Computer Networking

Introduction

Prof. Andrzej Duda
duda@imag.fr

http://duda.imag.fr

1

Course goals

- Understand TCP/IP and networking concepts
- Approach
  - bottom-up
  - descriptive,
  - use Internet as an example
  - join the application layer seen during the 1st year
- Organization
  - 27 h course
  - demos, exercises, questions, homework
  - slides are not exhaustive - you must take notes and ask questions!
- Your team
  - course: A. Duda
  - lab: Gilles Berger-Sabbatel, Sébastien Viardot, Laurentiu-Sorin Paun

2

Networking lab

- Important part of the course
  - perform required operations, write lab reports
  - 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 equipement: hubs, switches, routers
  - isolated from the rest of the network

3

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

4

Course support

- Web site
  - `http://duda.imag.fr/2at`
- J. Kurose, K. Ross “Computer Networking”, 2nd edition, Addison Wesley, 2002
- 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)

5

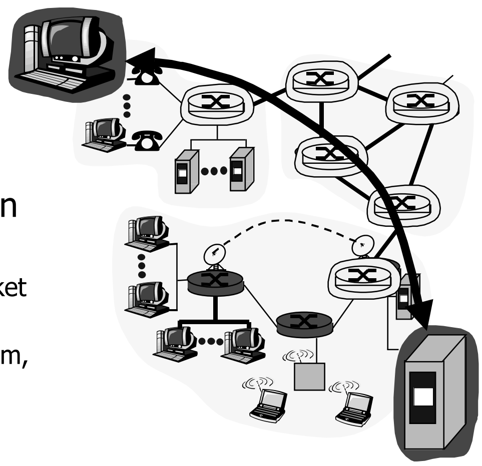
Overview

- 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

6

Inside the Internet

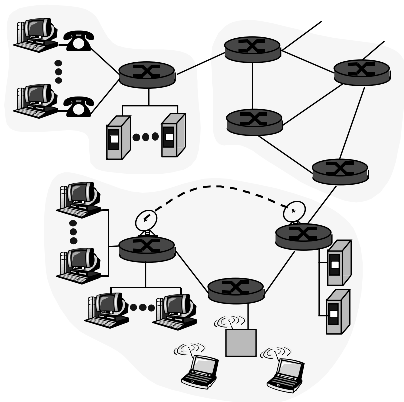
- Between end systems
  - TCP protocol for reliable transmission
- Inside the network core
  - IP protocol: forwarding packets between routers
- Between routers or between end system and router
  - high speed link: ATM, POS (Packet over SONET), satellite links
  - access network: Ethernet, modem, xDSL,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



7

Network structure

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

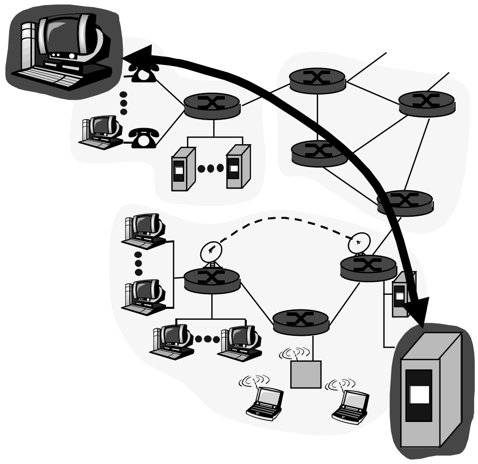


8

We are now going to delve a bit more deeply into the components of a computer network. We begin at the edge of network and look at the components with which we are most familiar--the computers (for example, PCs and workstations) that we use on a daily basis. Then, moving from the network edge to the network core we have switchs and routers. Finally, we have the access network – the physical link(s) that connect an end system to its edge router – that is, to the first router on a path from the end system to any other end system.

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



9

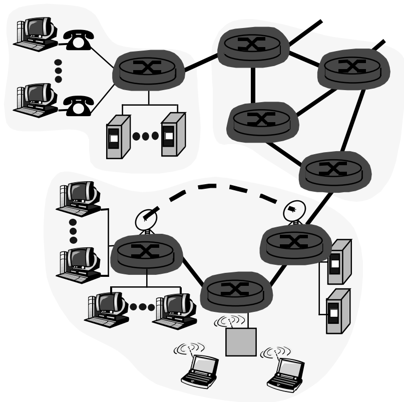
In computer networking jargon, the computers that we use on a daily basis are often referred to as hosts or end systems. They are referred to as hosts because they host (run) application-level programs such as a Web browser or server program, or an e-mail program. They are also referred to as end systems because they sit at the edge of the network.

Hosts are sometimes further divided into two categories: clients and servers. Informally, clients often tend to be desktop PCs or workstations, whereas servers are more powerful machines. But there is a more precise meaning of a client and a server in computer networking. In the so-called client/server model, a client program running on one end system requests and receives information from a server running on another end system. This client/server model is undoubtedly the most prevalent structure for Internet applications. The Web, e-mail, file transfer, remote login (for example, Telnet), newsgroups, and many other popular applications adopt the client/server model.

The other model used in computer networks is referred to as peer-to-peer model. In this model the two hosts takes the same role and run the same programs. A typical example of peer-to-peer application is the 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)



10

The network core is the mesh of routers that interconnect the end systems. In the figure, we highlights the network core in the thick, shaded lines.

There are two fundamental approaches towards building a network core: circuit switching and packet switching. In circuit-switched networks, the resources needed along a path (buffers, link bandwidth) to provide for communication between the end systems are reserved for the duration of the session. In packet-switched networks, these resources are not reserved; a session's messages use the resource on demand, and as a consequence, may have to wait (that is, queue) for access to a communication link.

The ubiquitous telephone networks are examples of circuit-switched networks. Consider what happens when one person wants to send information (voice or facsimile) to another over a telephone network. Before the sender can send the information, the network must first establish a connection between the sender and the receiver.

In modern packet-switched networks, the source breaks long messages into smaller packets. Between source and destination, each of these packets can take different communication links and packet switches (also known as routers). Packets are transmitted over each communication link at a rate equal to the full transmission rate of the link. Most packet switches use store-and-forward transmission at the inputs to the links. Store-and-forward transmission means that the switch must receive the entire packet before it can begin to transmit the first bit of the packet onto the outbound link. Thus store-and-forward packet switches introduce a store-and-forward delay at the input to each link along the packet's route. This delay is proportional to the packet's length in bits. In particular, if a packet consists of L bits, and the packet is to be forwarded onto an outbound link of R bps, then the store-and-forward delay at the switch is L/R seconds.

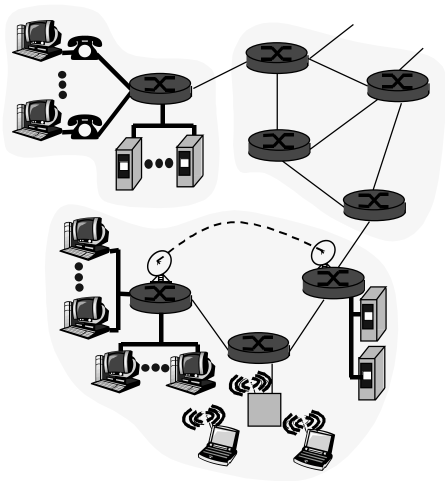
Access networks and physical media

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



11

The access networks are the physical link(s) that connect an end system to its edge router.The figure shows the access networks' links highlighted in thick, shaded lines.

Access networks can be loosely divided into three categories:

o **Residential access networks**, connecting a home end system into the network

o **Institutional access networks**, connecting an end system in business or educational institution into the network

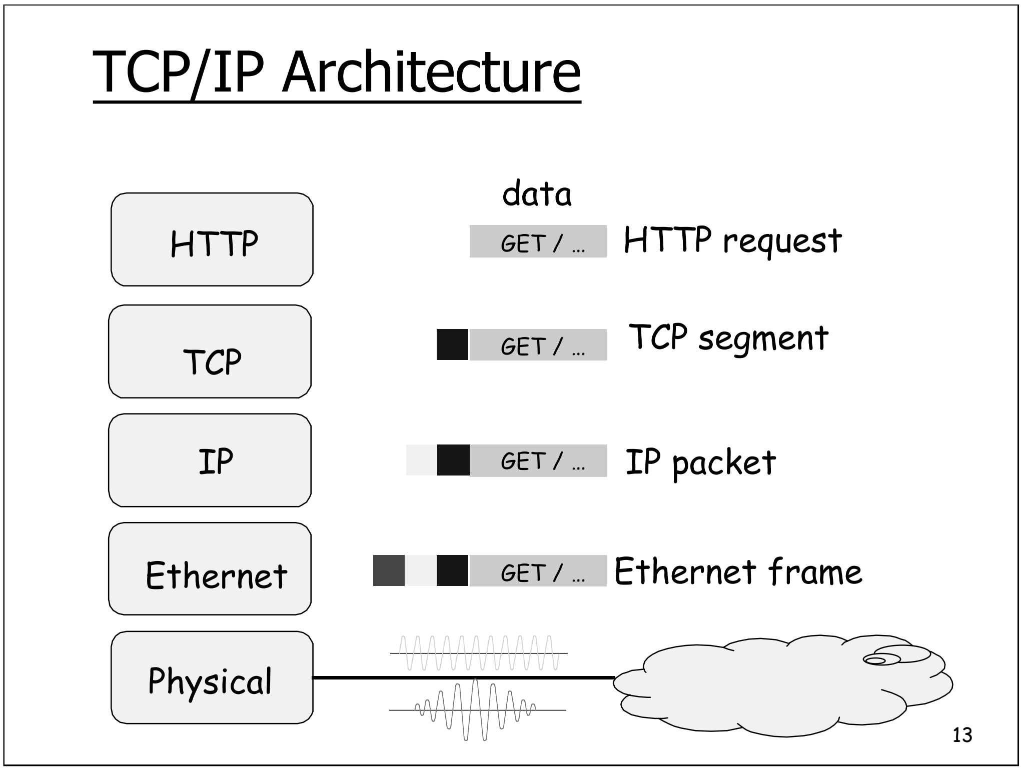
o **Mobile access networks**, connecting a mobile end system into the network

These categories are not hard and fast; some corporate end systems may well use the access network technology that we ascribe to residential access networks, and vice versa.

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

12


TCP/IP Architecture
data
HTTP
GET / ...
HTTP request
TCP
GET / ...
TCP segment
IP
GET / ...
IP packet
Ethernet
GET / ...
Ethernet frame
Physical
13

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)

14

The Application Layer is responsible for supporting network applications. The application layer includes many protocols, including HTTP to support the Web, SMTP to support electronic mail, and FTP to support file transfer. We shall see in Chapter 2 that it is very easy to create our own new application-layer protocols.

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 (transmisssion control protocol)
    - reliable
    - offers a stream service (unit of information is a byte)

15

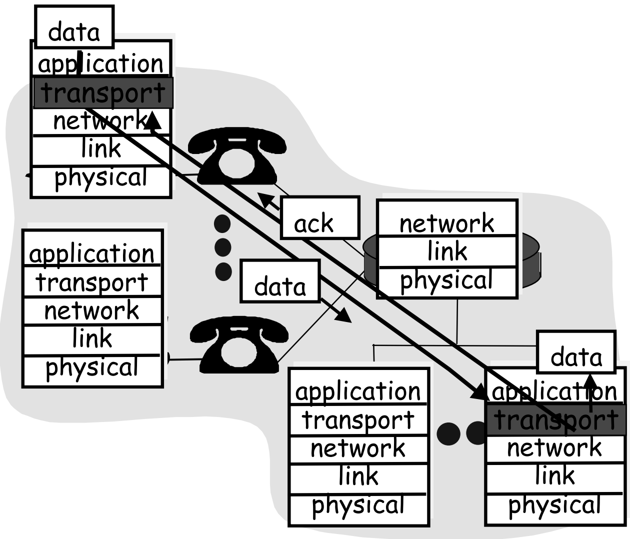
Physical, data link and network layers are sufficient to build a packet transport system between computers. However, this is not enough for the programmer. When you write a low-level program which uses the network (as we will do in this lecture), you do not handle packets, but data. The primary goal of the **transport layer** is to provide the programmer with an interface to the network.

Second, the transport layer uses the concept of **port**. A port is a number which is used locally (on one machine) and identifies the source and destination of the packet **inside the machine**. We will come back to the concept of ports later in this chapter.

The transport layer exists in two varieties: unreliable and reliable. The unreliable variety simply sends packets, and does not attempt to guarantee any delivery. The reliable variety, in contrast, makes sure that data does reach the destination, even if some packets may be lost from time to time. In the Internet there are two transport protocols, TCP and UDP, either of which can transport application-layer messages. TCP provides a connection-oriented service to its applications. This service includes guaranteed delivery of application-layer messages to the destination and flow control (that is, sender/receiver speed matching). TCP also segments long messages into shorter segments and provides a congestion control mechanism, so that a source throttles its transmission rate when the network is congested. The UDP protocol provides its applications a connectionless service, which is very much a no-frills service.

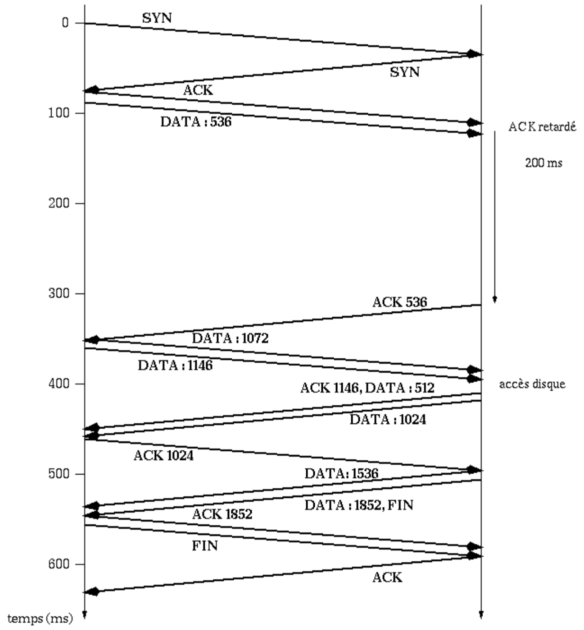
Layering: *logical* communication

- 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



16

TCP



HTTP Request : 1146 octets

HTTP Reply : 1852 octets

Taille maximale de segment : 536 octets

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

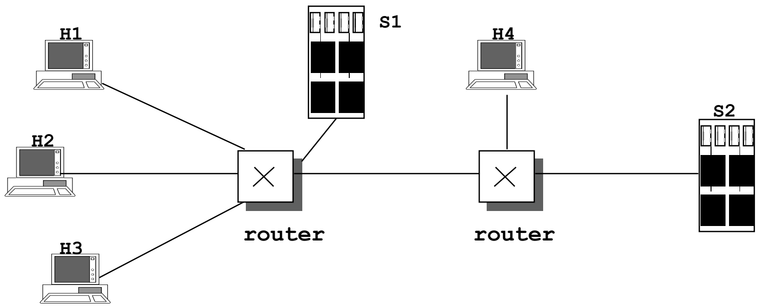
Problèmes :

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

7

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



18

Modern networks have more than physical and data link. The **network layer** is the set of mechanisms that can be used to send packets from one computer to another in the world. There are two types of networks:

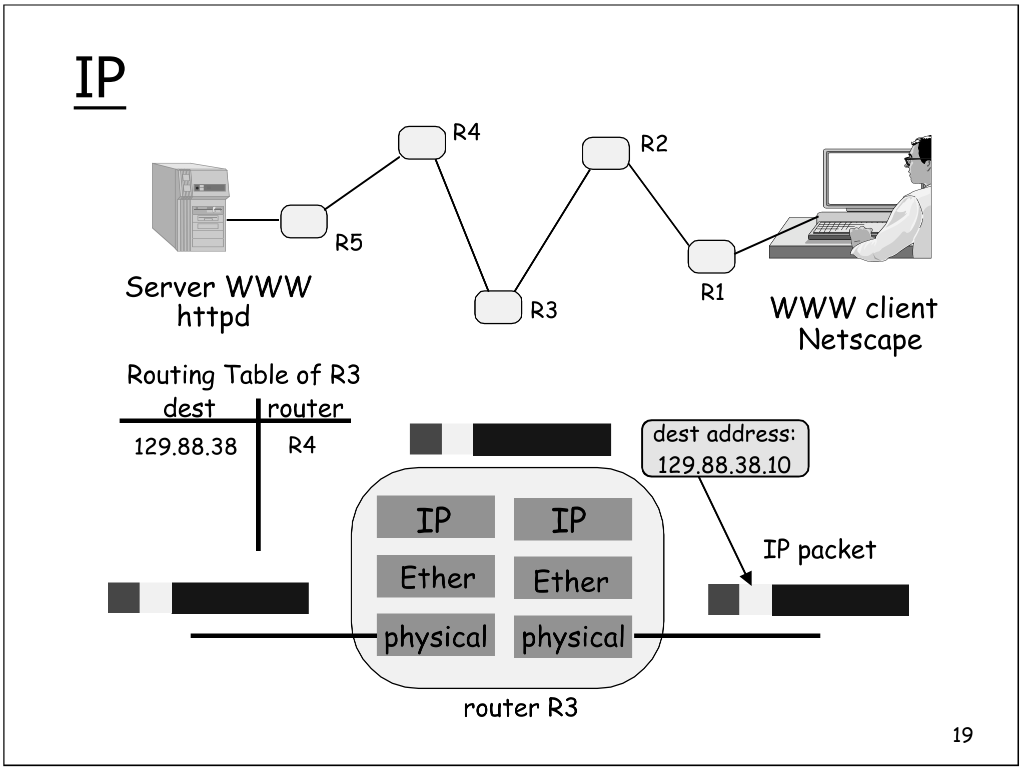
With **Packet switching,** data packets can be carried together on the same link. They are differentiated by addressing information. Packet switching is the basis for all data networks today, including the Internet, public data networks such as Frame Relay, X.25, or ATM.

Circuit Switching is the way telephone networks operate. A circuit emulates the physical signals of a direct end-to-end cable. When computers are connected by a circuit switched network, they establish a direct data link over the circuit. This is used today for modem access to a data network.

Modern circuit switches are based on byte multiplexing and are thus similar to packet switches, with the main difference that they perform non-statistical multiplexing (see later in this chapter).

A network has **Intermediate systems** (ISs): those are systems that send data to next ISs or to the destination. Using interconnected ISs saves cable and bandwidth. ISs are known under various terms depending on the context: routers (TCP/IP, AppleTalk,…), switches (X.25, Frame Relay, ATM, telephone), communication controllers (SNA), network nodes (APPN).

The Internet's network layer has two principle components. It has a protocol that defines the fields in the IP datagram as well as how the end systems and routers act on these fields. This protocol is the celebrated **IP protocol**. There is only one IP protocol, and all Internet components that have a network layer must run the IP protocol. The Internet's network layer also contains routing protocols that determine the routes that datagrams take between sources and destinations. The Internet has many routing protocols.



Modern networks have more than physical and data link. The **network layer** is the set of mechanisms that can be used to send packets from one computer to another in the world. There are two types of networks:

With **Packet switching,** data packets can be carried together on the same link. They are differentiated by addressing information. Packet switching is the basis for all data networks today, including the Internet, public data networks such as Frame Relay, X.25, or ATM.

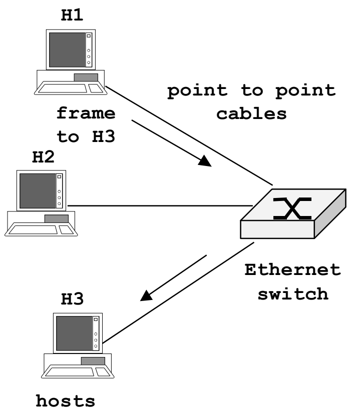
Circuit Switching is the way telephone networks operate. A circuit emulates the physical signals of a direct end-to-end cable. When computers are connected by a circuit switched network, they establish a direct data link over the circuit. This is used today for modem access to a data network.

Modern circuit switches are based on byte multiplexing and are thus similar to packet switches, with the main difference that they perform non-statistical multiplexing (see later in this chapter).

A network has **Intermediate systems** (ISs): those are systems that send data to next ISs or to the destination. Using interconnected ISs saves cable and bandwidth. ISs are known under various terms depending on the context: routers (TCP/IP, AppleTalk,…), switches (X.25, Frame Relay, ATM, telephone), communication controllers (SNA), network nodes (APPN).

The Internet's network layer has two principle components. It has a protocol that defines the fields in the IP datagram as well as how the end systems and routers act on these fields. This protocol is the celebrated **IP protocol**. There is only one IP protocol, and all Internet components that have a network layer must run the IP protocol. The Internet's network layer also contains routing protocols that determine the routes that datagrams take between sources and destinations. The Internet has many routing protocols.

Physical Layer
Data Link Layer



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

20

Physical Layer: The job of the physical layer is to move the *individual bits* within the frame from one node to the next. The protocols in this layer are again link dependent, and further depend on the actual transmission medium of the link (for example, twisted-pair copper wire, single-mode fiber optics). For example, Ethernet has many physical layer protocols: one for twisted-pair copper wire, another for coaxial cable, another for fiber, and so on. In each case, a bit is moved across the link in a different way.

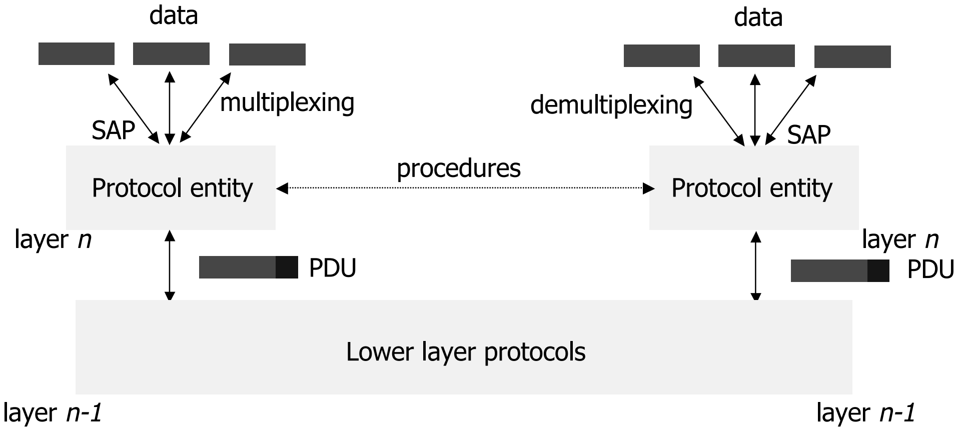
Link Layer: The services provided at the link layer depend on the specific link-layer protocol that is employed over the link. For example, some protocols provide reliable delivery on a link basis, that is, from transmitting node, over one link, to receiving node. The process is analogous to the postal worker at a mailing center who puts a letter into a plane that will deliver the letter to the next postal center along the route. Examples of link layers include Ethernet and PPP; in some contexts, ATM and frame relay can be considered link layers. As datagrams typically need to traverse several links to travel from source to destination, a datagram may be handled by different link-layer protocols at different links along its route. For example, a datagram may be handled by Ethernet on one link and then PPP on the next link. The network will receive a different service from each of the different link-layer protocols.

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

21

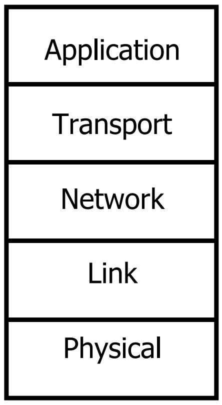
Protocol architecture



22

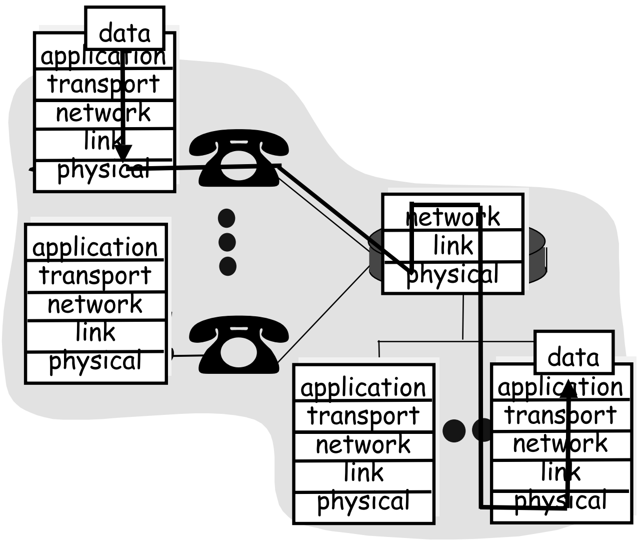
Internet protocol stack

- **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”



23

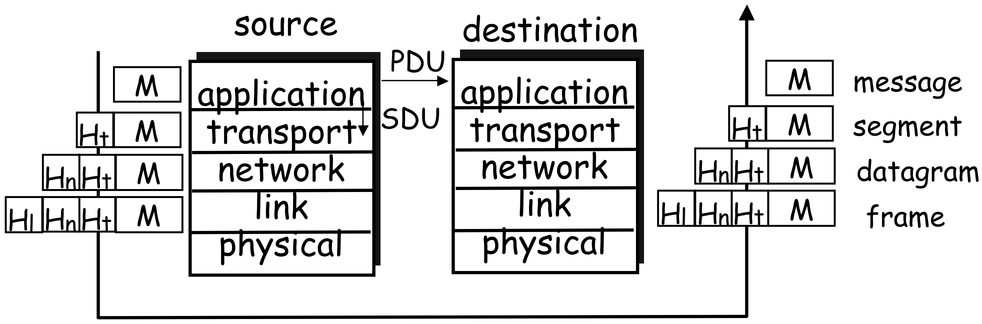
Layering: physical communication



24

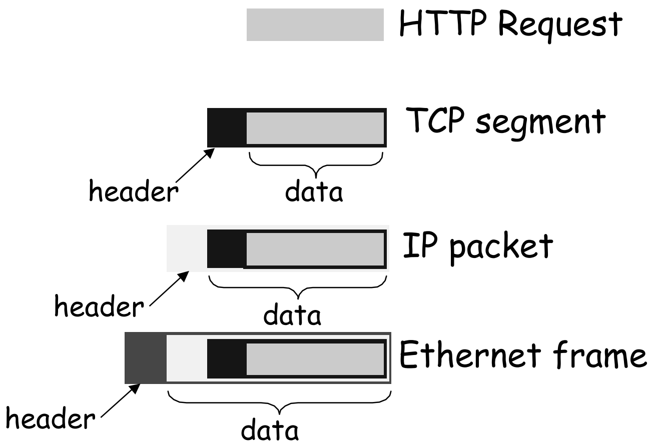
Protocol layering and data

Each layer takes data from above
- adds header information to create new data unit
- passes new data unit to layer below



25

Encapsulation



26

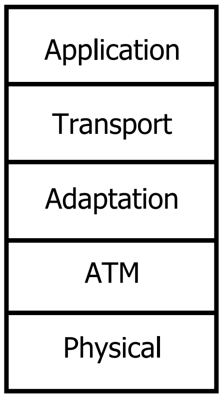
OSI ISO Model

Application	Common functions
Presentation	Interchangable formats
Session	Organizing dialog
Transport	Reliable transmission
Network	Forwarding in the network
Data link	Transmission between two nodes
Physical	Signal transmission

27

ATM protocol stack

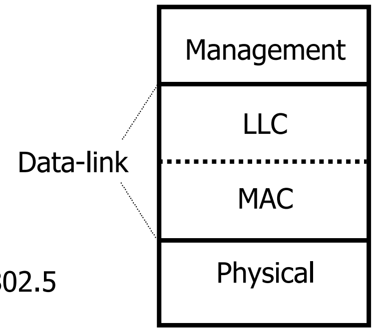
- **Application**: native applications, other protocols
  - LAN Emulation, IP, Signaling
- **Transport**: host-host data transfer
  - SSCOP
- **Adaptation**: adapt the ATM layer to different types of applications
  - circuit emulation, real-time data
  - AAL5 suitable for IP traffic
- **ATM**: cell switching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



28

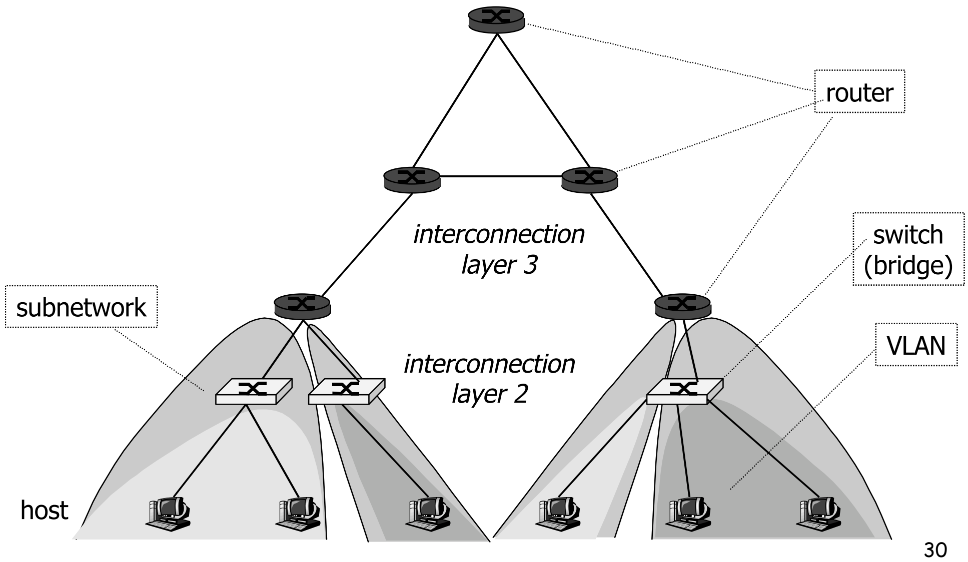
LAN stack

- **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"



29

Interconnection structure - layer 2 and 3



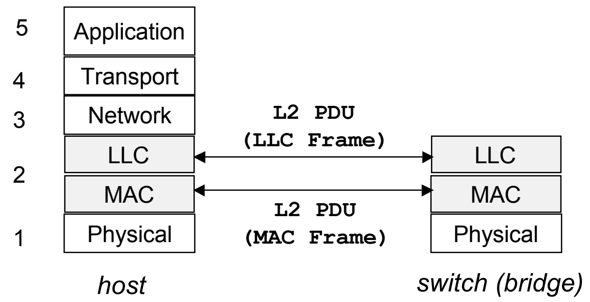
30

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

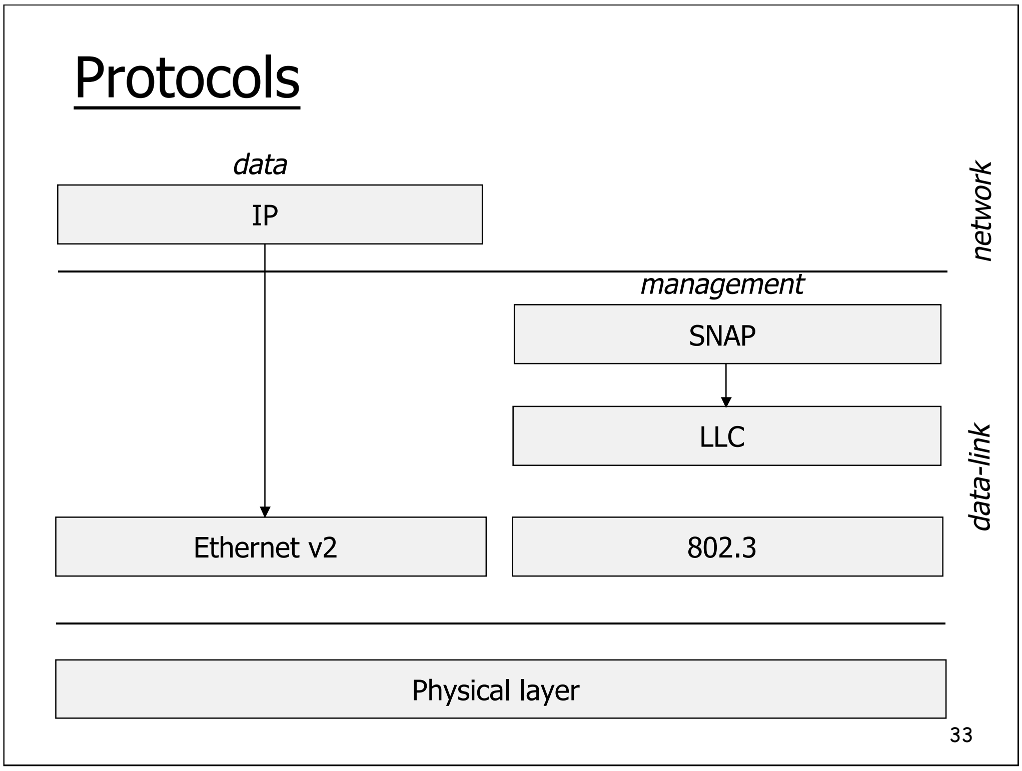
31

Protocol architecture



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

32

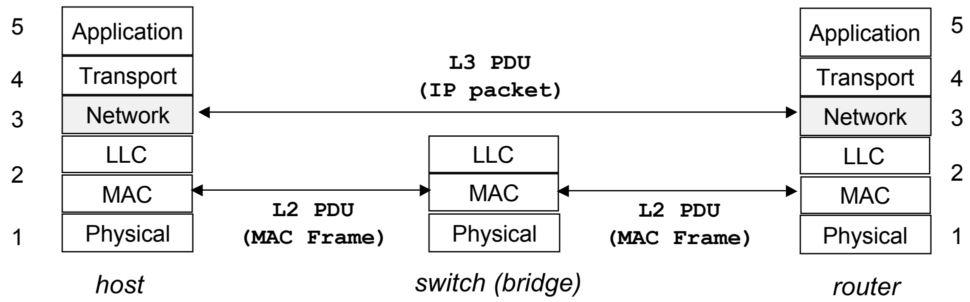

Protocols
data
IP
network
management
SNAP
LLC
data-link
Ethernet v2
802.3
Physical layer
33

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

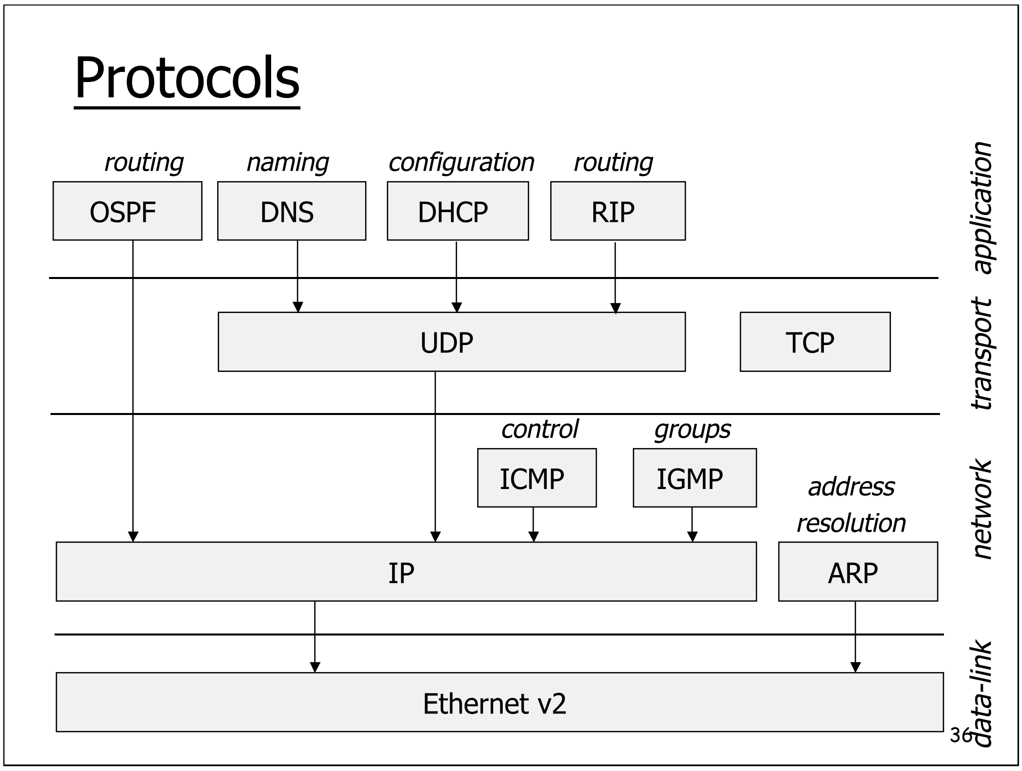
34

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)

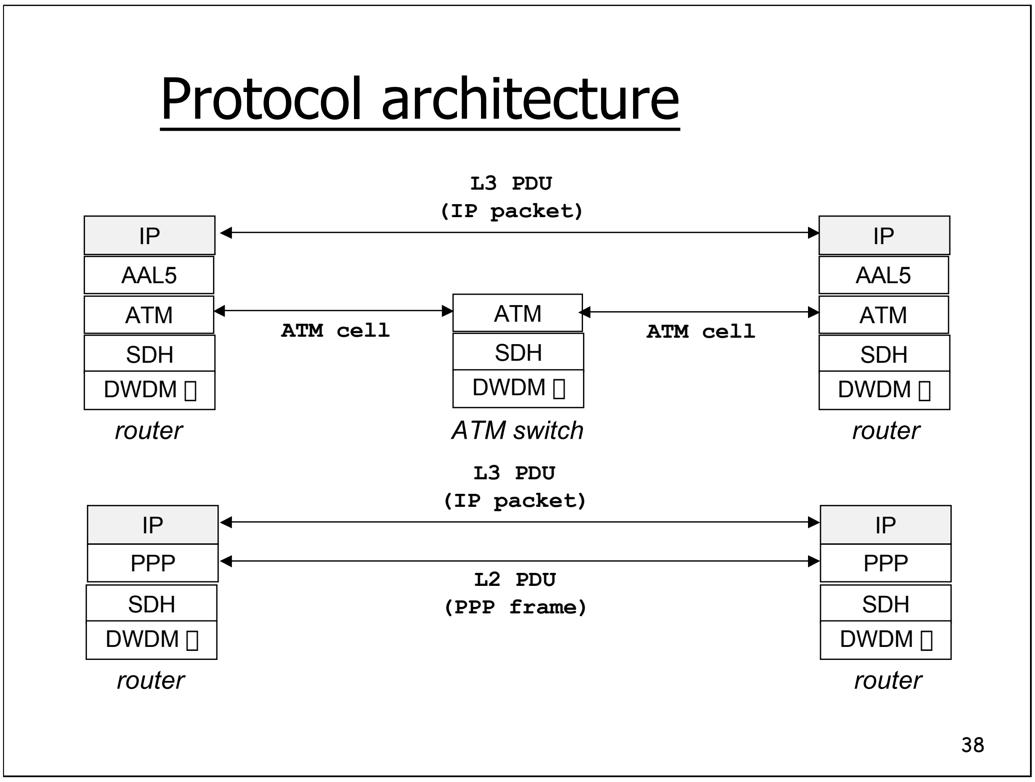
35


Protocols
routing
naming
configuration
routing
OSPF
DNS
DHCP
RIP
application
UDP
TCP
transport
control
groups
ICMP
IGMP
address
resolution
network
IP
ARP
Ethernet v2
data-link
36

Overlaid stacks? Long-haul links

- Fiber at physical layer (SONET/SDH)
  - Dense Wave Division Multiplexing (DWDM)
    - one color of the light □
- 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

37


Protocol architecture
L3 PDU
(IP packet)
IP
AAL5
ATM
SDH
DWDM
router
ATM cell
ATM
SDH
DWDM
ATM switch
ATM cell
IP
AAL5
ATM
SDH
DWDM
router
L3 PDU
(IP packet)
IP
PPP
SDH
DWDM
router
L2 PDU
(PPP frame)
IP
PPP
SDH
DWDM
router
38

Performance

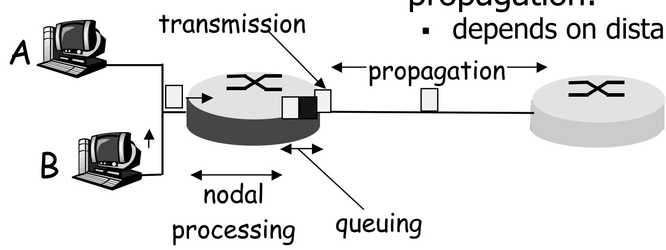
- Bit Rate (débit binaire) of a transmission system
  - bandwidth, throughput
  - number of bits transmitted per time unit
  - units: b/s or bps, kb/s = 1000 b/s, Mb/s = 10e+06 b/s, Gb/s=10e+09 b/s
  - OC3/STM1 - 155 Mb/s, OC12/STM4 - 622 Mb/s, and OC48/STM-16 - 2.5 Gb/s, OC192/STM-48 10 Gb/s
- Latency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

39

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

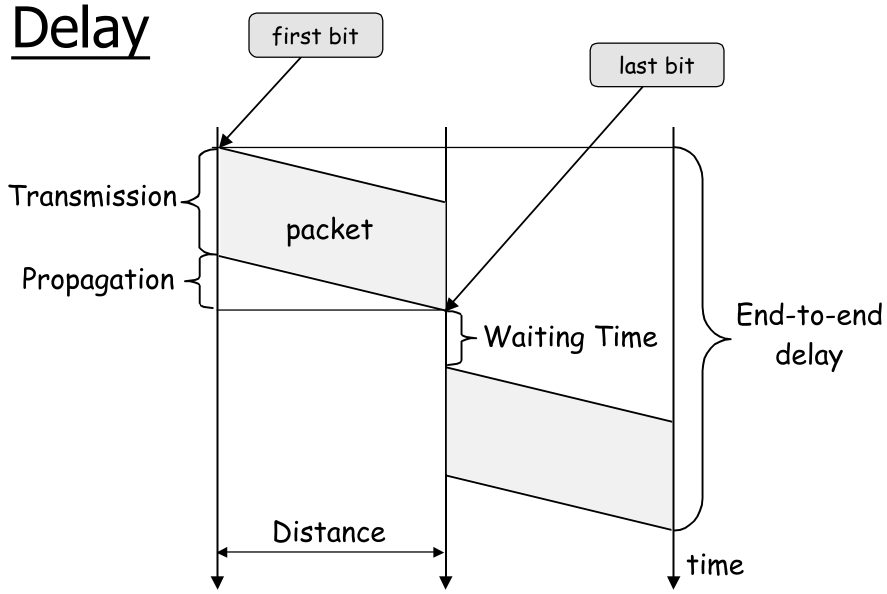
40

As a packet travels from one node (host or router) to the subsequent node (host or router) along this path, the packet suffers from several different types of delays at each node along the path. The most important of these delays are the nodal processing delay, queuing delay, transmission delay, and propagation delay; together, these delays accumulate to give a total nodal delay.

Processing Delay

The time required to examine the packet's header and determine where to direct the packet is part of the processing delay. The processing delay can also include other factors, such as the time needed to check for bit-level errors in the packet that occurred in transmitting the packet's bits from the upstream router to router A. Processing delays in high-speed routers are typically on the order of microseconds or less. After this nodal processing, the router directs the packet to the queue that precedes the link to router B. (In Section 4.6 we will study the details of how a router operates.)

Delay


first bit
last bit
Transmission
packet
Propagation
Waiting Time
End-to-end
delay
Distance
time

41

Performance

- Latency
  - Latency = Propagation + Transmission + Wait
  - Propagation = Distance / Speed
    - copper : Speed = 2.3□108 m/s
    - glass : Speed = 2□108 m/s
    - Transmission = Size / BitRate
- 5 □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

42

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 = 2e+08 m/s)?

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
latency	?	?	?	?
	modem	*satellite*	*LAN*	*Hippi*

43

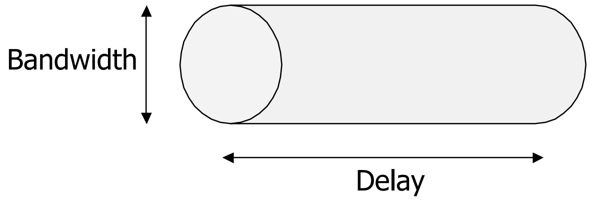
Example

- At time 0, computer A sends a packet of size 1000 bytes to B; at what time is the packet received by B (speed = 2e+08 m/s)?

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
latency	800.1 ms	108 ms	0.81 ms	8.1 μs
	modem	*satellite*	*LAN*	*Hippi*

44

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!

45

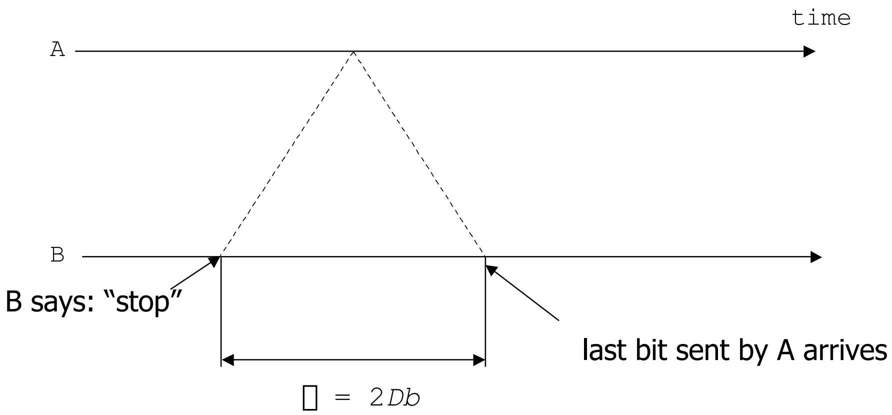
Bandwidth-Delay Product

- 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

46

Bandwidth-Delay Product

- Consider the scenario :



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

47

As an illustration of the effect of propagation, consider the scenario above.

The number □ is called the bandwidth-delay product. It expresses the number of bits in the pipe. We will find it important in the rest of the lecture - the performance of protocols depends on this parameter.

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 *T1*, then A retransmits

48

This example is a simple protocol, often used, for repairing packet or message losses. The idea is simple

- identifiy all packets with some number or some other means

- when you send one packet, wait until you receive a confirmation

- after some time, if no confirmation arrives, consider that the packet has been lost and retransmit.

Compute the maximum throughput of this protocol, assuming the source has an infinite supply of packets to send, the destination generates the confirmation instantly, and the bit rate of the channel is constant.

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

49

This example is a simple protocol, often used, for repairing packet or message losses. The idea is simple

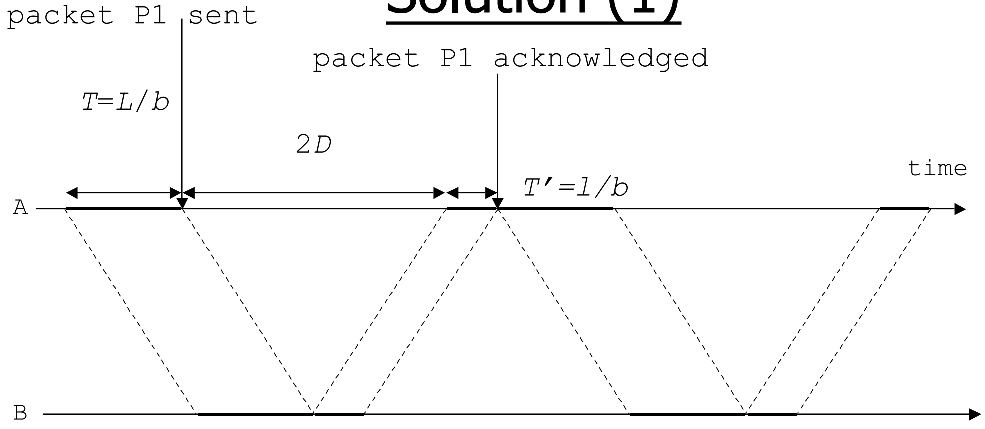
- identifiy all packets with some number or some other means

- when you send one packet, wait until you receive a confirmation

- after some time, if no confirmation arrives, consider that the packet has been lost and retransmit.

Compute the maximum throughput of this protocol, assuming the source has an infinite supply of packets to send, the destination generates the confirmation instantly, and the bit rate of the channel is constant.

Solution (1)



cycle time = $T + 2D + T'$

useful bits per cycle time = L

throughput = Lb / ($L + l + 2Db$)= b /(□ + □/L)

with □=(L+l)/L=overhead and □=2Db=**bandwidth-delay product**

50

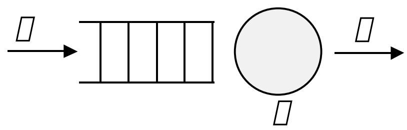
Solution (2)

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
	modem	*satellite*	*LAN*	*Hippi*
□$=2Db$	2 bits	200 000 bits	200 bits	200 bits
throughput = *b* □	99.98%	3.8%	97.56%	97.56%

51

Waiting time

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



$$N = \frac{\rho}{(1-\rho)} \qquad T = \frac{1}{\mu(1-\rho)} \qquad T = \frac{N}{\lambda}$$

52

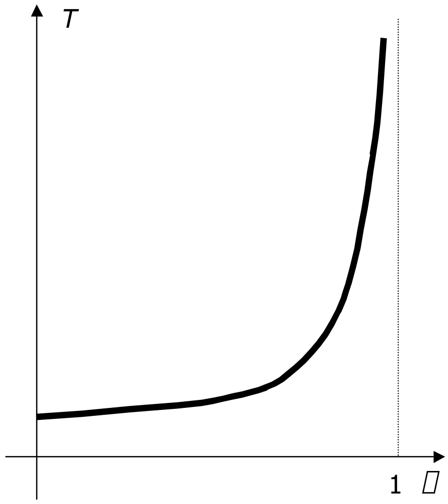
Waiting time

- 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

53

Waiting time



54

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

55