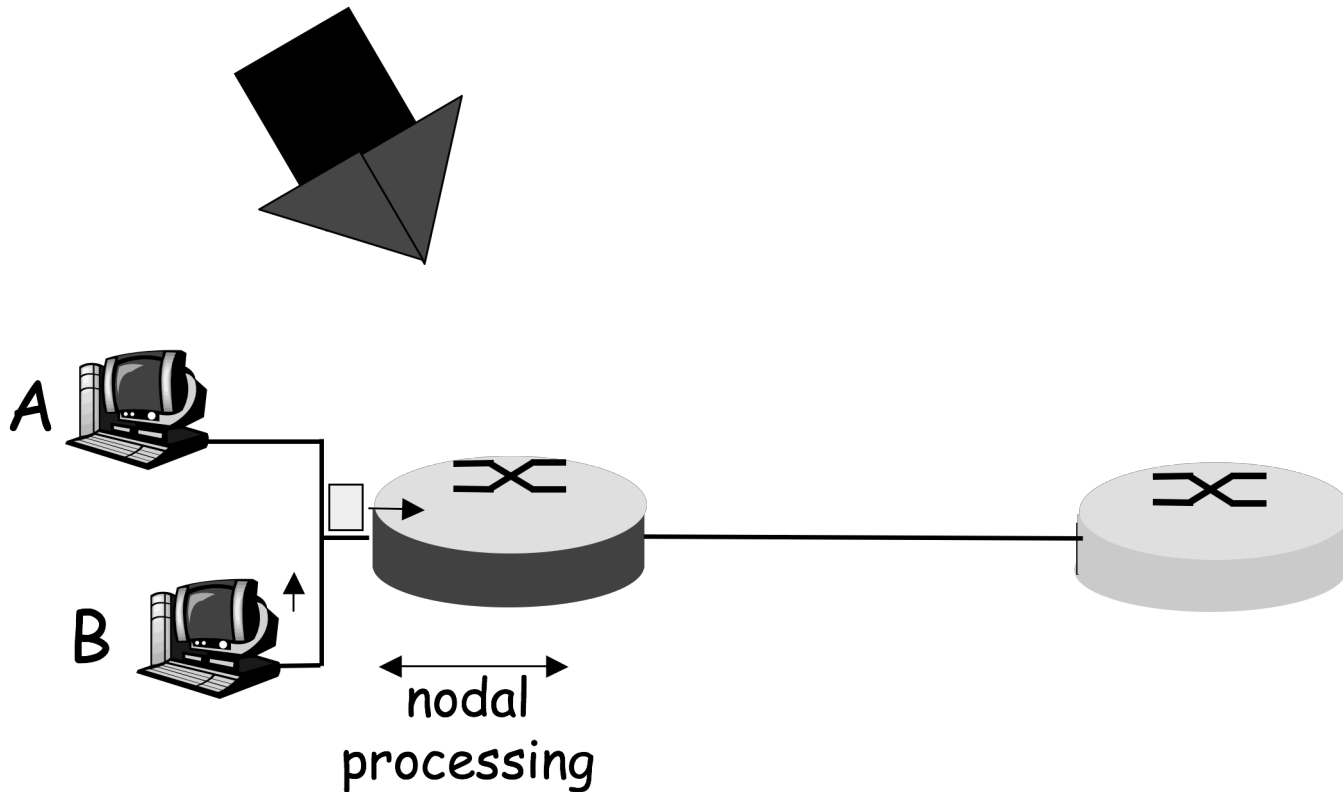


Delay in packet-switched networks

packets experience delay
on end-to-end path

- four sources of delay at each hop

- nodal processing:
 - check bit errors
 - determine output link



Delay in packet-switched networks

packets experience delay
on end-to-end path

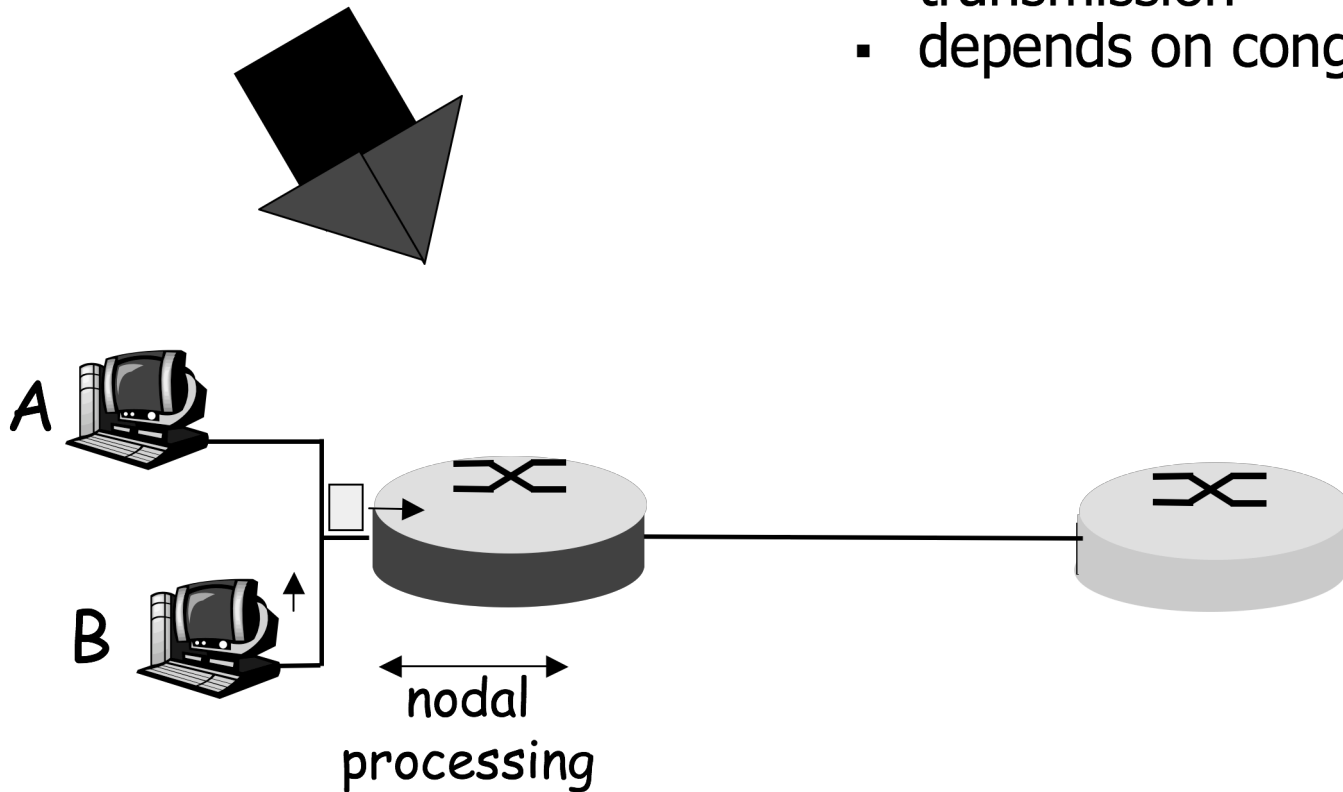
- four sources of delay at each hop

- nodal processing:

- check bit errors
- determine output link

- queuing

- time waiting at output link for transmission
- depends on congestion level of node



Delay in packet-switched networks

packets experience delay
on end-to-end path

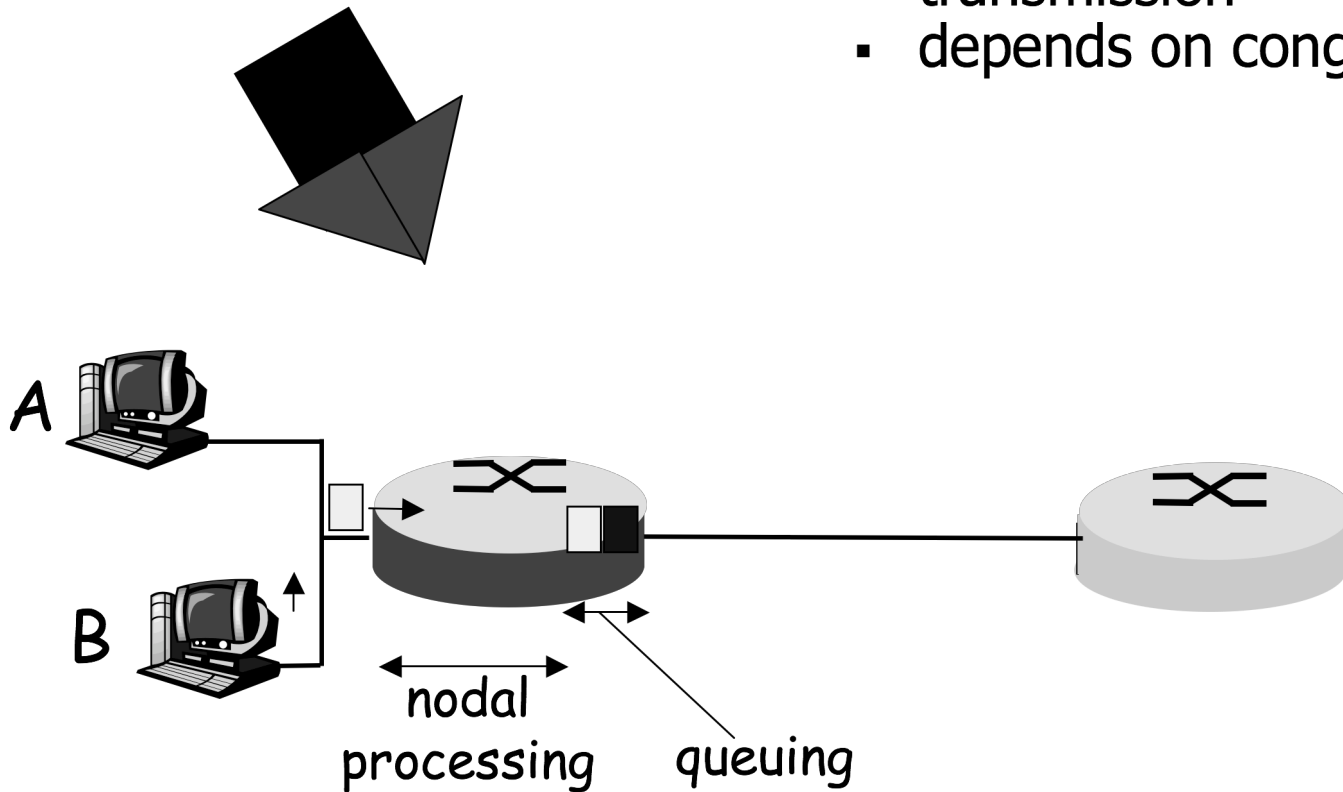
- four sources of delay at each hop

- nodal processing:

- check bit errors
- determine output link

- queuing

- time waiting at output link for transmission
- depends on congestion level of node



Delay in packet-switched networks

packets experience delay
on end-to-end path

- four sources of delay at each hop

- nodal processing:

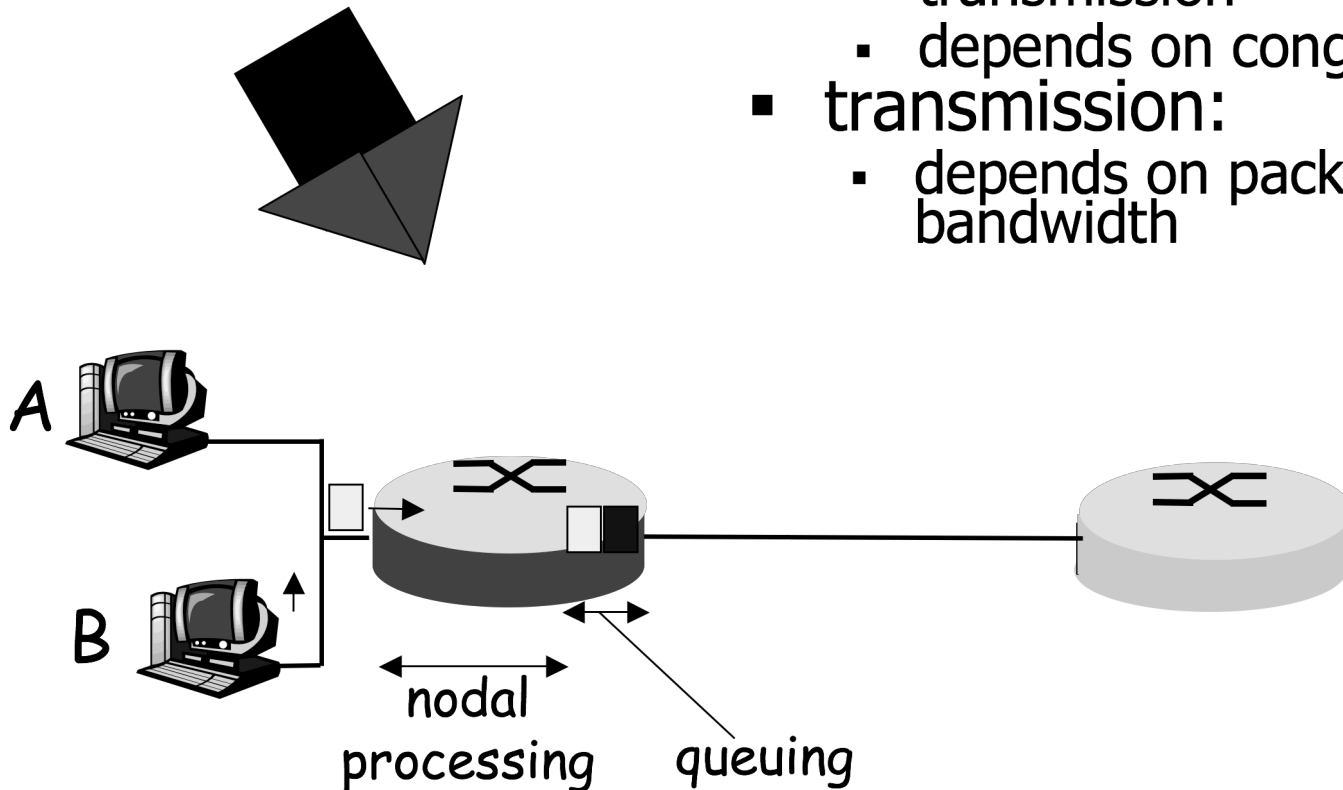
- check bit errors
- determine output link

- queuing

- time waiting at output link for transmission
- depends on congestion level of node

- transmission:

- depends on packet length and link bandwidth



Delay in packet-switched networks

packets experience delay
on end-to-end path

- four sources of delay at each hop

- nodal processing:

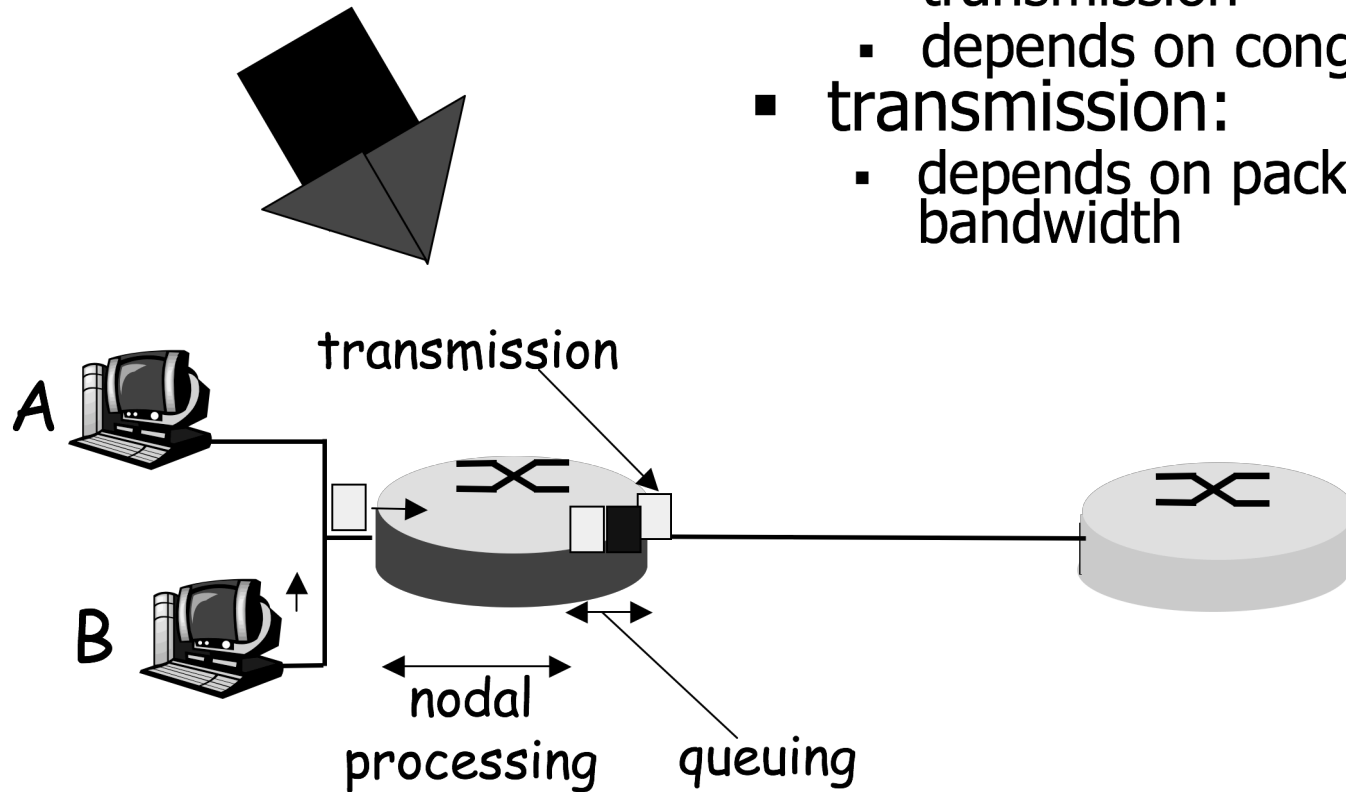
- check bit errors
- determine output link

- queuing

- time waiting at output link for transmission
- depends on congestion level of node

- transmission:

- depends on packet length and link bandwidth



Delay in packet-switched networks

packets experience delay
on end-to-end path

- four sources of delay at each hop

- nodal processing:

- check bit errors
- determine output link

- queuing

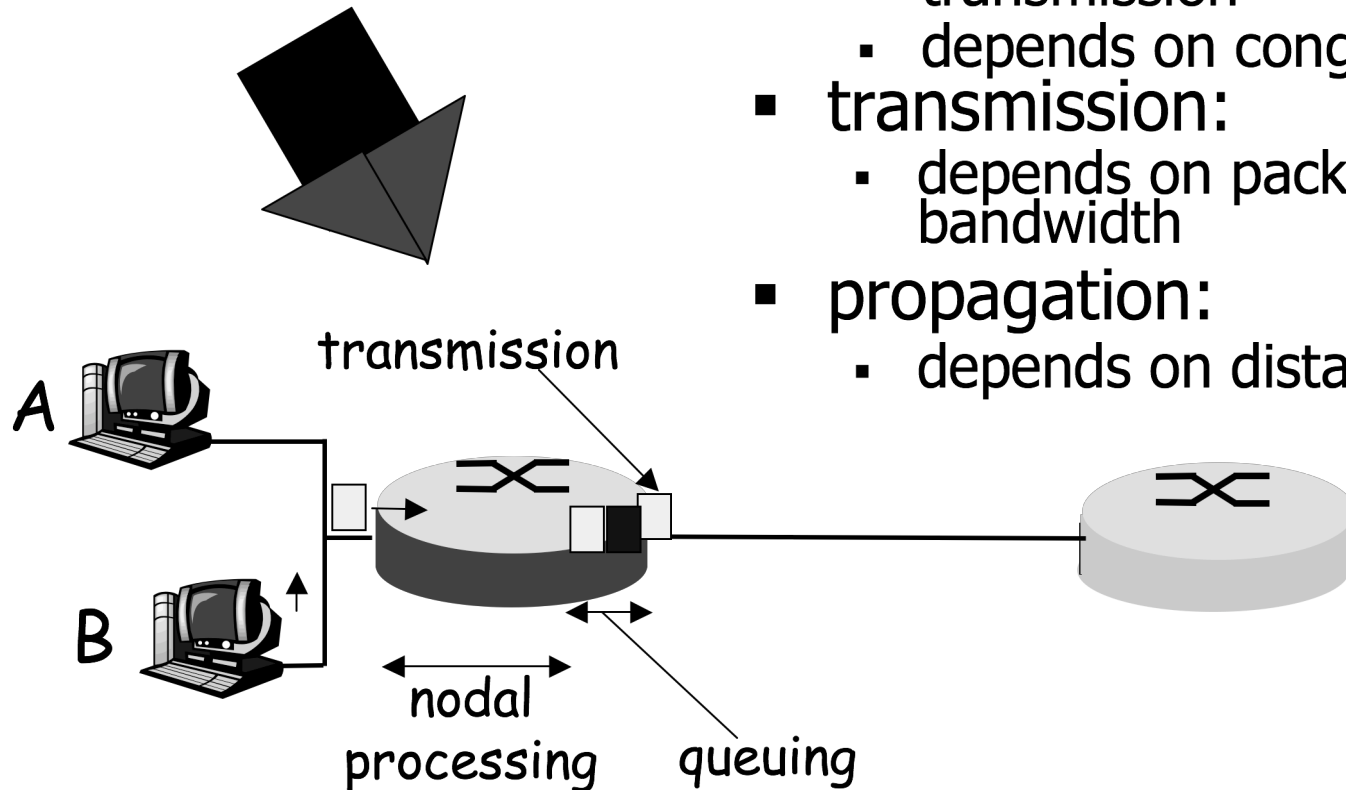
- time waiting at output link for transmission
- depends on congestion level of node

- transmission:

- depends on packet length and link bandwidth

- propagation:

- depends on distance between nodes



Delay in packet-switched networks

packets experience delay
on end-to-end path

- four sources of delay at each hop

- nodal processing:

- check bit errors
- determine output link

- queuing

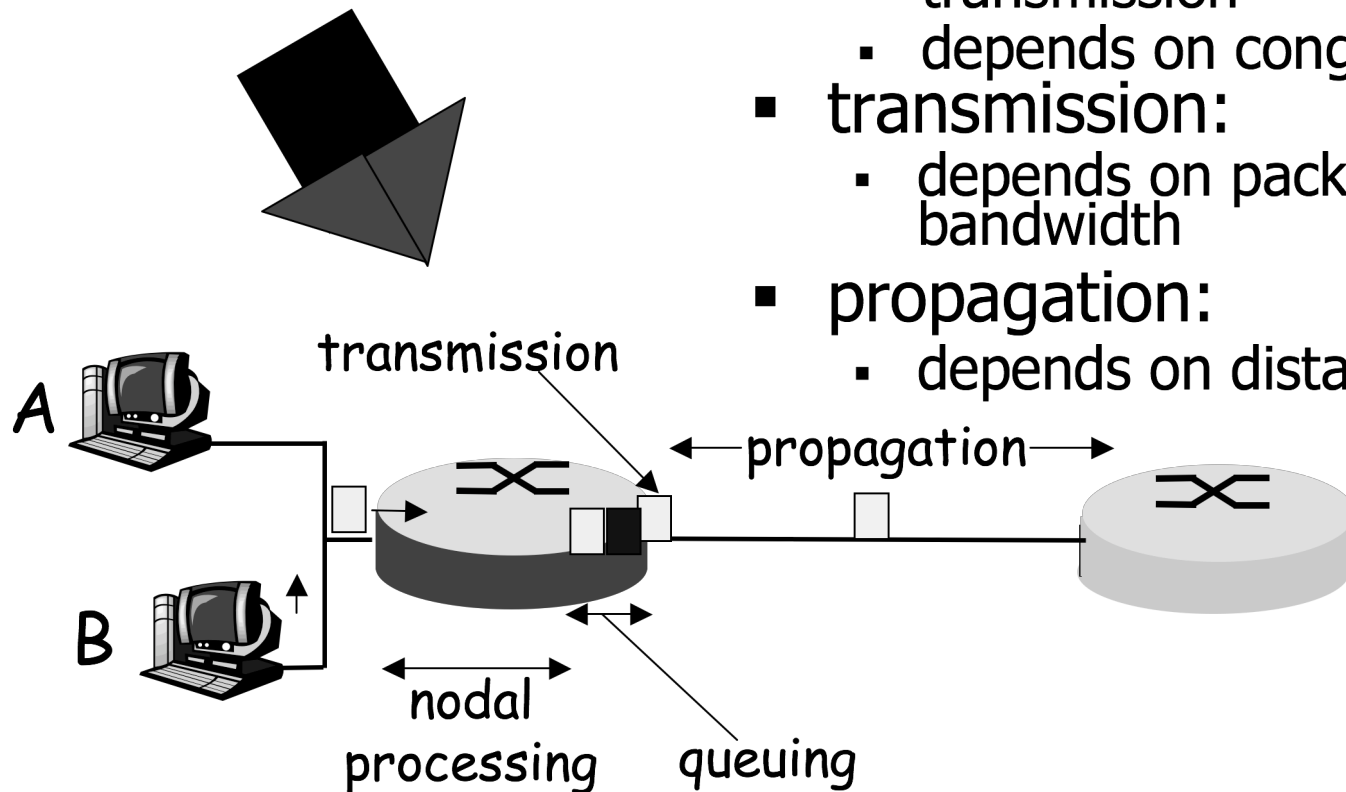
- time waiting at output link for transmission
- depends on congestion level of node

- transmission:

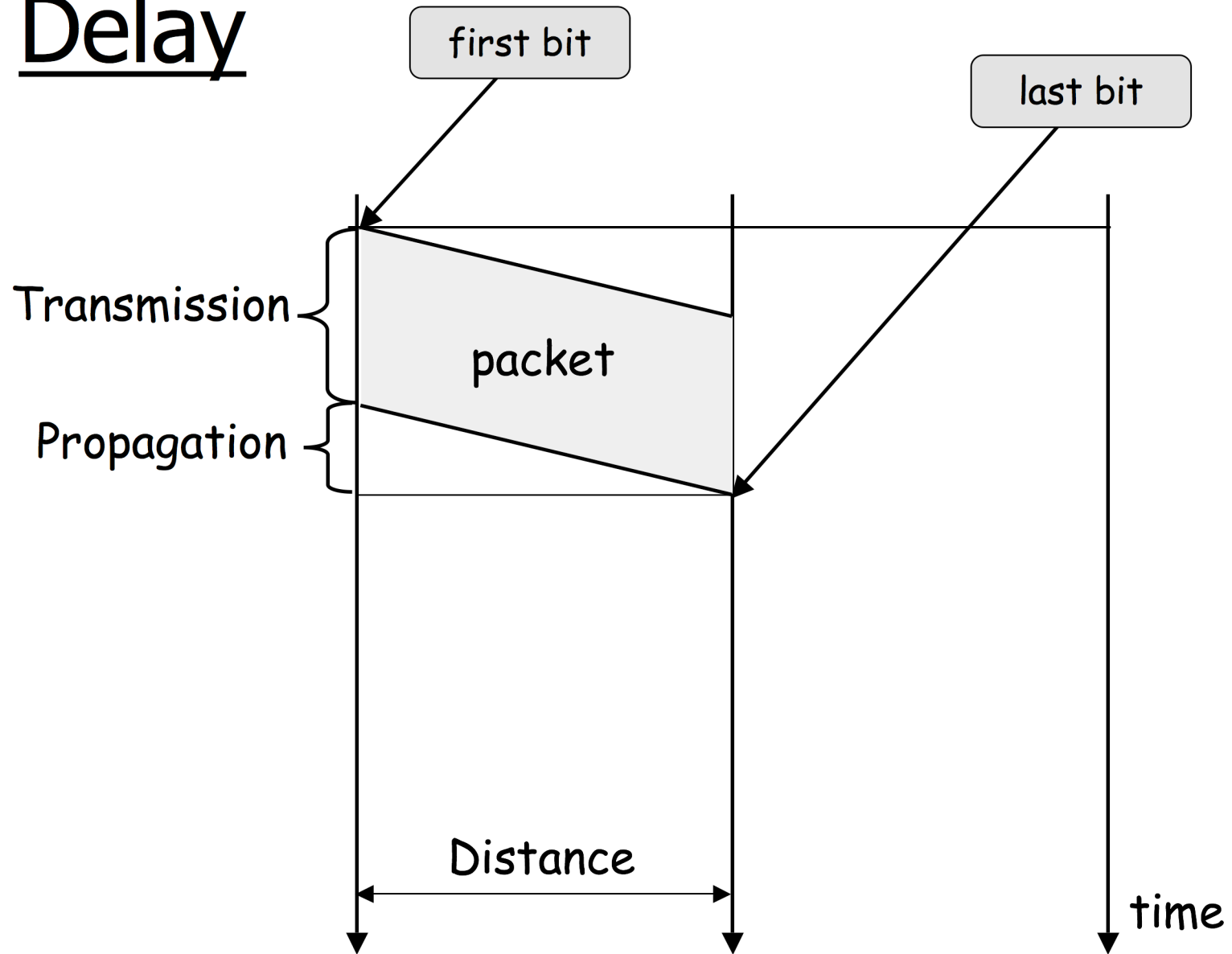
- depends on packet length and link bandwidth

- propagation:

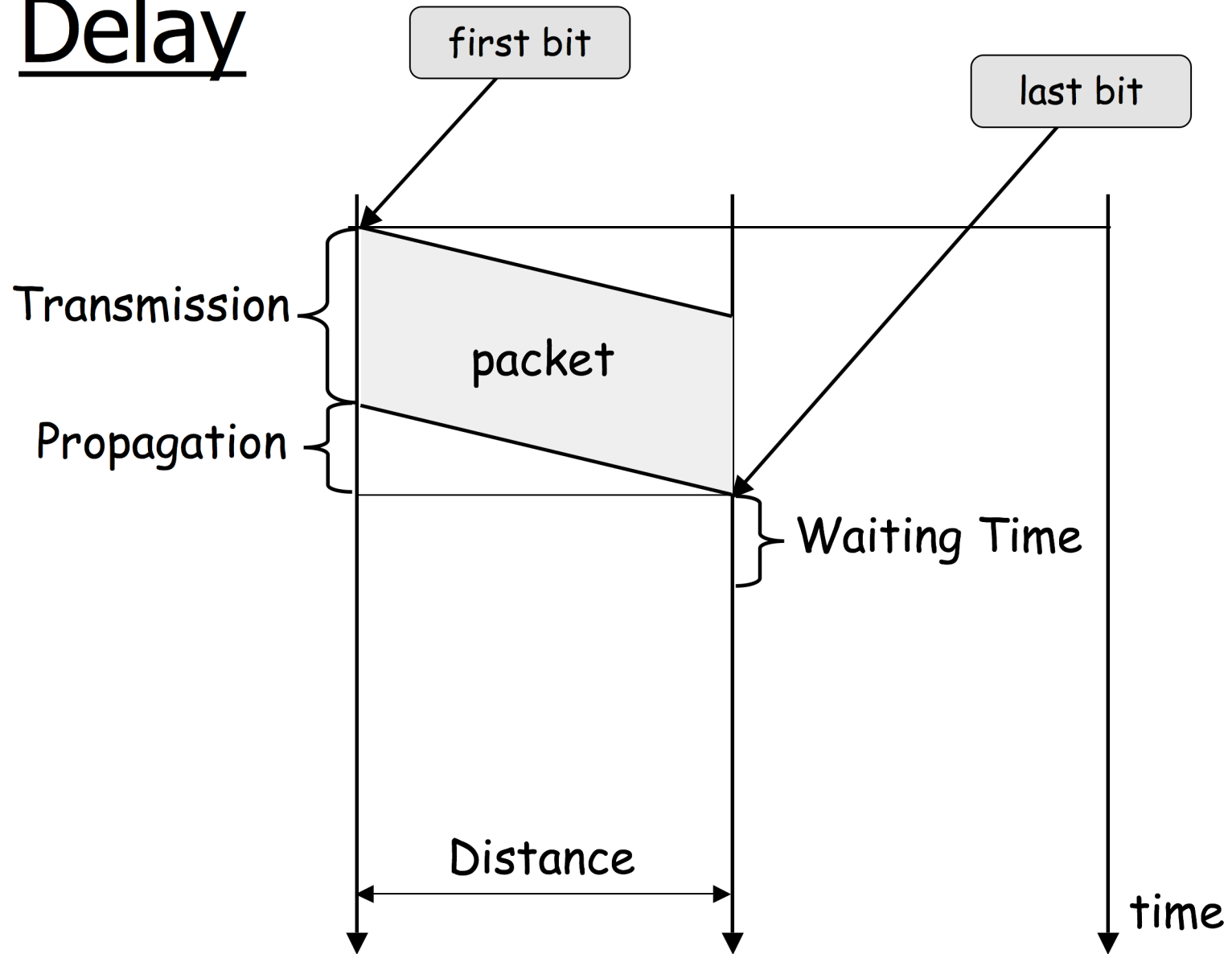
- depends on distance between nodes



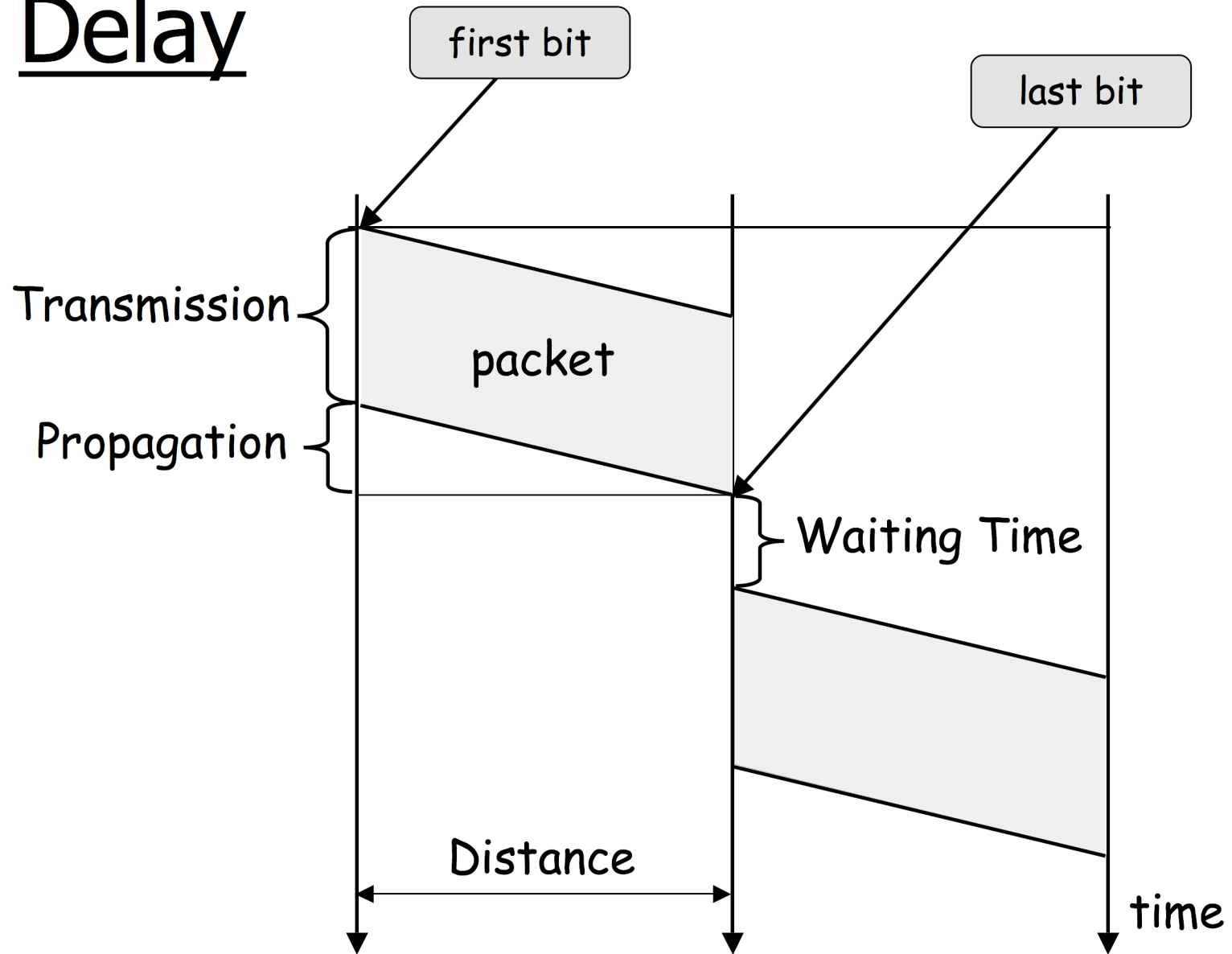
Delay



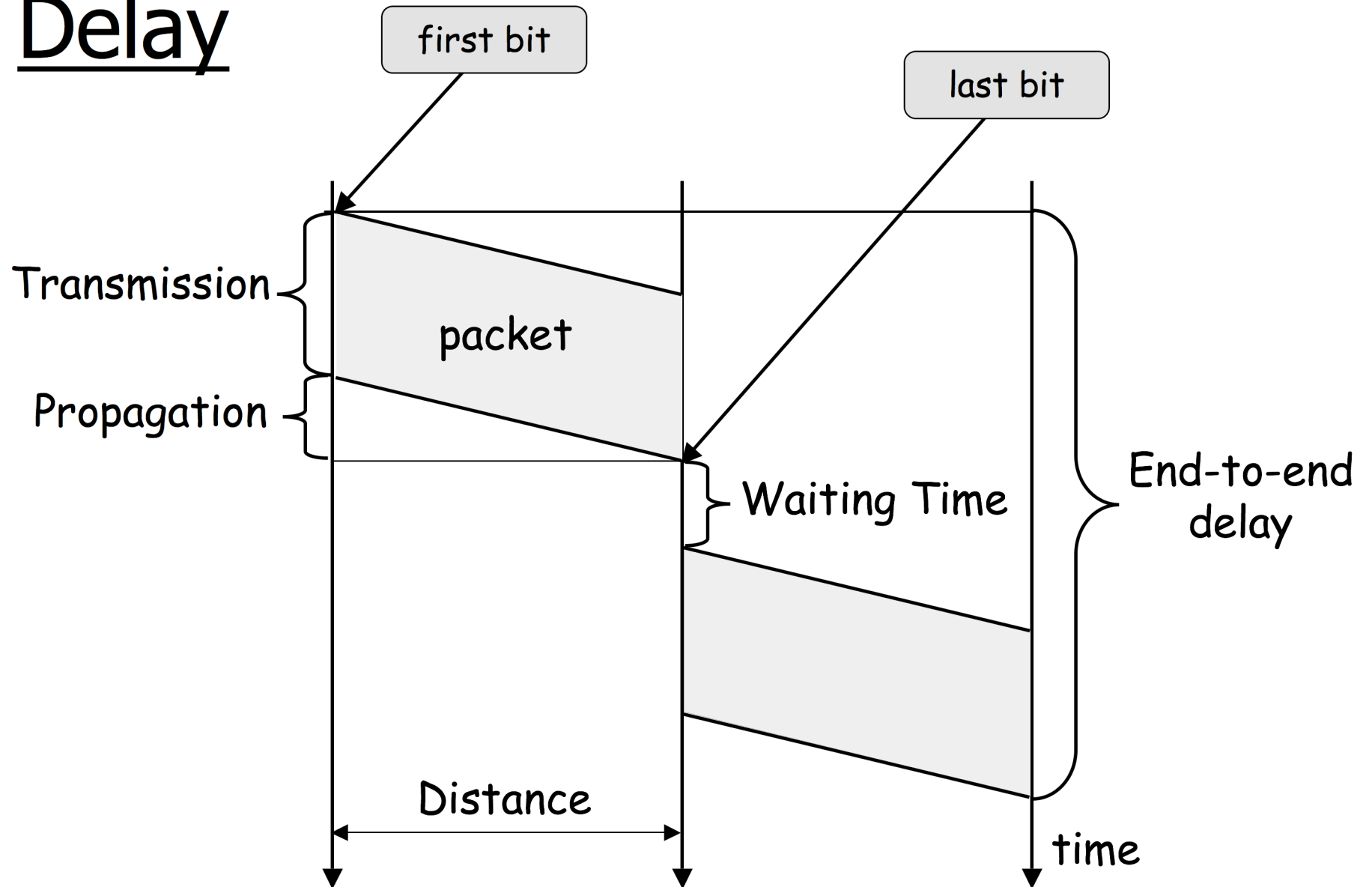
Delay



Delay



Delay



Performance

- Latency
 - Latency = Propagation + Transmission + Wait
 - Propagation = Distance / Speed
 - copper : Speed = 2.3×10^8 m/s
 - glass : Speed = 2×10^8 m/s
 - Transmission = Size / BitRate
- 5 μ s/km
- New York - Los Angeles in 24 ms
 - request - 1 byte, response - 1 byte: 48 ms
 - 25 MB file on 10 Mb/s: 20 s
- World tour in 0.2 s

Example

- At time 0, computer A sends a packet of size 1000 bytes to B; at what time is the packet received by B (speed = 2×10^8 m/s)?

<i>distance</i>	20 km	20000 km	2 km	20 m
<i>bit rate</i>	10kb/s	1 Mb/s	10 Mb/s	1 Gb/s
<i>propagation</i>	0.1ms	100 ms	0.01 ms	0.1 μ s
<i>transmission</i>	800 ms	8 ms	0.8 ms	8 μ s
<i>latency</i>	?	?	?	?

Example

- At time 0, computer A sends a packet of size 1000 bytes to B; at what time is the packet received by B (speed = 2×10^8 m/s)?

<i>distance</i>	20 km	20000 km	2 km	20 m
<i>bit rate</i>	10kb/s	1 Mb/s	10 Mb/s	1 Gb/s
<i>propagation</i>	0.1ms	100 ms	0.01 ms	0.1 μ s
<i>transmission</i>	800 ms	8 ms	0.8 ms	8 μ s
<i>latency</i>	?	?	?	?

modem

satellite

LAN

Hippi

Example

- At time 0, computer A sends a packet of size 1000 bytes to B; at what time is the packet received by B (speed = 2×10^8 m/s)?

<i>distance</i>	20 km	20000 km	2 km	20 m
<i>bit rate</i>	10kb/s	1 Mb/s	10 Mb/s	1 Gb/s
<i>propagation</i>	0.1ms	100 ms	0.01 ms	0.1 μ s
<i>transmission</i>	800 ms	8 ms	0.8 ms	8 μ s
<i>latency</i>	800.1 ms	108 ms	0.81 ms	8.1 μ s

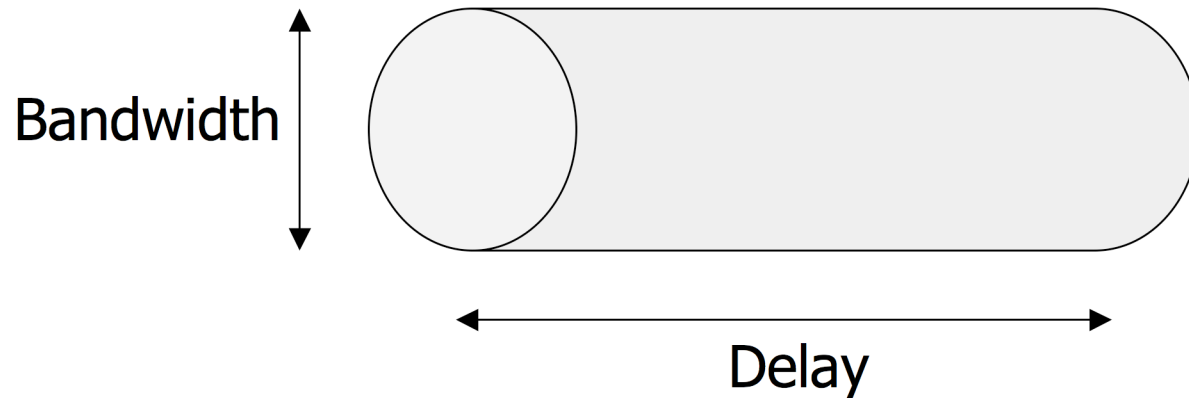
modem

satellite

LAN

Hippi

Bandwidth-Delay Product



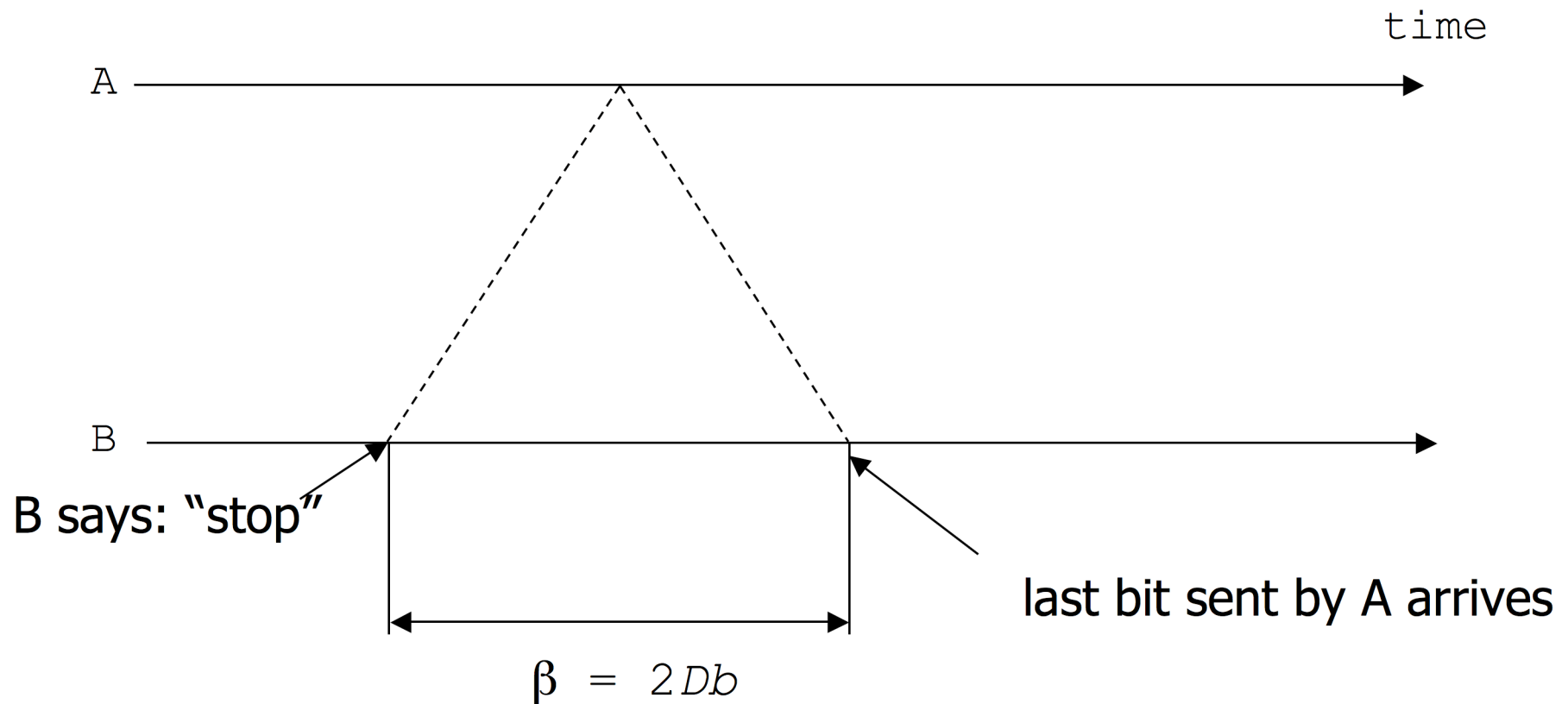
- Bandwidth-Delay product
 - how many bits should we send before the arrival of the first bit?
 - good utilization - keep the pipe filled!

Bandwidth-Delay Product

- File transfer: 1 Mbit, 100 ms delay
 - 1 Mb/s link, $D \times b = 0.1$ Mbit
 - 10 transmissions, 10% each time
 - 1 Gbit/s link, $D \times b = 100$ Mbit
 - 1 transmission, pipe not filled

Bandwidth-Delay Product

- Consider the scenario :



- β = maximum number of bits B can receive after saying stop
- large β means: delayed feedback
- amount of data "in the pipe"

A Simple Protocol: Stop and Go

- Packets may be lost during transmission: bit errors due to channel imperfections, various noises.
- Computer A sends packets to B; B returns an acknowledgement packet immediately to confirm that B has received the packet;
A waits for acknowledgement before sending a new packet; if no acknowledgement comes after a delay T_1 , then A retransmits

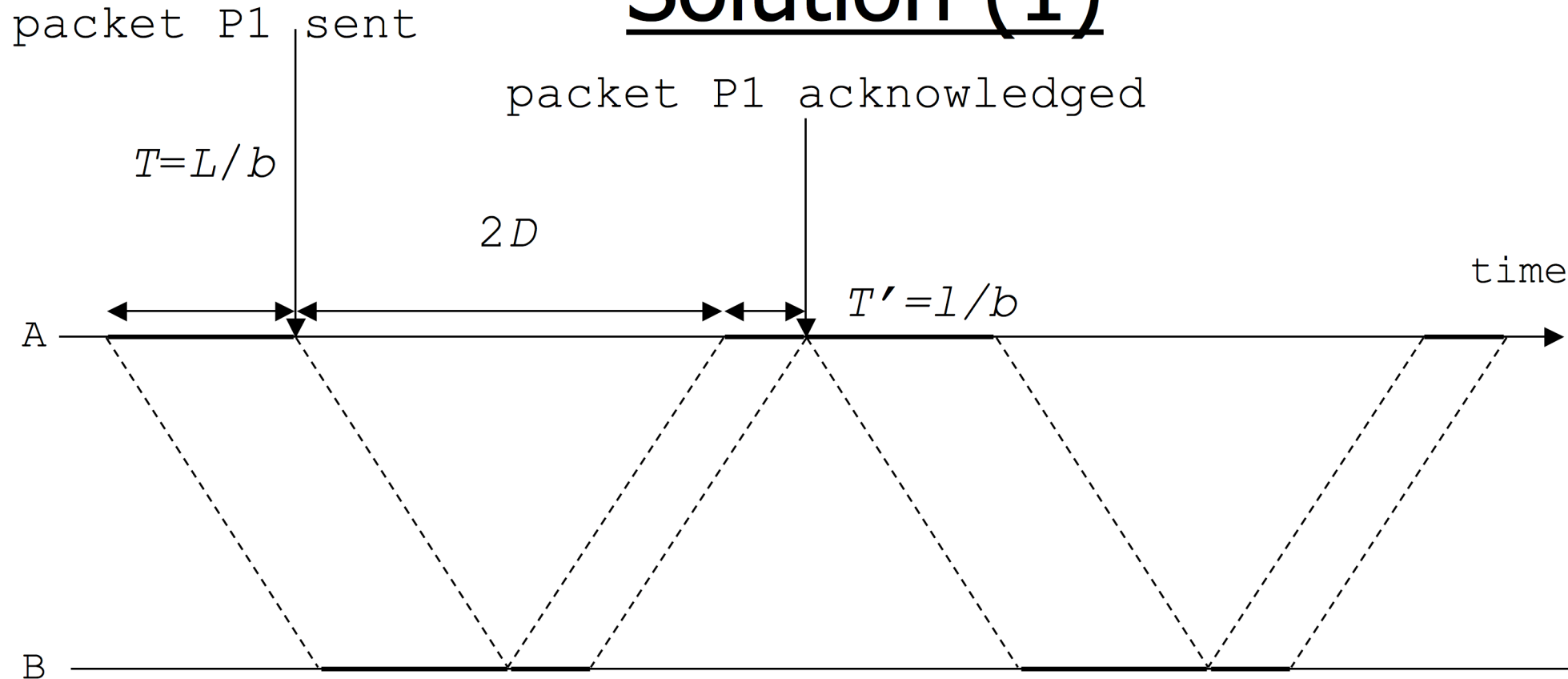
A Simple Protocol: Stop and Go

- **Question:** What is the maximum throughput assuming that there are no losses?

notation:

- packet length = L , constant (in bits);
- acknowledgement length = l , constant
- channel bit rate = b ;
- propagation = D
- processing time = 0

Solution (1)



$$\text{cycle time} = T + 2D + T'$$

$$\text{useful bits per cycle time} = L$$

$$\text{throughput} = Lb / (L + l + 2Db) = b / (\omega + \beta/L)$$

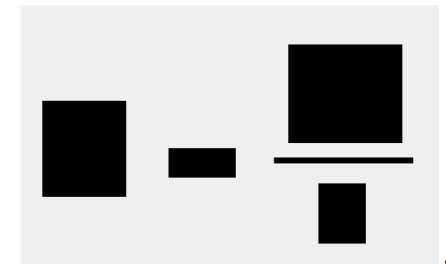
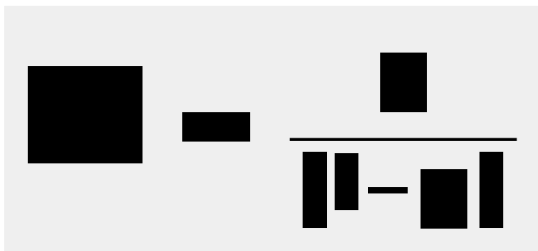
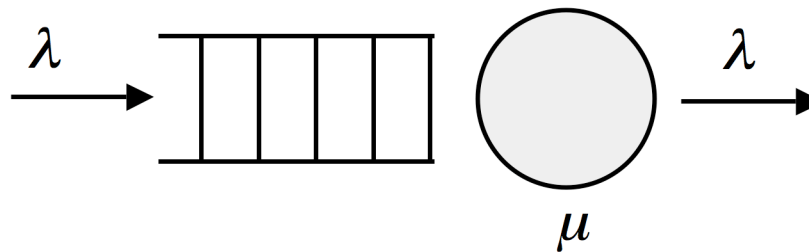
with $\omega = (L+l)/L = \text{overhead}$ and $\beta = 2Db = \text{bandwidth-delay product}$

Solution (2)

distance	20 km	20000 km	2 km	20 m
bit rate	10kb/s	1 Mb/s	10 Mb/s	1 Gb/s
propagation	0.1ms	100 ms	0.01 ms	0.1μs
transmission	800 ms	8 ms	0.8 ms	8 μs
reception time	800.1 ms	108 ms	0.81 ms	8.1 μs
	<i>modem</i>	<i>satellite</i>	<i>LAN</i>	<i>Hippi</i>
$\beta = 2 Db$	2 bits	200 000 bits	200 bits	200 bits
throughput = $b \times 99.98\%$		3.8%	97.56%	97.56%

Waiting time

- Queueing system M/M/1
 - interarrival times $\sim$ exponentially distributed
 - service times $\sim$ exponentially distributed
 - arrival rate λ , service rate μ , utilization $\rho = \lambda/\mu$
 - number of packets N , waiting time T

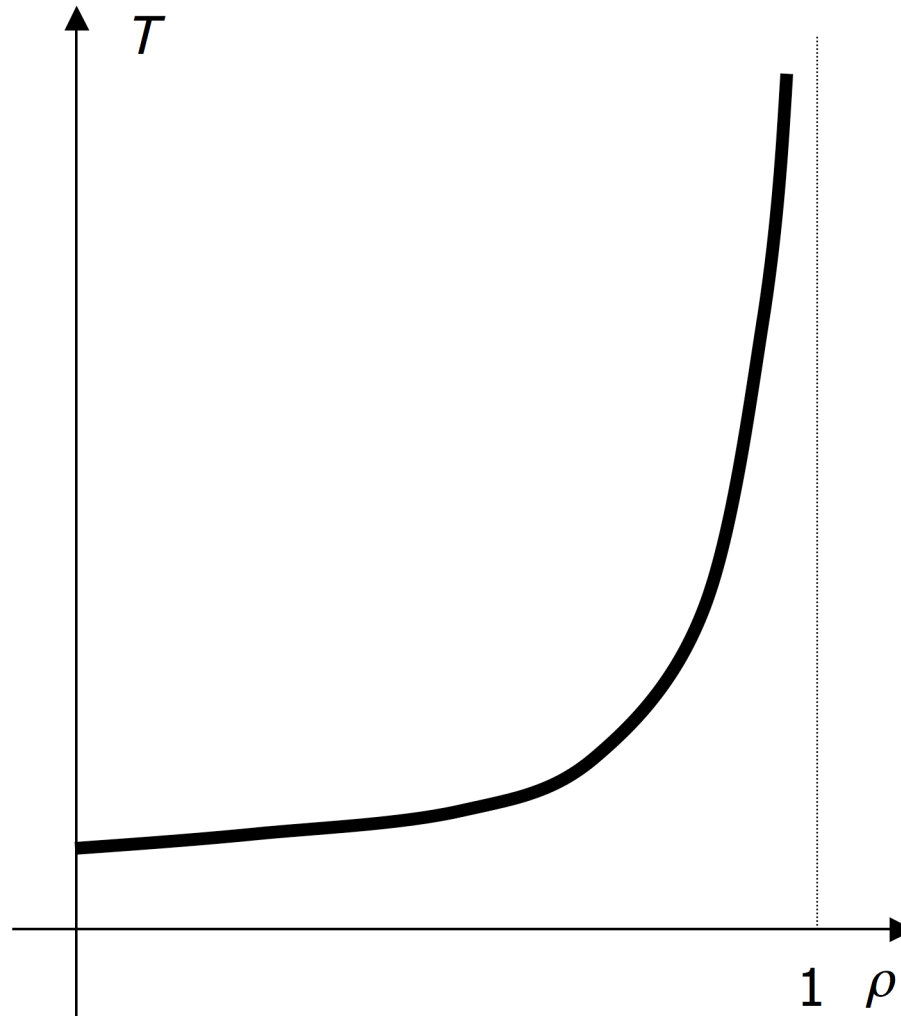


Waiting time

- Average packet length 1500 bytes
 - link with 1 Mb/s bit rate (propagation = 0)
 - transmission time 12 ms
 - service rate 83 packet/s

λ	[p/s]	10	40	60	70
$1/\lambda$	[ms]	100	25	16	14
T	[ms]	13	23	43	76

Waiting time



Summary

- Network architectures
 - protocol architectures
 - different protocol stacks, overlaid stacks
 - interconnection structure
 - switches, routers
 - related protocols
 - complex protocol families
- Performance
 - transmission
 - propagation
 - bandwidth-delay product
 - queueing delay