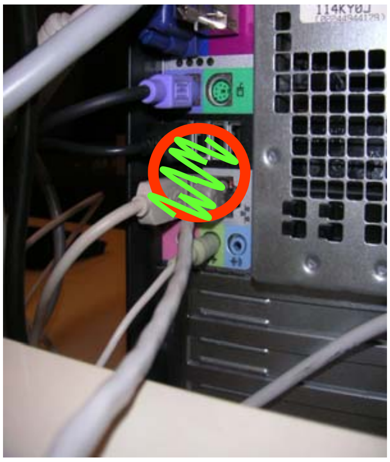
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

The TCP/IP Architecture

Jean-Yves Le Boudec

Fall 2009



* Objective

- Understand Layered Model of Communication Systems
  - Know the function of every layer
  - Why and What – we will see the How in other modules.
- Understand basic performance issues

* Contents

1. The Layered model of the Internet, Packet Switching
2. Protocol, service and other definitions
3. All you need to know about the physical layer

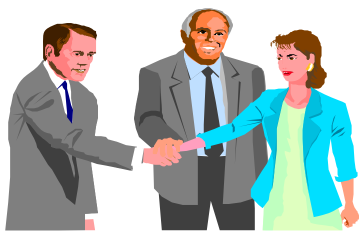
Why was TCP/IP invented ?

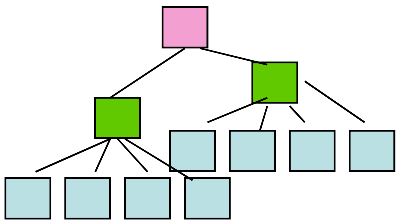
- By « network » we mean one of the following
  - Internet
  - SMS
  - telephone
- We focus on the Internet. Similar concepts are used in other networks, with a different terminology.
- Communication networks use a *layered approach*

Layered Architecture

- TCP/IP is a layered architecture
- Why ?
  - Divide and conquer – make things manageable
- What is it ?

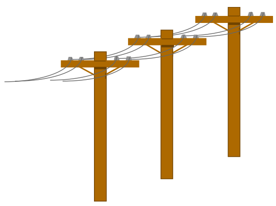
Communication
- Application
- Transport



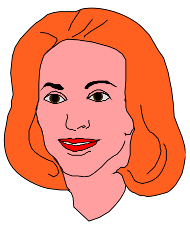


Interconnection
- Network
- Links

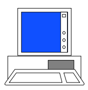
Distance
- Physical



Application Layer helps people and machines communicate



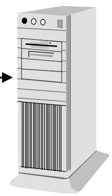
user clicks:
http://www.zurich.ibm.com/RZ.html



- Uses well defined "protocols" (set of rules and messages)
  ex: HTTP
- In the simplest case, involves 2 computers
- If you write an application that uses the network, you define your own "Application Layer"

Web server

IP addr = 193.5.61.131
GET www.zurich.ibm.com/RZ.html



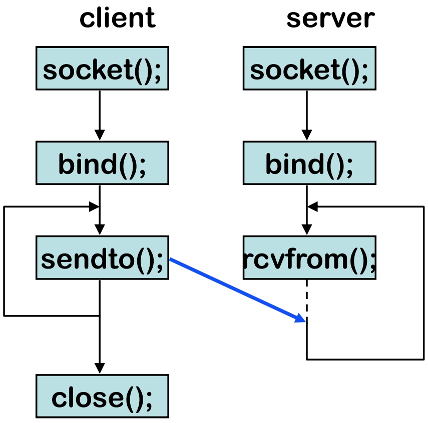
data (HTML page)

Transport Layer helps Application layer

- Transport Layer provides programming interface to the application layer
  - Relieve programmer from repetitive tasks
- In TCP/IP there are two main transport protocols
  - UDP (User Datagram Protocol)
    - offers a datagram service to the application (unit of information is a message)
    - Unreliable (message may be lost)
    - No sequence guarantee
  - TCP (Transmission Control Protocol)
    - Reliable: if some data is lost somewhere, TCP retransmits it
    - Stream service: the data is delivered at destination in the order it was sent by source (sequence guarantee)
    - (but unit of information is a byte; grouping of data into blocks may be different at destination than at source)
- application may use UDP or TCP depending on requirements
- programming interface is called the socket API

* UDP is used via a Socket Library

- The socket library provides a programming interface to TCP and UDP
- The figure shows toy client and server UDP programs. The client sends one string of chars to the server, which simply receives (and displays) it.
  - socket() creates a socket and returns a number (=file descriptor) if successful
  - bind() associates the local port number with the socket
  - sendto() gives the destination IP address, port number and the message to send
  - recvFrom() blocks until one message is received for this port number. It returns the source IP address and port number and the message.



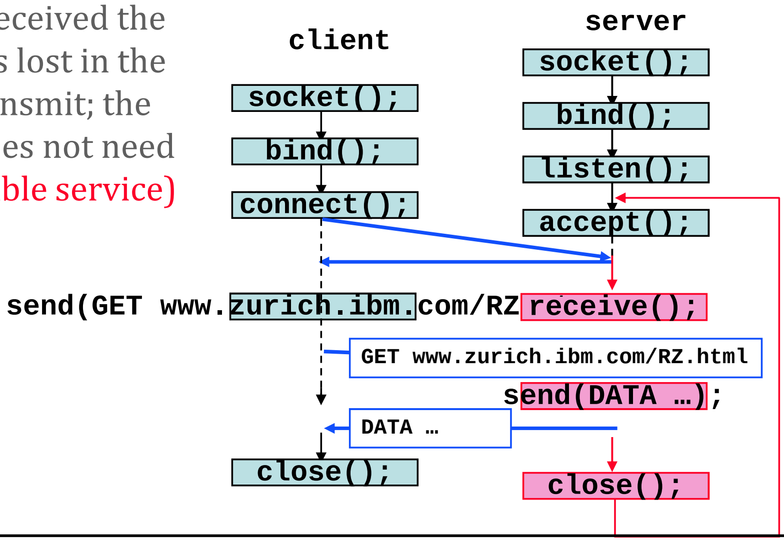

```
% ./udpClient <destAddr> bonjour les amis
%
```

```
% ./udpServ &
%
```

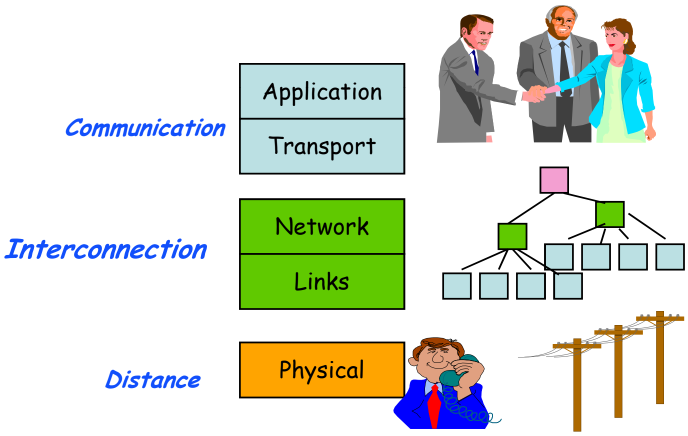
* TCP is used via a socket library

Example of a toy program (we will see more details later)

- When client program uses the TCP send() function of the socket library, it knows that TCP will do all the necessary jobs to check that the server has indeed received the message. If some data is lost in the network, TCP will retransmit; the application program does not need to bother about it (reliable service)
- When client program uses the TCP send() function of the socket library, the library may break the message into pieces and deliver the pieces separately, but in sequence (stream service)

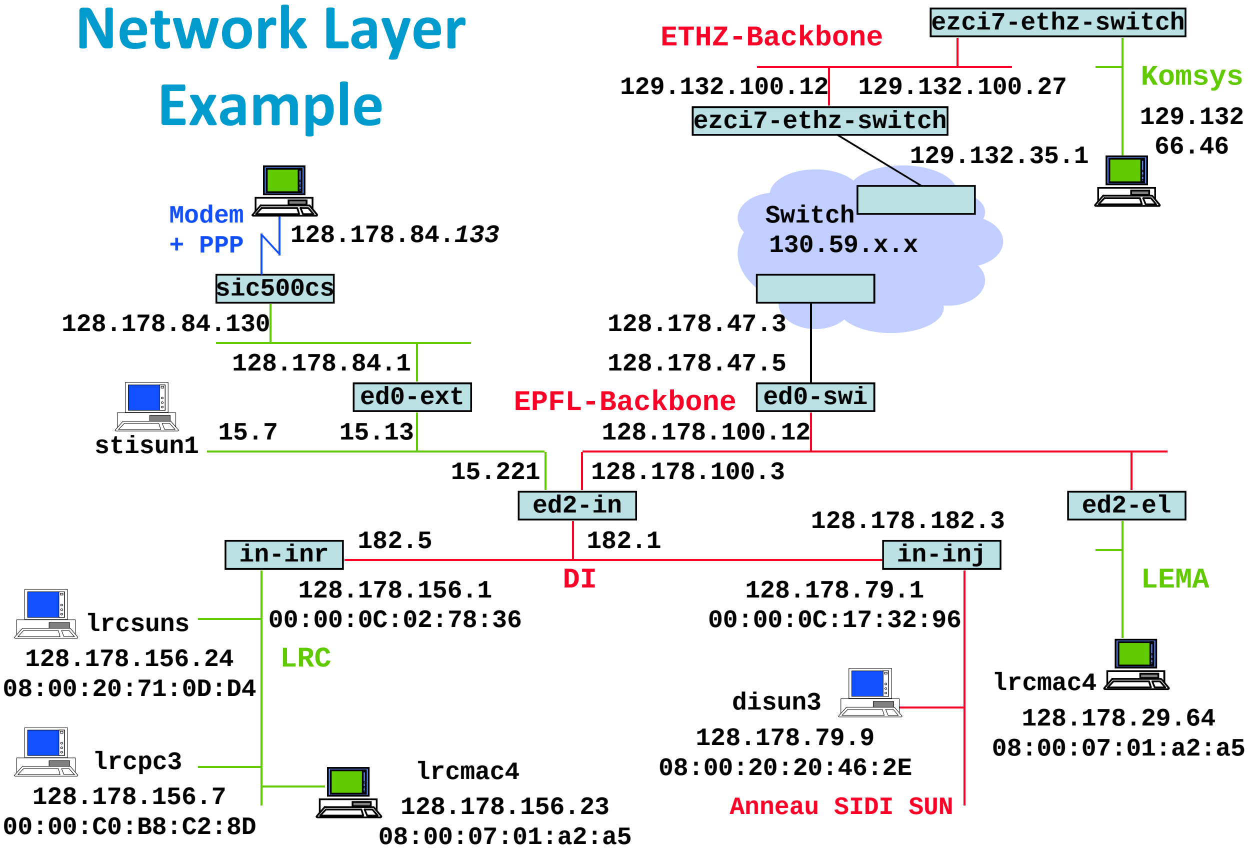


Network Layer provides full connectivity

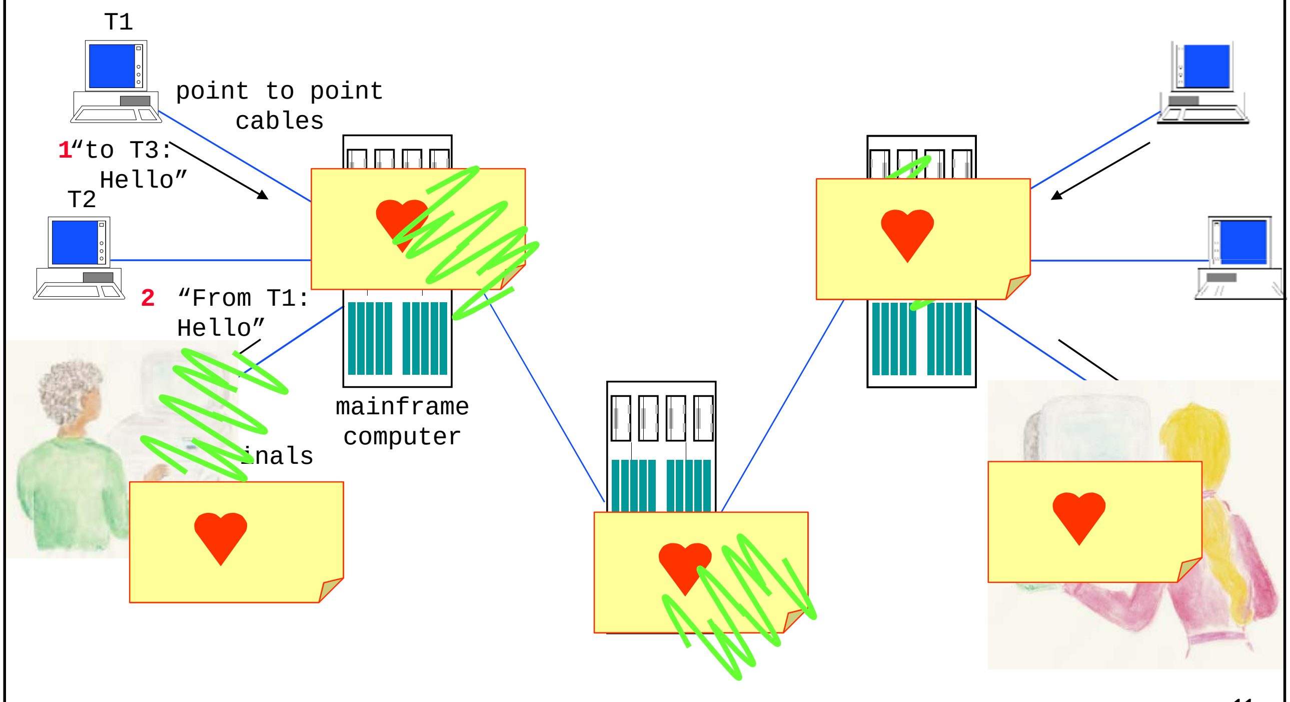


- uses *intermediate systems* « routers », not visible to users
- the internet uses *packet switching*
  - all information is cut into pieces · 1500B
  - all hosts and routers have IP addresses and use them to communicate

Network Layer Example

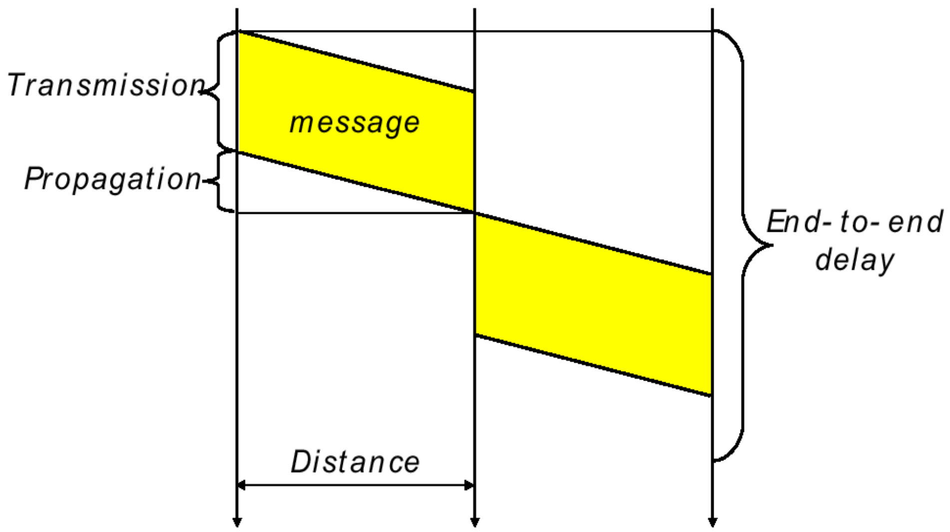


* The Very First Computer Networks (Bitnet, SNA) used Store and Forward

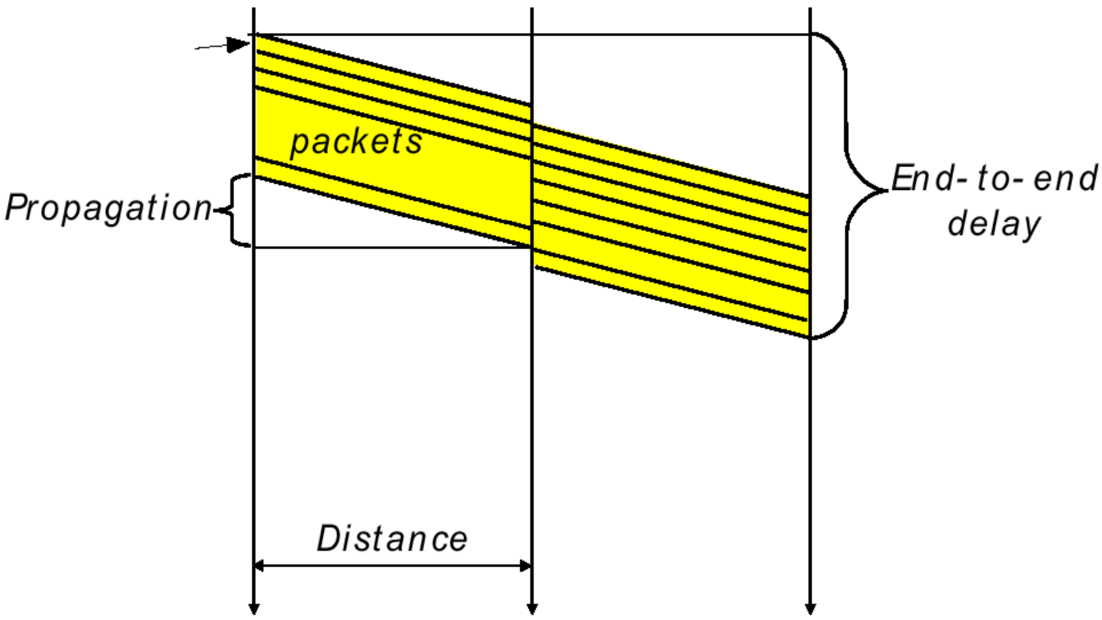


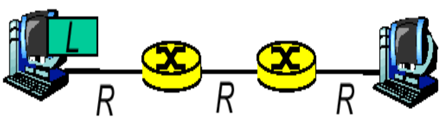
The Network Layer of the Internet uses Packet Switching -- Why ?

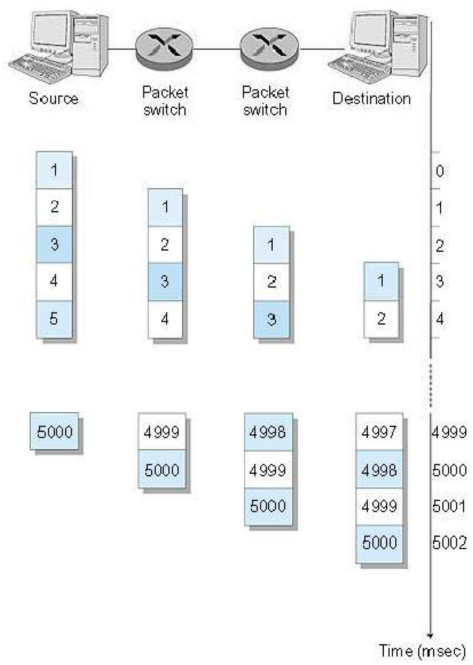
- Time diagram of store and forward



Packet Switching Reduces Delay and Queue Lengths







- 1 MB, link rate 1.5 Mb/s

  Delay: 15 sec
  Buffer: 1024 KB

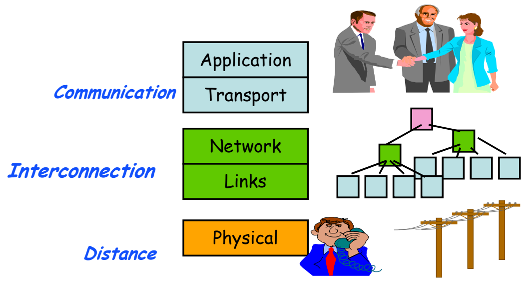
- With packet size 1500B

  Delay: 5 sec
  Buffer : 1.5 KB

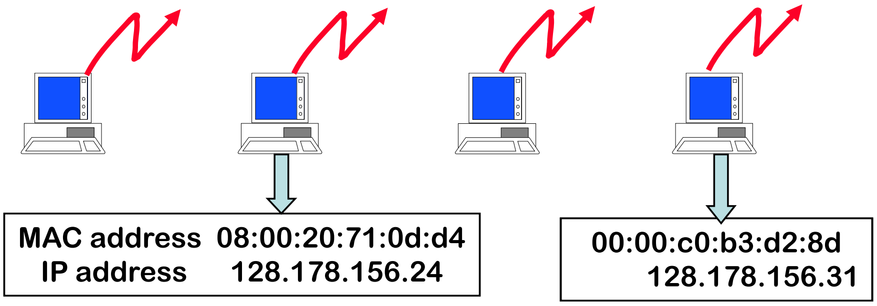
Circuit vs Packet Switching

- 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 or X.25, and even ATM. Packet switches have queues.
- **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.
- 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. Intermediate systems are known under various terms depending on the

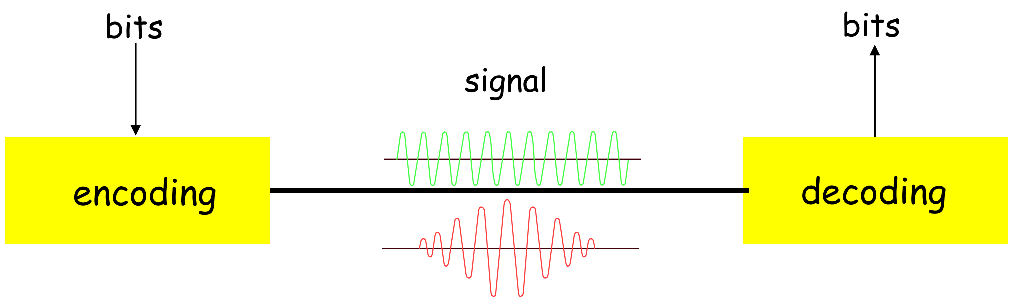
Link Layer defines how several hosts can use the same radio frequency



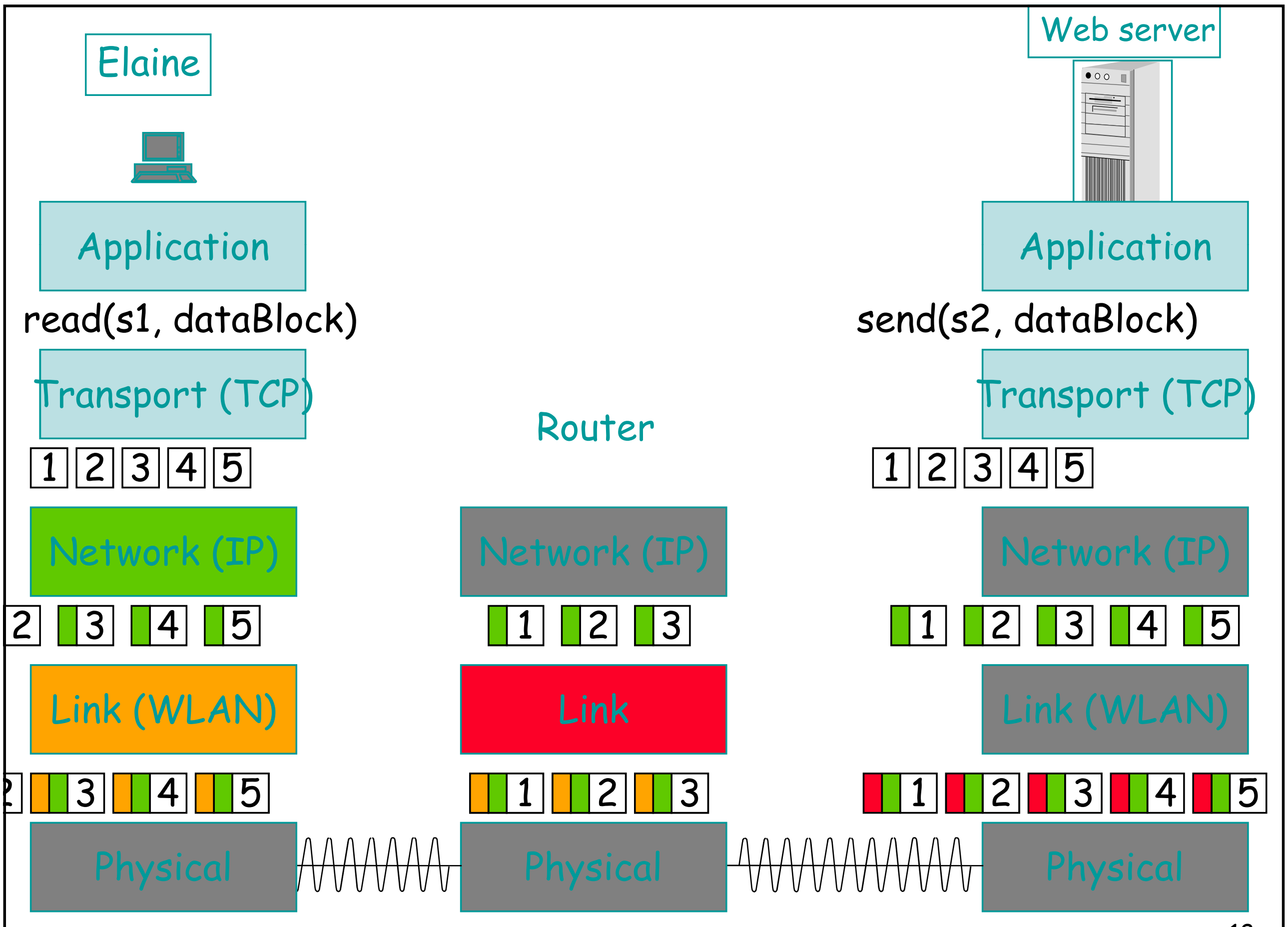
- Every system has a hardware address (= MAC address)
- Plus an IP address



Physical Layer Transforms Bits and Bytes into Electromagnetic Waves



- Encoding of bits as physical signals
- Is technology specific: there are several Ethernet physical layers, several WLAN 802.11 physical layers
- Acoustic instead of electromagnetic is used under water


Elaine
Web server
Application
Application
read(s1, dataBlock)
send(s2, dataBlock)
Transport (TCP)
Transport (TCP)
Router
1 2 3 4 5
1 2 3 4 5
Network (IP)
Network (IP)
Network (IP)
2 3 4 5
1 2 3
1 2 3 4 5
Link (WLAN)
Link
Link (WLAN)
3 4 5
1 2 3
1 2 3 4 5
Physical
Physical
Physical

Why do we call that stuff *Layers* ?

- By the nature of communication, we have two types of interactions: between peers – and between layers

  This is why we call this stuff a set of *layers*

2. Protocol, service and other fancy definitions

- Peer entities
  - two (or more) instances of the same layer
- Protocol and a PDU:
  - the rules of the operation followed by peer entities
  - the data exchanged is called PDU (Protocol Data Unit)

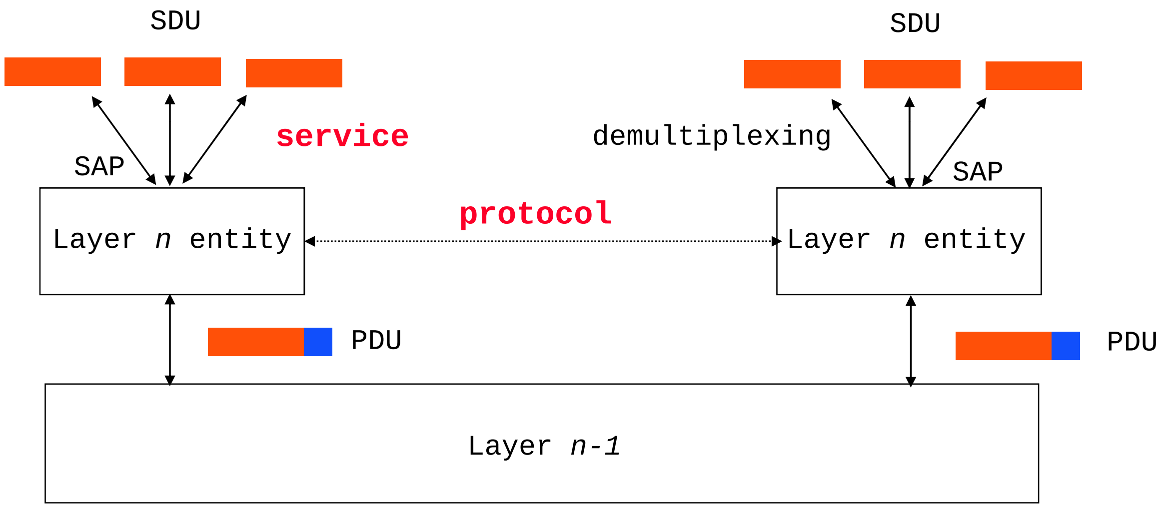
  there is one protocol (or more) at every layer

  Examples of protocols are: TCP; UDP;IP; Ethernet

- Service and a SDU
  - the interface between a layer and the layer above - SAP (Service Access Point)
  - the interface data is called a SDU (Service Data Unit)
- Connection
  - a protocol is connection oriented if the peer entity must be synchronized before exchanging useful data (connection set up); otherwise it is connectionless.

    The telephone system is connection oriented: before A can send some information to B, A has to call B (or vice versa) and say "hello". The postal (mail) system is connectionless. If A wants to send some information to B, A can write a letter and mail it, even if B is not ready to read it.

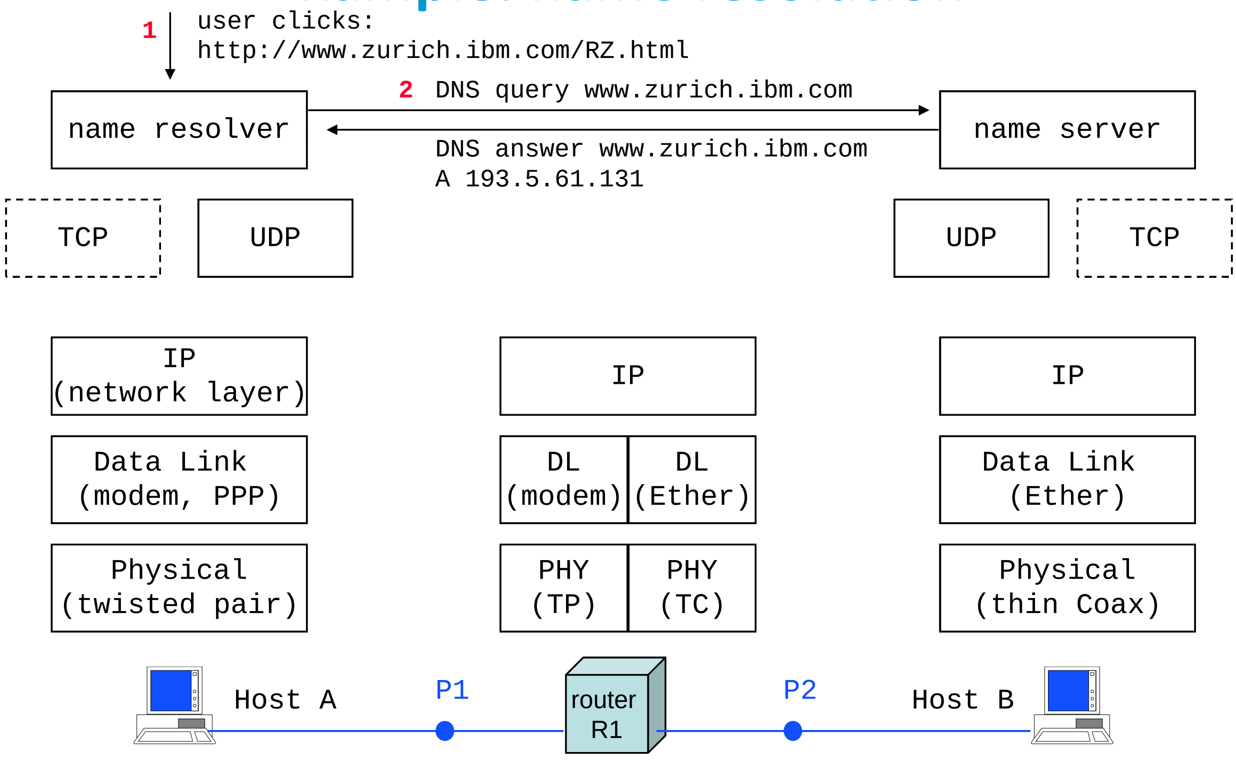
Protocol Architecture



- Networking functions are structured as a layered model:
- - layer *n* communicates with other layer *n* entities using layer *n* **PDUs**
- - layer *n* uses the **service** of layer *n-1* and offers a service to layer *n+1*.
- - entities at the same layer are said **peer entities**
- - operation rules between peer entities are called **protocol**

- Layering of protocol entities is reflected by the term of a **protocol stack.**

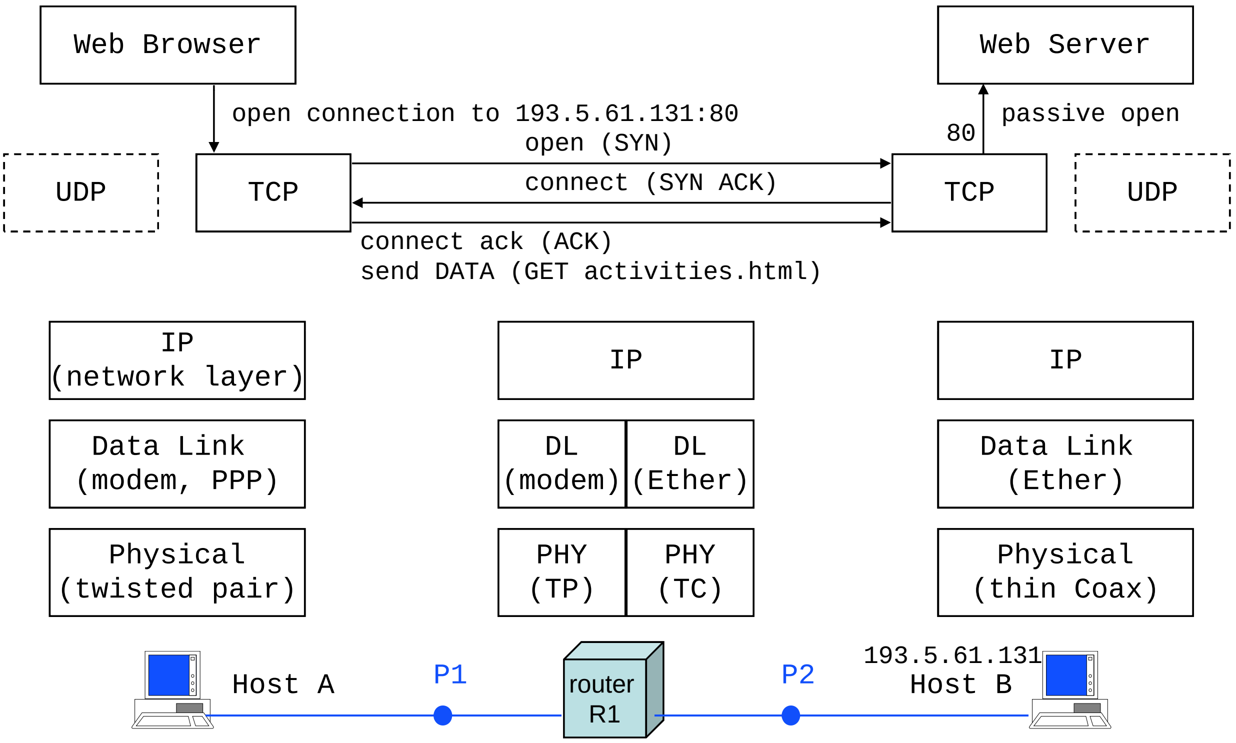
Example: name resolution



- Flow 2 illustrates the **query/response protocol** of the Domain Name System (DNS). The **name resolver** and the **name server** are two application programs, probably C programs making calls to the socket library. The programs use UDP, which is the non-reliable transport protocol in the TCP/IP stack.

- Let us apply the terminology on this example.
- “name resolver” uses the UDP service: it creates a request to send data to “name server”. “name server” is identified by its IP address (for example: 128.178.15.7). “name resolver” also knows that “name server” can be reached by means of port 53 (a well known convention used in the Internet). The SDU is the request, with the data. The transport-PDU is called a datagram. It contains the data, the address and the port numbers. It is identified by 2 in the figure.
- UDP creates a request to IP to send data to the name server machine identified by the IP address 128.178.15.7. The network-PDU is called an IP packet. It contains the UDP datagram plus the IP addressing information (and some other information, see later).
- IP creates a request to send a data frame over the modem. The modem card creates a data-link PDU, called a modem “frame”. The frame contains the IP packet, maybe compressed. Then the data link layer requests transmission of the frame; the physical layer SDU is a bit. The physical layer PDU is an electromagnetic signal.
- At the router
  - the data frame is received, understood as an IP packet
  - IP reads the IP destination address (128.178.15.7) and decides to forward it over its Ethernet interface
  - IP creates a request to send the data frame over the Ethernet. An Ethernet frame is created and sent to the name server machine

An example with TCP



- Here is a second example.

- A web browser always uses TCP for communication with a web server.
- The web browser starts by requesting from the transport layer the opening of a connection for reliable data transport. TCP opens a connection to the peer entity at the web server machine by starting a 3-way handshake. If the connection can successfully be opened, then data can flow between the web client and server. TCP monitors missing packets and retransmits them as appropriate.

- The web browser and server can thus assume that they have a reliable data pipe between them transporting data in sequence and without errors, at least as long as the TCP layer does not close the connection.

- TCP is connection oriented. What is shown is the connection setup phase. TCP uses IP, which is connectionless. UDP is connectionless.

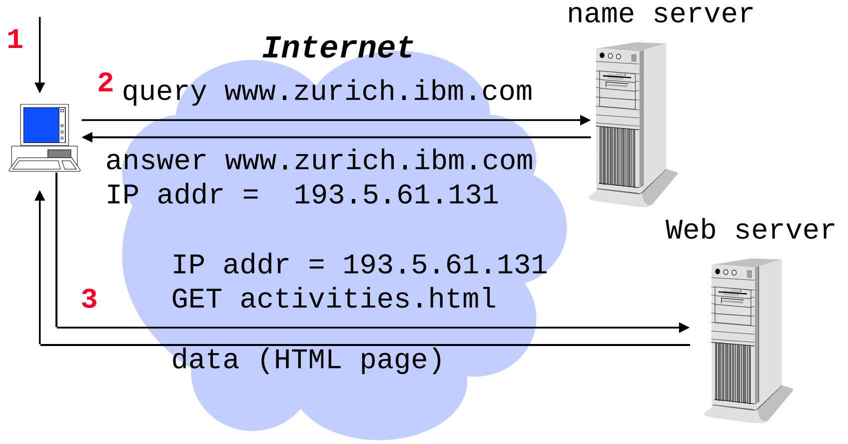
- An observer at P1 or P2 would see the beginning of the message between web clients and servers only in the third data frame.

What is the Client-Server model?

distributed applications use the client-server model

- **server** = program that awaits data (requests) to be sent to it
  - interprets a request and send a response
- **clients** send data (requests) to servers
  - wait for a response

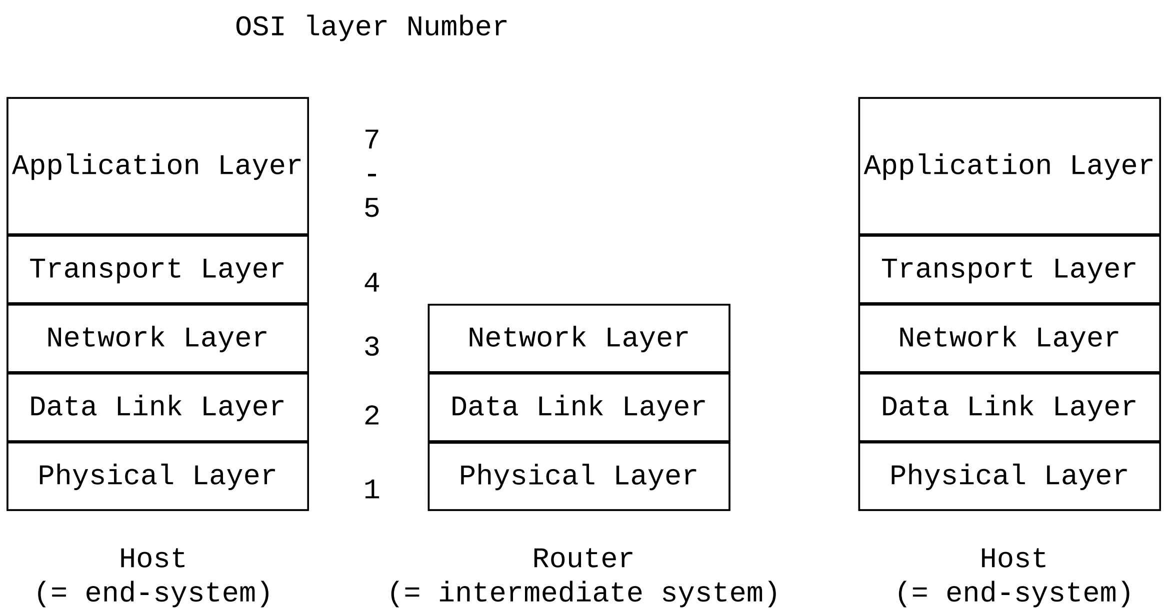
```
user clicks:
http://www.zurich.ibm.com/activities.html
```



- We use the terms “client” and “server” in the following sense.

- When two entities say A and B, want to communicate, there is a boostrap problem: how can you initialize both A and B such that the communication can take place. One solution is to manually start A, then B, but this defeats the purpose of networking. The only way we have found so far is to request that one of the two, say B, is started and immediately puts itself in a **listening** position. We say that B is a server. A system, such as A, which talks to B, is said to be a client.
- Being a server or a client is relative to a given protocol. For example, consider the application level protocol called FTP (file transfer protocol). The FTP server is a machine that waits for other machines to send requests for logging in. When an FTP client has contacted an FTP server, then after an initial navigation phase, the FTP client has to wait for the FTP server to open a connection back to the client (try it !). In that interaction, the FTP client is a TCP server, namely, a machine which waits for some other machine to open a TCP connection.

The TCP/IP Architecture



- An **architecture** is a set of external behaviour specifications for a complete communication system. It describes protocols and services, but not how to implement them.
- The **TCP/IP Architecture**, or the **Internet Architecture** is described by a collection of Internet standards, published in documents called RFCs (Requests For Comments), available for example from ftp://ftp.switch.ch/standard.
- The picture shows all the layers of the Internet Architecture. There exists, inside every layer, a number of protocols that we will discover in this course.
- *There exist other architectures, each of them having a different set of layers and names for layers. There are:*
  - *proprietary architectures: SNA (IBM), Decnet (Digital), AppleTalk (Apple), XNS (Xerox), UUCP (Unix internal protocols), etc*
  - *the ITU architecture defines public networks for telephony, telex, fax, data networks (X.25, Frame Relay, mail and directory services) and ATM*
  - *the IEEE LAN architecture defines layers 1 and 2 for local area networks. We will see some details later.*
  - *The **OSI (Open Systems Interconnection) architecture** is an official standard, similar to the TCP/IP architecture, but is now obsolete. It has 7 layers instead of 5. However, the OSI model is still frequently used to describe systems; this is why the application layer is often called "layer 7".*
- *Today, the TCP/IP architecture has become dominant, so this is the only one we will study in detail. The ITU architecture (Frame Relay and ATM) does also play an important role and we will study it at the end of the course.*
- *Different architectures do not interoperate by themselves at the protocol level. For example, the OSI transport protocols are not compatible with TCP or UDP.*

Test Your Understanding

- **Q** Say *what* each layer of the TCP/IP architecture is doing.

solution

Questions

- When data is transferred from a web server to a PC, TCP is run
  - A. Only in the server and the PC
  - B. Only in the server
  - C. In the server, the PC and the routers in-between if there is any
- What is client ? A server ?

Questions

- An application program normally uses TCP or UDP to send data
  - True
  - False

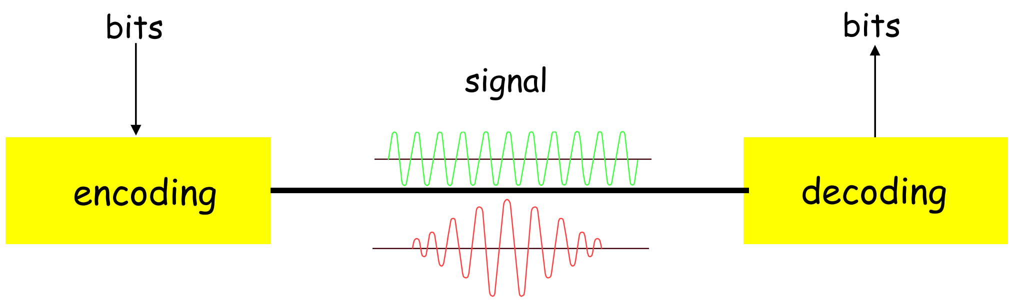
- What is the difference between TCP and UDP ?

Questions

- What is the difference between an IP address and a domain name ?

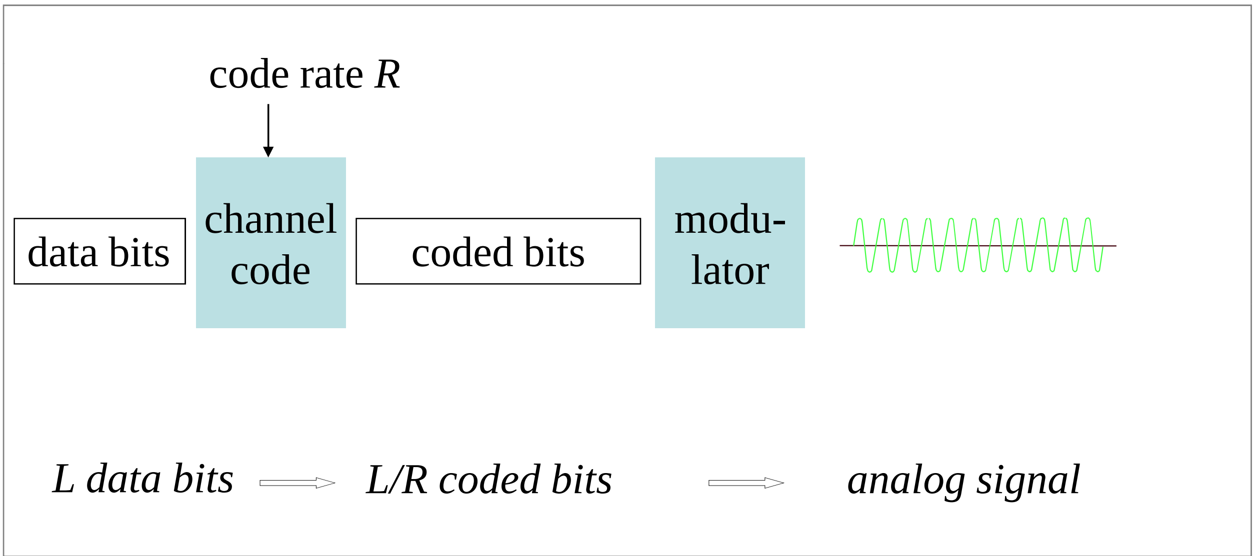
- In a IP packet there is
  - A. the IP address of the source
  - B the IP address of the destination
  - C the name of the destination
  - D the name of the source

* 3. Physical Layer



- Encodes of bits as physical signals
- Is technology specific: there are several Ethernet physical layers, several WLAN 802.11 physical layers

* PHY does Coding plus Modulation



- channel code adds redundancy for coping with noise and errors

Bit Rates

- **Bit Rate** of a transmission system = number of bits transmitted per time unit; is measured in **b/s**, 1 kb/s = 1000 b/s, 1 Mb/s = 10^6 b/s, 1Gb/s=10^9 b/s

- *The bit rate of a channel is the number of bits per second. The bandwidth is the width of the frequency range that can be used for transmission over the channel. The bandwidth limits the maximal bit rate that can be obtained using a given channel. The purpose of information theory is to find the best possible bit rate on a given channel.*

  For example: Shannon-Hartley law: $C_{max} = B \log_2 (1 + S/N)$, with B = bandwidth (Hz), S/N = signal to noise ratio (not expressed in dB); for example: telephone circuit: B = 3 kHz, S/N = 30 dB, C_{max} = 30 kb/s

  In computer science, many people use "bandwidth" instead of "bit rate". In communication theory, this is regarded as a gross mistake.

- Practical Bit Rates:
  - modem: 2.4 kb/s to 56kb/s
  - ADSL line: 124 kb/s to 10 Mb/s
  - Ethernet: 10 Mb/s, 100 Mb/s, 1Gb/s
  - Wireless LAN: 1 to 50 Mb/s
  - ATM: 2 Mb/s to 622 Mb/s
  - Optical carriers: 155 Mb/s to 49 Gb/s

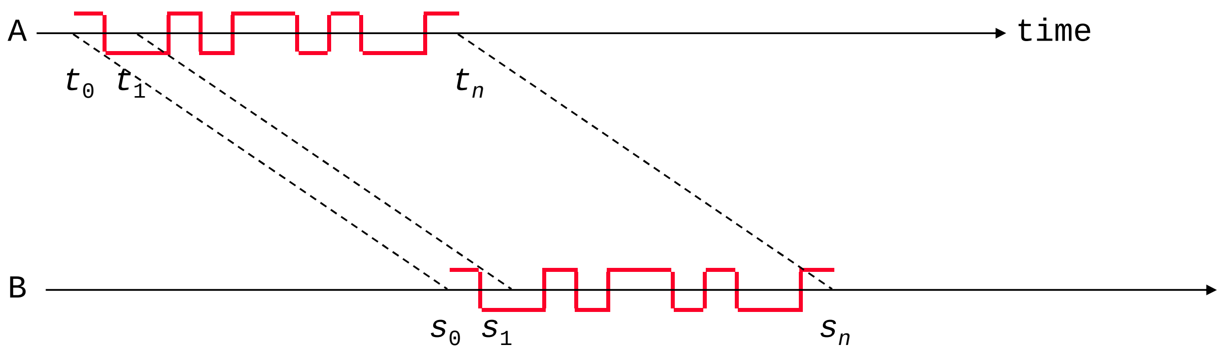
Transmission Time

- Transmission time = time to send *x* bits at a given bit rate
- **Q.** time to send 1 MB at 10 kb/s = ?

solution

Propagation

- **Propagation** between A and B = time for the head of signal to travel from A to B



$s_i - t_i = D$ (propagation delay)

- $D = d/c$, where d = distance, c =signal celerity (speed of light)
  copper: c= 2.3e+08 m/s;   glass: c= 2e+08 m/s;
- Rule of thumb: 5 μs/km; example: earth round trip in fiber: D = 0.2 s
- time through circuits also adds to propagation delays
- Lausanne - Concarneau over acoustic channel. D = ???   *A.1 hour*

Throughput

- **Throughput** = number of useful data bits / time unit
- It is *not* the same as the bit rate. Why ?
  - protocol overhead: protocols like UDP use some bytes to transmit protocol information. This reduces the throughput. If you send one byte messages with UDP, then for every byte you create an Ethernet packet of size 1 + 8 + 20 + 26 = 53 bytes, thus the maximum throughput you could ever get at the UDP service interface if you use a 64 kb/s channel would be 1.2 kb/s.
  - protocol waiting times: some protocols may force you to wait for some event, as we show on the next page.
- Same units as a bit rate

  b/s, kb/s, Mb/s

Pigeon outruns South African ADSL

11 September 2009 | 14:28

A South African information technology company has proved it's faster for them to send data by carrier pigeon than using the country's leading internet provider.

A South African information technology company has proved it's faster for them to transmit data by carrier pigeon than to send it using Telkom, the country's leading internet service provider.

Internet speed and connectivity in Africa's largest economy are poor because of a bandwidth shortage. It is also expensive.

Winston the pigeon has easily outpaced South Africa's leading broadband netw ork it moving data (AAP)

An 11-month-old pigeon, Winston, took one hour and eight minutes to fly the 80 km (50 miles) from Unlimited IT's offices near Pietermaritzburg to the coastal city of Durban with a data card strapped to its l

Including downloading, the transfer took two hours, six minutes and 57 seconds – the time it took for only four percent of the data to be transferred using a Telkom line.

Pigeon's throughput ?
A:

Telecom's throuput ?
A:

Pigeon outruns South African ADSL

11 September 2009 | 14:28

A South African information technology company has proved it's faster for them to send data by carrier pigeon than using the country's leading internet provider.

Winston the pigeon has easily outpaced South Africa's leading broadband netw ork it moving data (AAP)

A South African information technology company has proved it's faster for them to transmit data by carrier pigeon than to send it using Telkom, the country's leading internet service provider.

Internet speed and connectivity in Africa's largest economy are poor because of a bandwidth shortage. It is also expensive.

An 11-month-old pigeon, Winston, took one hour and eight minutes to fly the 80 km (50 miles) from Unlimited IT's offices near Pietermaritzburg to the coastal city of Durban with a data card strapped to its l

Including downloading, the transfer took two hours, six minutes and 57 seconds – the time it took for only four percent of the data to be transferred using a Telkom line.

Pigeon's throughput ?
A:
4 000 000 000 * 8 bits / 7293 s
=
4.39 Mb/s

Telecom's throuput ?
A:
0.04 * 4.39 Mb/s
=
176 kb/s

* Example. The Stop and Go Protocol

- 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;

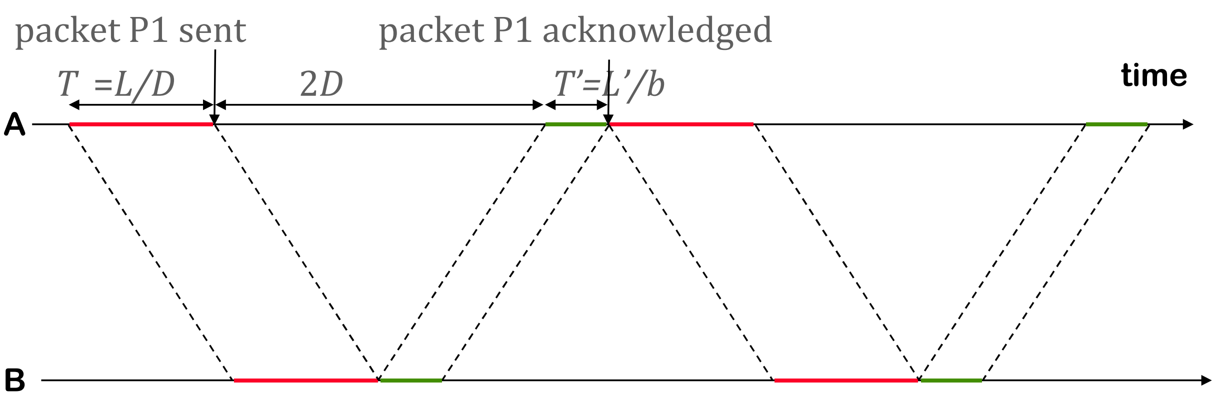
  A waits for acknowledgement before sending a new packet; if no acknowledgement comes after a delay *T1*, then *A* retransmits

Example: 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 delay = *D*
- processing time is negligible

* Solution: The Stop and Go Protocol



Cycle Time $= T + 2D + T'$

useful bits per cycle time $= L$

throughput $= \frac{L}{T+2D+T'} = \frac{b}{1+\frac{L'}{L}+\frac{2Db}{L}}$

overhead

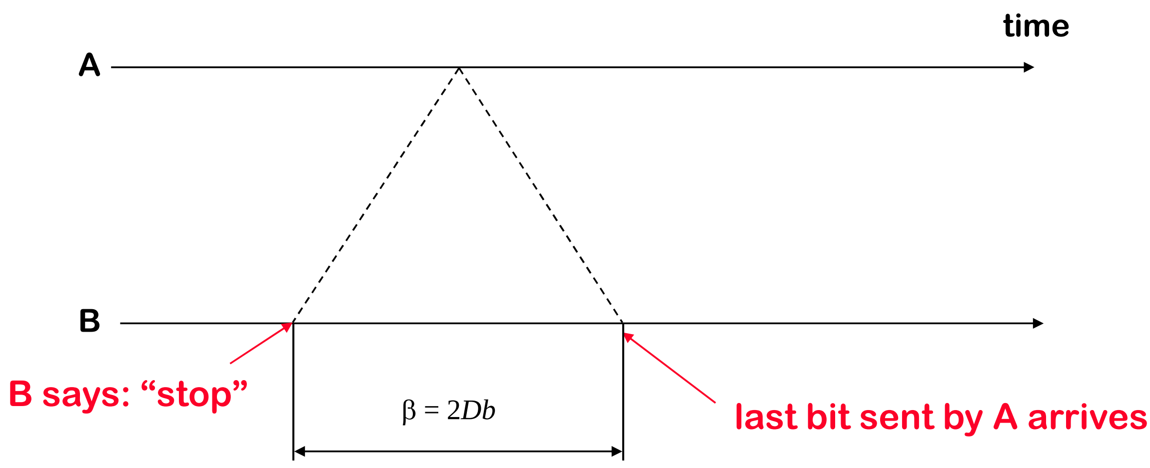
« bandwidth »-delay product

* Solution (2)

distance	20 km	20000 km	2 km	20 m
bit rate	10 kb/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
	GSM	*WAN*	*WiMax LAN*	*Gb LAN*
bw delay product	2 bits	200 000 bits	200 bits	200 bits
throughput = b ×	99.98%	3.8%	97.56%	97.56%

"Bandwidth-Delay" Product

- Consider the scenario :



- β = maximum number of bits B can receive after saying stop
- large β means: delayed feedback

Test Your Understanding

- **Q.** Does packet switching reduce propagation or transmission delay ?

- **Q.** If transmission is one hop only, is there any benefit to breaking a large data file into smaller blocks (called «packets ») ?

Questions

- When a program passes data to a UDP socket, can we say that this data is
  - A PDU ?
  - An SDU ?

- What is the common name for
  - an IP PDU ?
  - An Ethernet PDU ?
  - A TCP PDU ?
  - A UDP PDU ?

Facts to Remember (this document)

- Computer networks are organized using a layered model
- There is one layered model per architecture
  - ex. TCP/IP, Appletalk, Novell Netware, OSI
- The transport layer of TCP/IP provides a programming interface to the application layer. It exists in two forms: UDP (unreliable, datagram) and TCP (reliable, stream) .
- Know the difference between propagation and transmission times.

Solutions

*Test Your Understanding

- **Q** Say *what* each layer of the TCP/IP architecture is doing.
- **A.**
  - Layer 1 = PHY transmits bits on cables or over adio waves in the air or in free space
  - Layer 2 = MAC allows several systems to use the same cable or radio waves
  - Layer 3 = network layer interconnects all systems; has mainly intermediate systems.
  - Layer 4 = transport provides a programming interface to the application
  - Layer 5 = application (also called layer 7 in OSI model) provides applications that allow people and machines to communicate.

back

*Transmission Time

- Transmission time = time to send *x* bits at a given bit rate
- **Q.** time to send 1 MB at 10 kb/s = ?
  A. 8×10^6 bits / 10^4 b/s = 800 s

back

*Test Your Understanding

- **Q.** Does packet switching reduce propagation or transmission delay ?
  A. transmission, in multihop scenario
- **Q.** If transmission is one hop only, is there any benefit to breaking a large data file into smaller blocks (called «packets ») ?
  A.
  - not from a delay view point
  - but because of bit errors on the channel, it is likely that a very large block is not correct, splitting and retransmitting only the incorrect blocks may be better
  - practical buffer sizes may be an issue

back