Introduction to TCP/IP

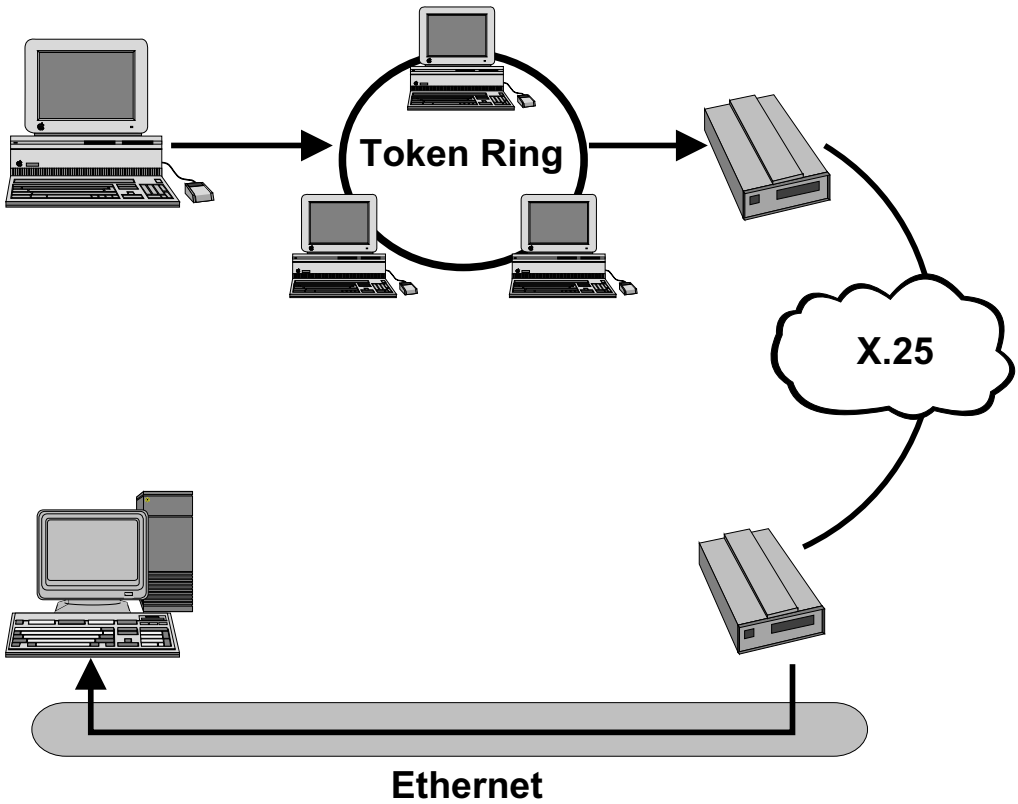
Heiko Holtkamp
May 22 & June 5, 1997

- Introduction/Motivation
- Some Basics...
- What is TCP/IP?
- General Description of TCP/IP
  - TCP/IP protocol architecture
  - Application Layer
  - Transport Layer
  - Internet Layer
  - Network Access Layer
- The Future?! (optional)
  - Problems with IPv4
  - Internet Protocol Next Generation (IPng/IPv6)
- Where to get more Information

Some Basics...

Network: A *Computer Network* is an interconnected collection of autonomous computers.
Two computers are said to be *interconnected* if they are able to exchange information.

internet/Internet: An *internet (internetwork)* is any collection of separate physical networks, interconnected by a common protocol, to form a single logical network.
The *Internet* is the worldwide collection of interconnected networks that uses IP to link various physical networks into a single logical network.



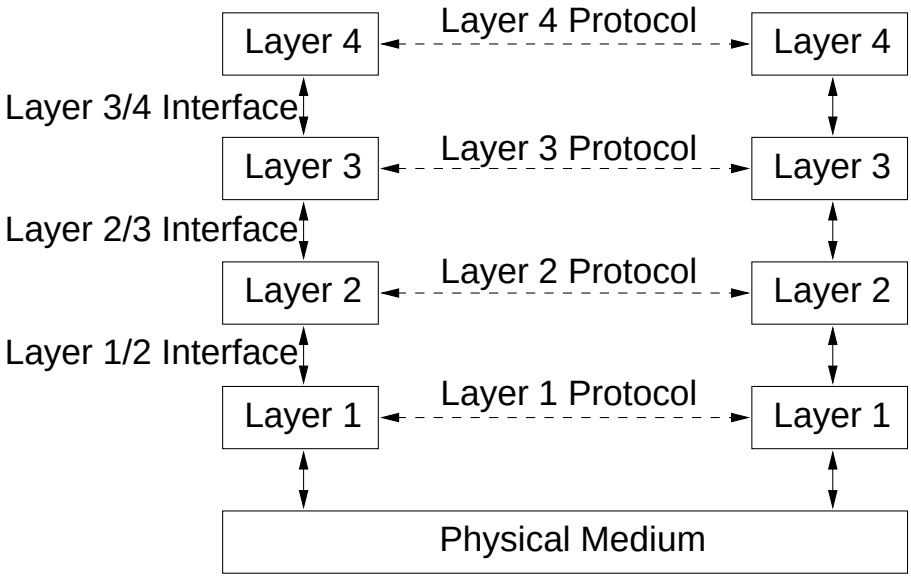
Some Basics...

Protocol: 'Protocols are formal rules of behavior.'

Protocols are rules that coordinate communication between communication partners. The main tasks of protocols are:

- addressing of the communication end points
- control of dataflow
- reliability of (data-transmission) *services*

In data communications we distiguish between two types of protocols: *connection-oriented* and *connectionless* protocols.

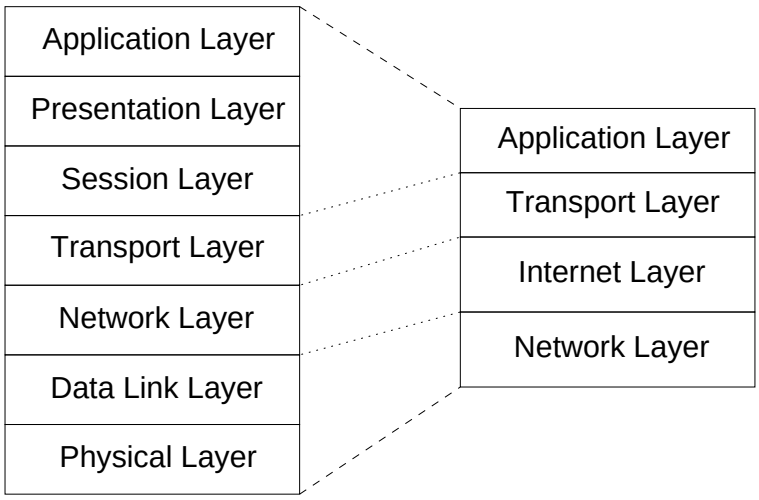


Protocol Stack

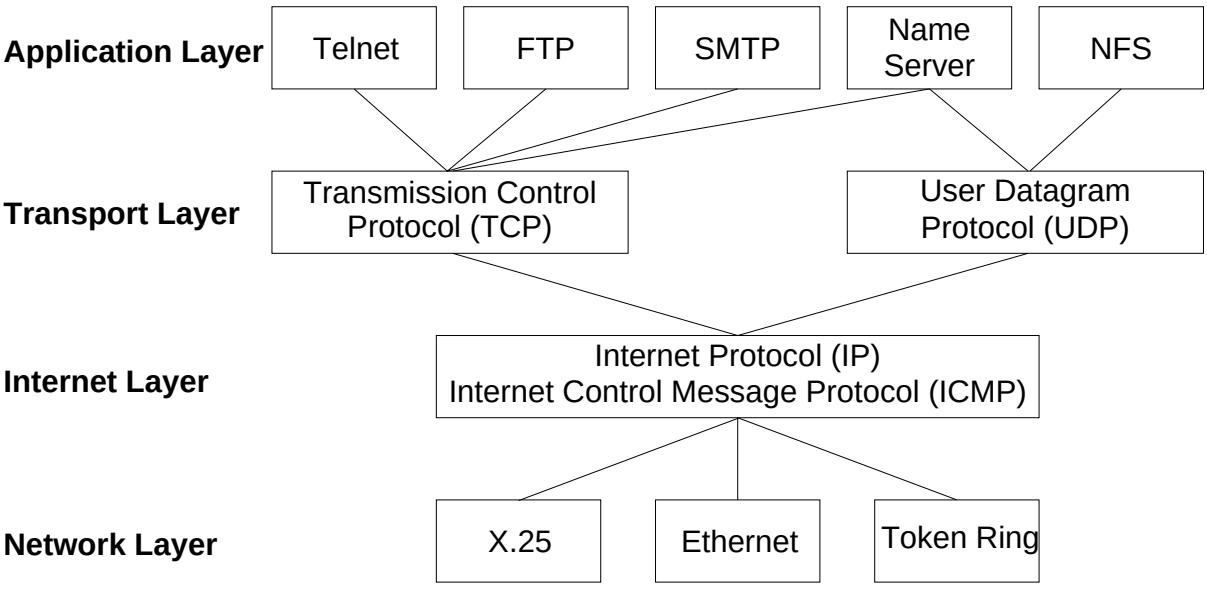
What is TCP/IP?

- TCP/IP is a set of protocols developed to allow cooperating computers to share resources across a network.
- The most accurate name for the set of protocols we are describing is the 'Internet Protocol Suite'.
- TCP (*Transmission Control Protocol*) and IP (*Internet Protocol*) are two of the protocols in this suite.
- Because TCP and IP are the best known of the protocols, it has become common to use the term TCP/IP to refer to the whole family.
- TCP/IP was developed by a community of researchers centered around the ARPAnet.

TCP/IP Protocol Architecture



Comparison of OSI and TCP/IP architecture



TCP/IP architecture model

TCP/IP Protocol Architecture

Data Encapsulation

- Each layer in the stack adds control information to ensure proper delivery.
- This control information is called a *header*.
- Each layer treats all of the information it receives from the layer above as data and places its own header in front of that information.
- The addition of delivery information at every layer is called *encapsulation*.

Application Layer

At the top of the TCP/IP protocol architecture is the *Application Layer*. The application protocols run on top of TCP/IP. All the networking details are done by TCP and IP, so the application programs can treat network connections as if they were simple byte streams.

The most widely known applications are:

Telnet: Network Terminal Protocol; provides remote login over the network.

FTP: File Transfer Protocol; used to for interactive file transfer.

SMTP: Simple Mail Transfer Protocol; protocol for electronic mail.

DNS: Domain Name Service - Name Service; maps IP addresses to the names assigned to network devices.

NFS: Network File System; allows files to be shared by various hosts on the network.

HTTP: Hyper Text Transfer Protocol; the standard transfer protocol in the WWW.

But there are **many** other application protocols...

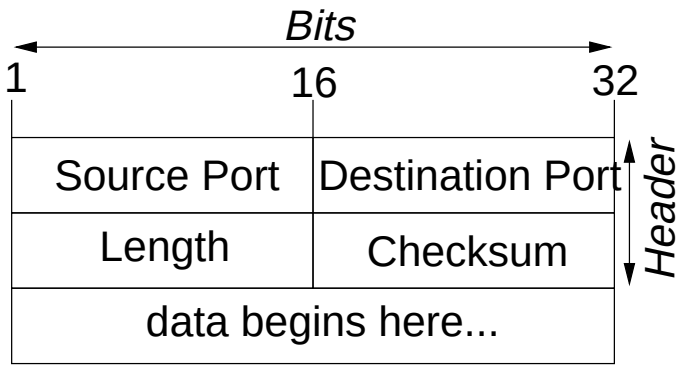
Transport Layer

The two most important protocols in the Transport Layer are the *Transmission Control Protocol (TCP)* and the *User Datagram Protocol (UDP)*.

- TCP is a reliable, connection-oriented, byte-stream protocol.
- UDP is an unreliable, connectionless protocol.

Transport Layer

- The User Datagram Protocol gives application programs direct access to a data delivery service (like IP).
- Unreliable means that there are no techniques in the protocol for verifying that the data reached the other end of the network correctly.
- UDP uses 16-bit numbers for the Source and Destination *Port*, to deliver data to the correct application process.

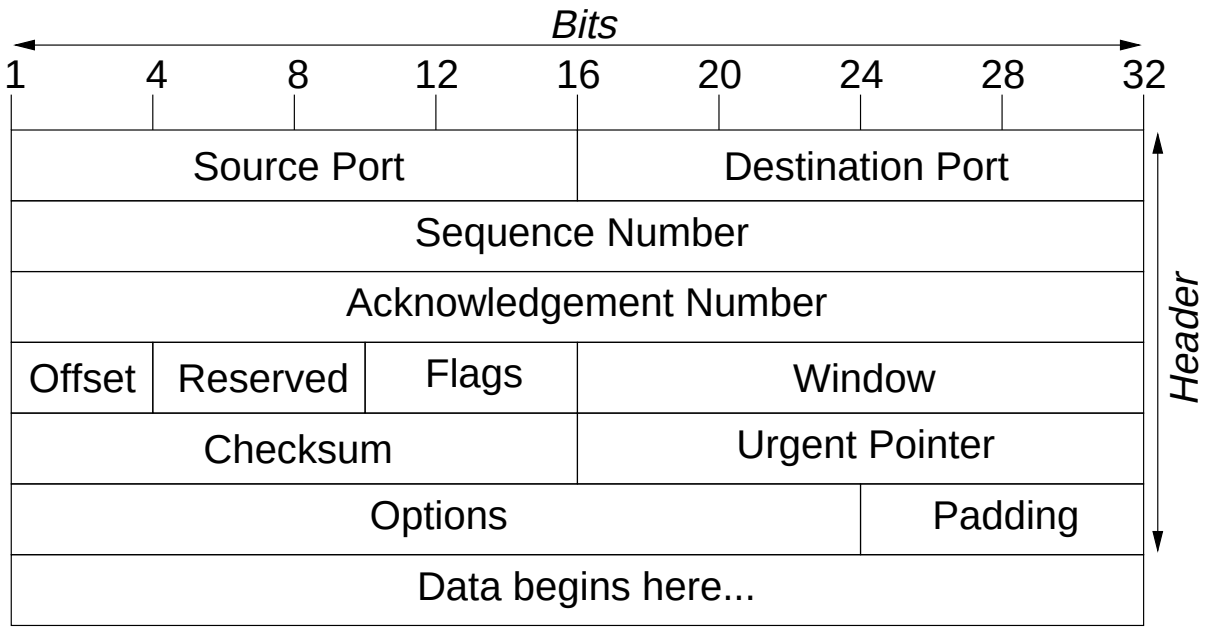


- Why is UDP used?
  - If the amount of data being transmitted is small, the overhead of creating connections and ensuring proper delivery may be greater than the work for retransmitting the data.
  - Sometimes applications use their own techniques for reliable data delivery so they don't require this service from the transport layer.

Transport Layer

What does it mean: *TCP is a reliable, connection-oriented, byte-stream protocol*?

- TCP provides reliability with a mechanism called *Positive Acknowledgement with Re-transmission (PAR)*.
- The data unit exchanged between communicating TCP modules is called a *segment*.



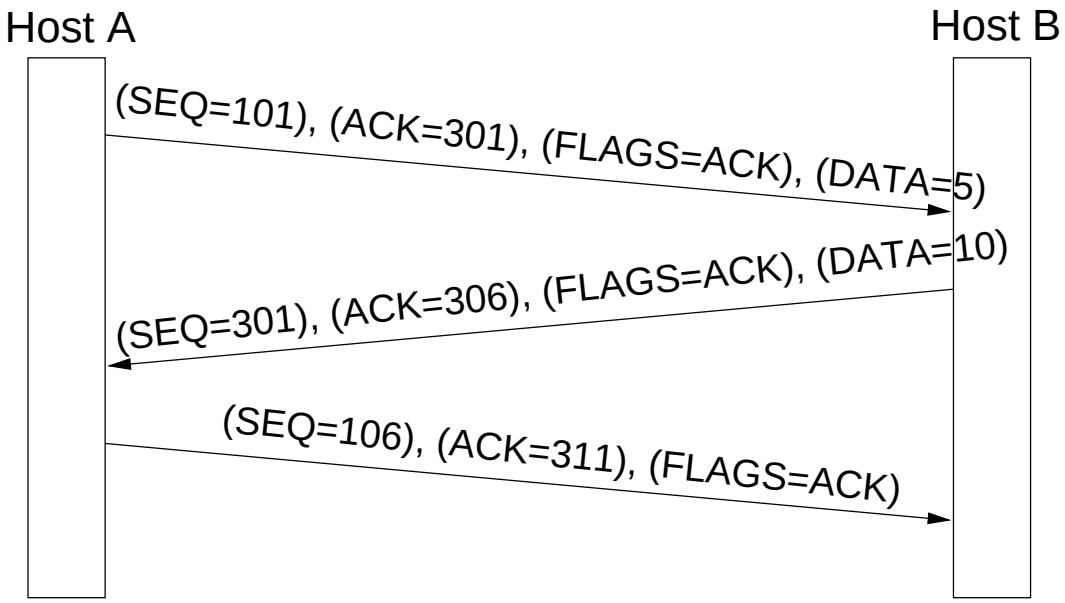
Transport Layer

- TCP is connection-oriented; it establishes a logical end-to-end connection between the two communicating hosts.
  - Control information, called a *handshake*, is exchanged between the two endpoints to establish a connection before data is transmitted.
  - The type of handshake used by TCP is called a *three-way handshake* because three segments are exchanged.

  - After this exchange Host A has the positive evidence that Host B is alive and ready to receive data.
  - When data transfer is finished Host A and Host B exchange a three-way handshake with segments with the FIN bit set to close the connection.

Transport Layer

- TCP views the data it sends as a continuous stream of bytes, not as independent packets.
- Therefore TCP takes care to maintain the sequence in which bytes are sent and received.
- The *Sequence Number* and *Acknowledgement Number* fields in the TCP segment header keep track of the bytes.



- TCP is also responsible for delivering data received from IP to the correct application.
- The application that the data is assumed for is identified by a 16-bit number called the *port number*.

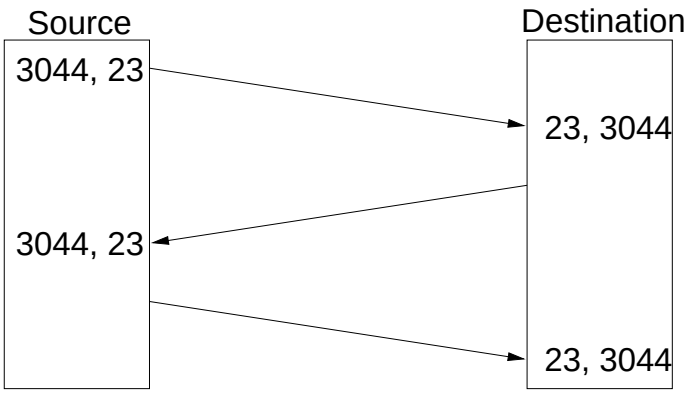
Transport Layer

Dont't forget to say . . . .

- TCP devides the 'application data stream' into pieces of a maximum size of 64 kByte (normally 1.500 Bytes).
- Each of this pieces is sent with its TCP header as an IP datagram.
- The destination TCP reconstructs the pieces back to the original data stream.

Transport Layer

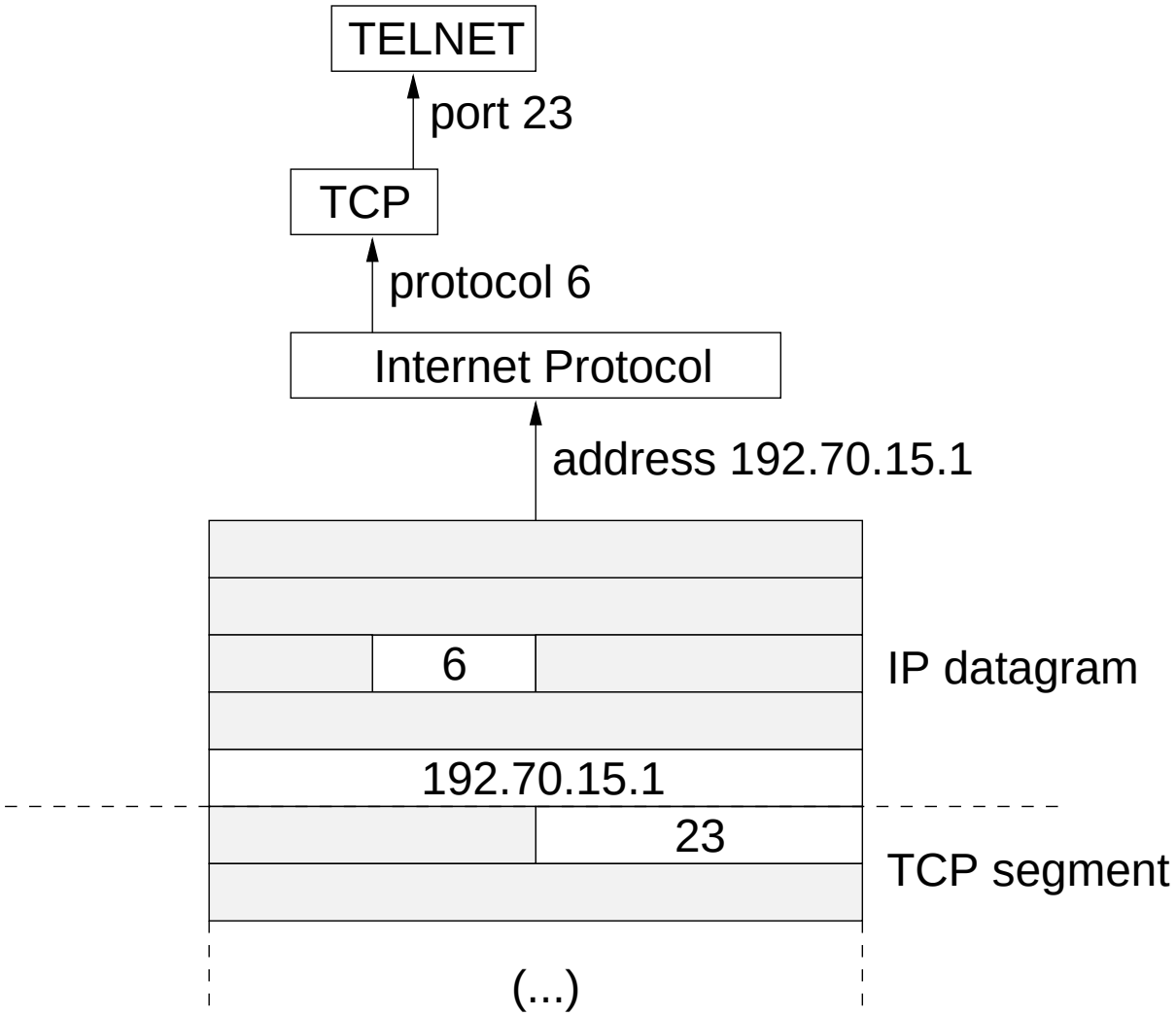
- Port numbers are used for addressing in the transport layer. As noted before, 'applications' are identified by 16-bit port numbers.
- The *source port number* identifies the process that send the data; the *destination port number* identifies the process that is to receive the data.
- On UNIX systems, port numbers are specified in `/etc/services`.
- Port numbers below 256 – also called *well-known ports* – are reserved for 'well-known services' (like FTP, SMTP and TELNET)
- There is a second type of port number for *dynamically-allocated ports*. They are assigned to processes when needed.
- Dynamically-assigned ports provide the flexibilty needed to support multiple users.
- It is the pair of port numbers, source and destination, that uniquely identifies each network connection.



Transport Layer

```
# Network services, Internet style, (Updated from RFC 1340)
#
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
qotd            17/tcp          quote
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp             21/tcp
telnet          23/tcp
smtp            25/tcp          mail
time            37/tcp          timserver
time            37/udp          timserver
nameserver      42/tcp          name            # IEN 116
whois           43/tcp          nicname
domain          53/tcp          nameserver      # name-domain server
domain          53/udp          nameserver
tftp            69/udp
finger          79/tcp
www             80/tcp          http    # WorldWideWeb HTTP
www             80/udp                  # HyperText Transfer Protocol
link            87/tcp          ttylink
hostnames       101/tcp         hostname        # usually from sri-nic
exec            512/tcp
login           513/tcp
who             513/udp         whod
shell           514/tcp         cmd             # no passwords used
syslog          514/udp
printer         515/tcp         spooler         # line printer spooler
talk            517/udp
ntalk           518/udp
route           520/udp         router routed   # RIP
```

Protocols, Ports and Sockets



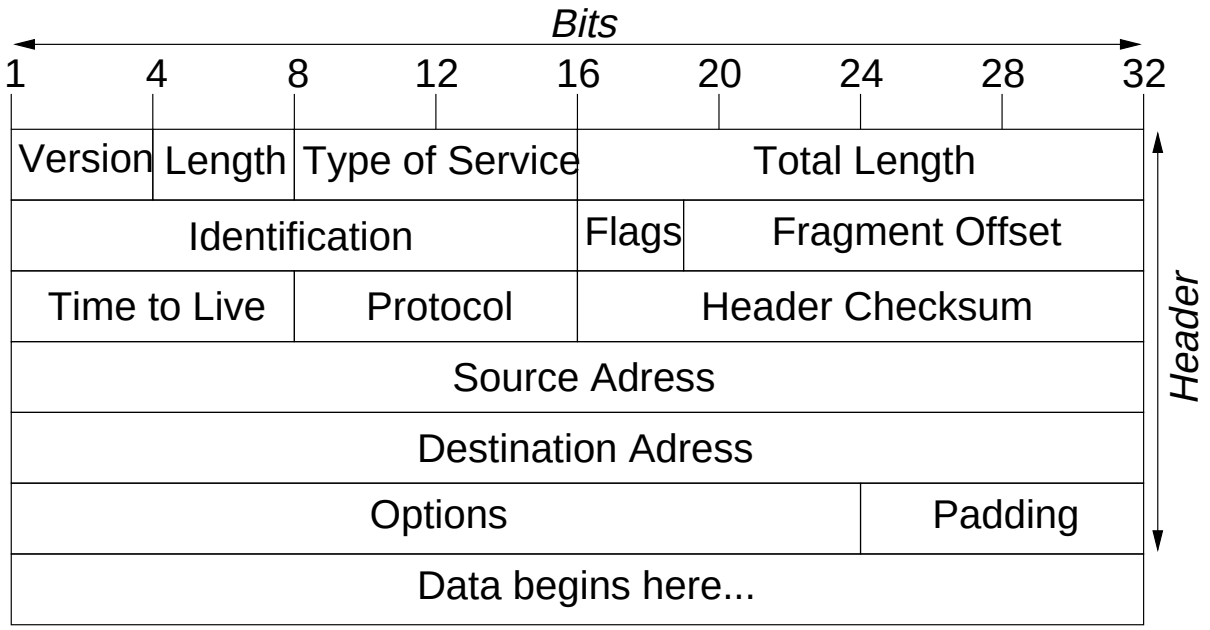
Internet Layer

The *Internet Protocol (IP)* provides the basic packet delivery service on which TCP/IP networks are built.

- Functions of IP
  - defines the *datagram*, which is the basic unit of transmission in the Internet.
  - defines the Internet *addressing scheme*
  - routing datagrams to remote hosts
  - performing *fragmentation* and re-assembly of datagrams
- Characteristics of IP:
  - connectionless protocol
  - unreliable protocol
  - correct
  - packet switching

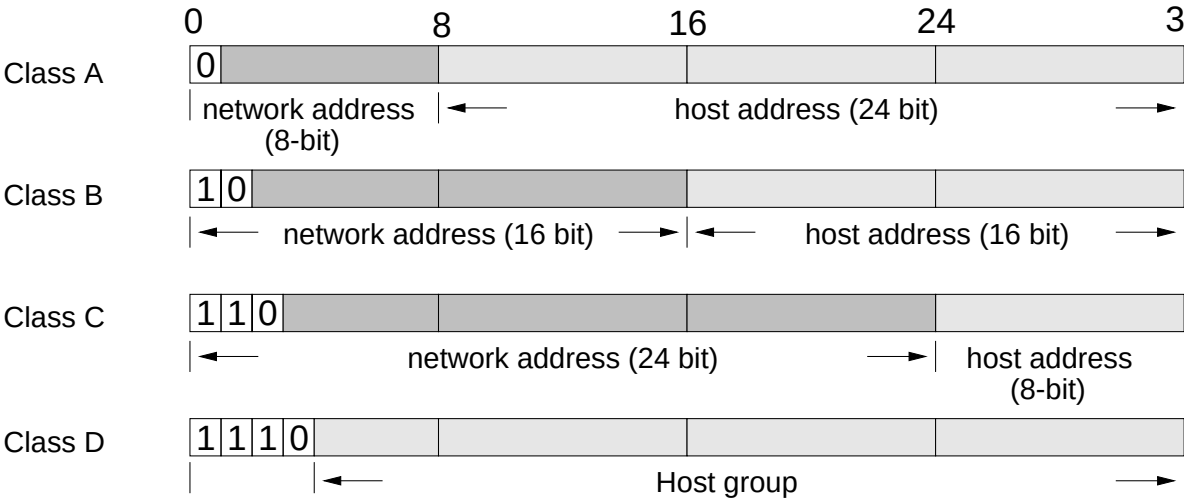
Internet Layer

- A *packet* is a block of data that carries with it the information necessary to deliver it.
- A *packet-switching network* uses the addressing information in the packets to switch packets from one physical network to another, moving them towards their final destination.
- The *datagram* is the packet format defined by the Internet Protocol.
- The first words before data begins are control information called the *header*. The header contains all the information necessary to deliver the packet.



Internet Layer

- IP delivers the datagram by checking the destination address in word 5 of the header.
- The destination address (also the source address) is a standard 32-bit *IP address* that defines a specific network and a specific host on that network. IP addresses are usually written as four decimal numbers (bytes) separated by dots (e.g. 128.34.35.12).



- If the destination address is not on the local network, the datagram is passed to a *gateway (router)* for delivery.
- Deciding which gateway to use is called *routing*. IP makes routing decisions for each individual datagram.

Internet Layer

- Every host and every *gateway (router)* has its own unique IP address.
- An IP address contains a *network part* and a *host part*.
- A host which is connected to more than one network, has a different IP address for every network (e.g. a gateway).
- As mentioned before IP addresses are divided into classes:

  class A: a first byte value less than 128; the first byte identifies the network number, the next three bytes the host address.

  class B: a first byte value between 128 and 191; the first two bytes identify the network number, the next two bytes the host number.

  class C: a first byte value between 192 and 223; the first three bytes identify the network, the last byte is the host number.

  class D: a first byte value greater than 223; 'reserved address'
- The network numbers are assigned by the *Network Information Center (NIC)* to avoid conflicts.
- Often IP addresses are called *host addresses* - but this is misleading: IP addresses are assigned to network interfaces, not to computers.

Internet Layer

- The standard structure of an IP address can be locally modified by using host-address bits as additional network-address bits.
- These 'new' network bits define a network within the larger network, called a *subnet*.
- The subnet is only known locally. To the rest of the Internet, the address is still interpreted as a standard IP address.
- A subnet is defined by applying a bit mask, the *subnet mask*, to the IP address:
  - If a bit is on (1) in the mask, that equivalent bit in the address is interpreted as a network bit.
  - If a bit is off (0) in the mask, the bit belongs to the host part of the address.

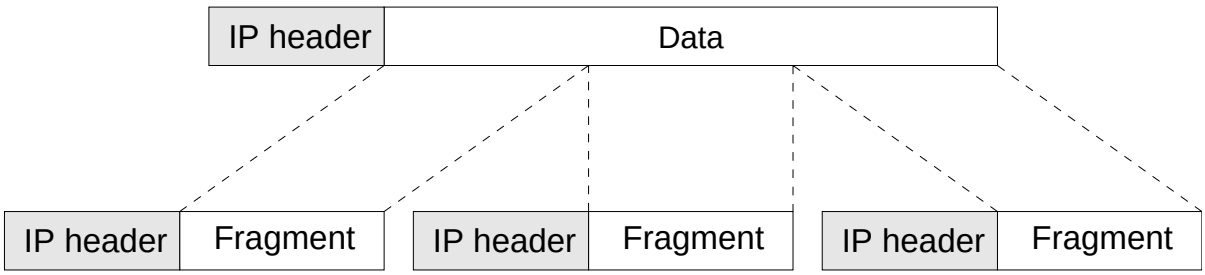
10	Network	Subnet	Host

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0

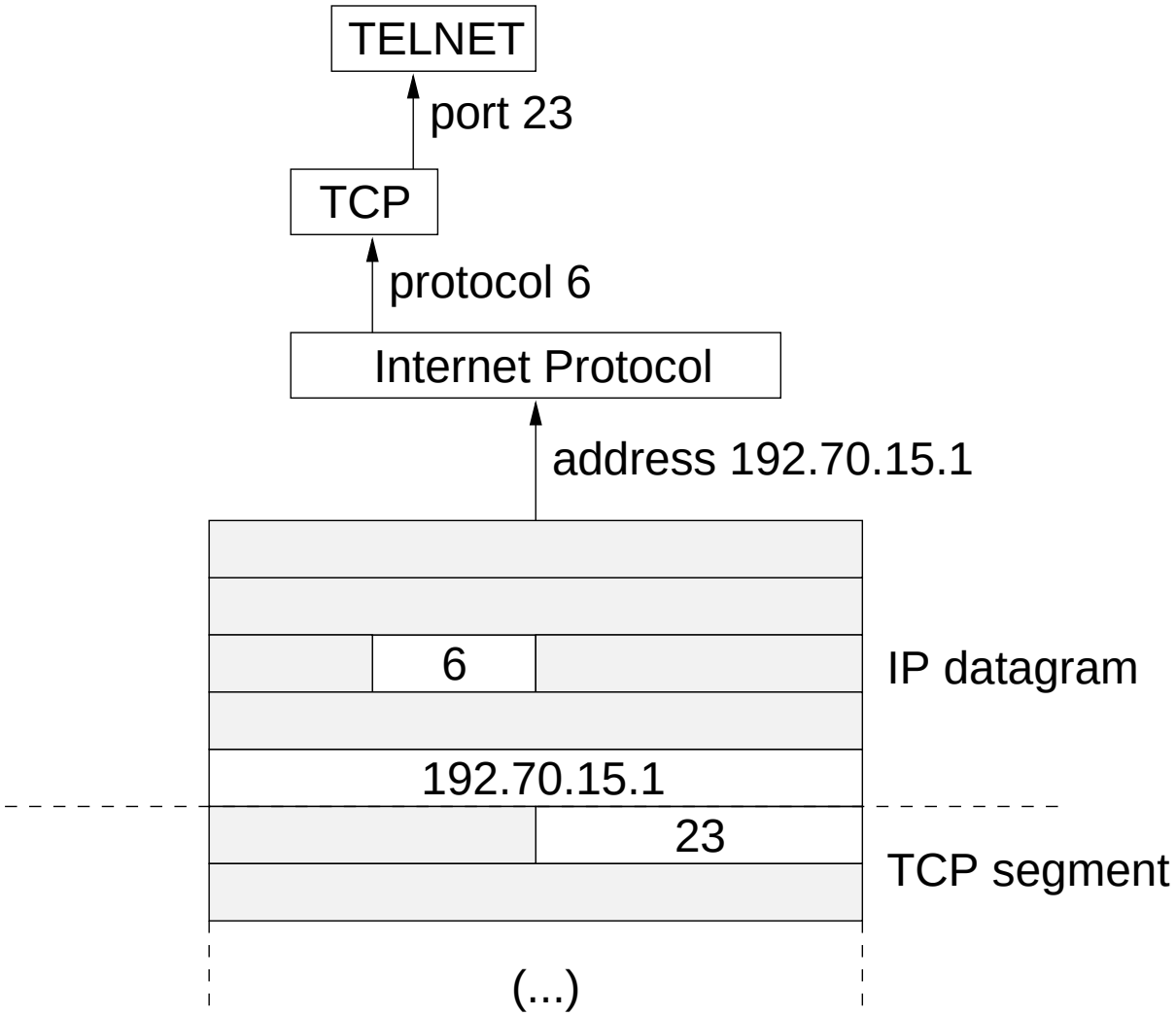
In this mask, the 16-bit host-number is divided into a 6-bit subnet-number and a 10-bit host-number (62 LANs, 1.022 hosts).

Internet Layer

- As a datagram is routed through different networks, it may be necessary that a gateway divides the datagram into smaller pieces.
- This process is called *fragmentation*.
- The format of each fragment is the same as the format of any 'normal' fragment.
- Word 2 in the header contains information that identifies each fragment and provides information how to re-assemble the fragments back into the original datagram.

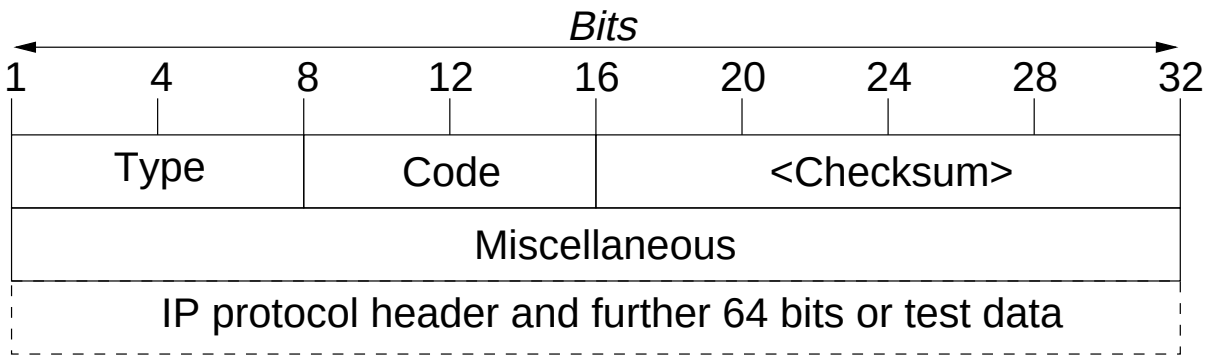


Protocols, Ports and Sockets



Internet Layer

- The *Internet Control Message Protocol (ICMP)* is an integral part of IP and must be implemented by every IP module.
- The only task of ICMP is to transport error and diagnostic data for IP.
- Because ICMP has to transport various information, only the basic structure of the ICMP protocol header is fixed; the meanings of the individual fields are variable.



- ICMP sends messages that perform the following control, error reporting and informational functions:
  - Flow control
  - Detection of unreachable destinations
  - Redirecting routes
  - Checking remote hosts

Network Layer

The *Network Layer* is the lowest layer of the TCP/IP protocol hierarchy.

- Protocols in this layer define how to use a specific network to transmit an IP datagram.
- Protocols in the Network Layer must know the details of the underlying network (packet structure, packet length, addressing, . . .) to correctly format the data being transmitted over the network.
- For new hardware technologies, new Network Layer protocols must be developed.
- Because of this there are many protocols, one for each physical network standard.
- Functions performed at this level include encapsulation of IP datagrams into *frames* for transmission over the network and mapping of IP addresses to physical network addresse.
- Example RFCs that define protocols in the Network Layer are:
  - RFC826, *Address Resolution Protocol (ARP)*
  - RFC894, *A Transmission of IP Datagrams over Ethernet Networks*
  - RFC903, *Reverse Address Resolution Protocol (RARP)*
  - RFC877, *Transmission of IP datagrams over X.25*
  - . . .

Where to get more information

- take a look at my homepage (http://www.rvs.uni-bielefeld.de/~heiko)
- 'Requests for Comments – RFCs' (http://www.internic.net/rfc)
- Charles L. Hedrick: Introduction to the internet protocols (http://www.doc.ic.ac.uk/~ih/pc_info/tcpip/intro/intro0.html)
- Hunt C.: TCP/IP Network Administration. O'Reilly & Assoc., 1995
- Santifaller M.: TCP/IP and ONC/NFS - Internetworking in a UNIX Environment. Addison Wesley, Wokingham, England, 1994, 2nd ed.
- Comer D.E., Stevens D.L.: Internetworking with TCP/IP: Volume I-III. Prentice Hall, Englewood Cliffs, New Jersey, 1994
- Tanenbaum A.S.: Computer Networks. Prentice Hall, Upper Saddle River, New Jersey, 1996, 3rd ed.

  (see also `http://www.prenhall.com/divisions/ptr/tanenbaum/book.html`)
- . . .