2 Internet History and Technology
2.0 Definition

What is the Internet?

Definition by the Federal Networking Council (FNC, 24.10.95):

"The Federal Networking Council (FNC) agrees that the following language reflects our definition of the term *"Internet":*

"**Internet**" refers to the ***global information system*** that -

- is logically linked together by a ***globally unique address space*** based on the Internet Protocol (IP) or its subsequent extensions/follow-ons;
- is able to support communications using the ***Transmission Control Protocol / Internet Protocol (TCP/IP)*** suite or its subsequent extensions/follow-ons, and/or other IP-compatible protocols; and
- provides, uses or makes accessible, either publicly or privately, ***high level services*** layered on the communications and related infrastructure described herein."

2.1 History

How did the internet evolve? *(based on various articles and web-resources)*

- **1957** start of Sputnik
  ⇒ USA establish ***Advanced Research Projects Agency*** (ARPA)
- **1962** report of Paul Baran (RAND corporation, for the US Air Force) on ways of maintaining control over missiles and bombers in case of a nuclear attack.
  ⇒ design of a military communication network, allowing a second strike even after the complete destruction of several nodes (cities).
  central idea: **packet switching**, i.e.
  – splitting messages into packets (datagrams)
  – sending and forwarding packets in the network, until the destination address has been reached.
- **1968** ARPA decides to have ARPANET built.
- **1969** 4 universities are connected by the first ARPANET:
  – Univ. of California at Los Angeles,
  – SRI (in Stanford),
  – Univ. of California at Santa Barbara
  – University of Utah.

  Backbones: 50Kbps Hosts: 4

Internet history (2):

- **1972**:
  – first e-mail program *(author: R. Tomlinson, BBN)*
  – ARPA renamed *The Defense Advanced Research Projects Agency* (DARPA)
  – ARPANET uses the *Network Control Protocol* or *NCP* for file transfer (communication between hosts in the same network)

  Backbones: 50Kbps Hosts: 23
- **1973:**
  – Vinton Cerf (Stanford), Bob Kahn (DARPA) begin to design the **network protocol TCP/IP**
    - goal: communication between different computer networks
- **1974:**
  – first use of the term **Internet** by Cerf /Kahn in their paper on the Transmission Control Protocol.
- **1976:**
  – design of the **ethernet** for data transfer over coaxial lines *(Robert M. Metcalfe)*
    - essential for the design of *Local Area Networks* (LAN).
  – construction of SATNET (packet satellite network) to improve the links between USA and Europe (using the commercial satellites Intelsat of the *International Telecommunications Satellite Organization*)

Internet history (3):

- **1976** (cont.)
  – design of UUCP (Unix-to-Unix CoPy) at AT&T Bell Labs, one year later standard part of UNIX
  – Department of Defense tests TCP/IP and decides to use it in ARPANET.

  Backbones: 50Kbps ARPANET + satellite and wireless interconnections

  Hosts: 111+
- **1979**:
  – USENET (decentralised news group network) developed by Steve Bellovin (student at the University of North Carolina), Tom Truscott and Jim Ellis using UUCP.
  – BITNET ("Because it's Time Network") founded by IBM, uses "store and forward" routing; used for email and for list servers (e.g. THEORYNET).
- **1981**:
  – National Science Foundation establishes CSNET as another backbone at 56 Kbps for institutions without access to the ARPANET
  – Vinton Cerf develops plan for inter-network links between CSNET and ARPANET

  Backbones: 50Kbps ARPANET, 56Kbps CSNET + satellite and wireless interconnections

  Hosts: 213

Internet history (4):

- **1983**:
  - creation of the Internet Activities Board (IAB)
  - TCP/IP becomes the standard protocol of ARPANET.
  - start of **European Academic and Research Network (EARN)** (funded by IBM)
  - Domain Name System (DNS) designed at University of Wisconsin. Allows the use of names which are transformed into IP-numbers by the server

  ⇒ much simpler access to the network

  Backbones: 50Kbps ARPANET, 56Kbps CSNET
  + satellite and wireless interconnections

  Hosts: 562

- **1984**:
  - ARPANET divided into MILNET (military) and ARPANET (research). DoD supports both networks.
  - CSNET signs contract with MCI for providing new backbone at 1.5 Mbps (25 times faster than before!), IBM provides new router, Merit administrates the network, called NSFNET from now on. The old network is still called CSNET.

  Backbones: 50Kbps ARPANET, 56Kbps CSNET
  + satellite and wireless interconnections

  Hosts: 1024

Internet history (5):

- **1985:**
  - NSF uses the first new backbone lines (T1 lines)

  Backbones: 50Kbps ARPANET, 56Kbps CSNET, 1.544Mbps (T1) NSFNET
  + satellite and wireless interconnections

  Hosts: 1.961

- **1986**:
  - *Internet Engineering Task Force (IETF)* founded for coordinating the work on ARPANET, on the US Defense Data Network (DDN), and on the Internet kernel.

  Backbones: 50Kbps ARPANET, 56Kbps CSNET, 1.544Mbps (T1) NSFNET
  + satellite and wireless interconnections

  Hosts: 2.308

- **1987**:
  - BITNET und CSNET form the Corporation for Research and Educational Networking (CREN) within the National Science Foundation (NSF)

  Backbones: 50Kbps ARPANET, 56Kbps CSNET, 1.544Mbps (T1) NSFNET
  + satellite and wireless interconnections

  Hosts: 28.174

Internet history (6):

- **1988:**
  - T1 NSFNET backbone completely installed, but already too slow due to increased network traffic.
  - Merit & Partners found research institute Advanced Network Systems (ANS)
    ⇒ concept for T3, a 45 Mbps link to be installed in NSFNET before the end of 1991.

  Backbones: 50Kbps ARPANET, 56Kbps CSNET, 1.544Mbps (T1) NSFNET
  + satellite and wireless interconnections

  Hosts: 56,000

- **1990:**
  - ARPANET replaced completely by NSFNET
  - Tim Berners-Lee (CERN, Geneva) implements a **hypertext system** to improve exchange of information among researchers in high energy physics

  Backbones: 56Kbps CSNET, 1.544Mbps (T1) NSFNET
  + satellite and wireless interconnections

  Hosts: 313,000

- **1991:**
  - end of CSNET, start of Gopher.

  Backbones: partially 45Mbps (T3) NSFNET, some private networks
  + satellite and wireless interconnections

  Hosts: 617,000

Internet history (7):

- **1992**:
  - foundation of the Internet Society
  - **World-Wide Web** released for public use by CERN.
  - NSFNET has complete T3 backbone (45Mbps)
  - first MBone broadcast
  - The term "Surfing the Internet" is coined by Jean Armour Polly

  Backbones: 45Mbps (T3) NSFNET, many private networks
  + satellite and wireless interconnections

  Hosts: 1,136,000

- **1993**:
  - NSF founds InterNIC for Internet services:
    - data base services (by AT&T)
    - registration services (by Network Solutions Inc.),
    - information services (by General Atomics/CERFnet).
  - Mosaic-Browser for the WWW developed by Marc Andreessen (NCSA) and the University of Illinois

  Backbones: 45Mbps (T3) NSFNET, many private networks
  + satellite and wireless interconnections

  Hosts: 2,056,000

Internet history (8):

- **1994**:
  - strong growth of the INTERNET because of the WWW
  - **Netscape provides new browser**
  - first commercial applications, e.g.:
    - ordering pizzas via WWW home page of Pizza Hut
    - opening of First Virtual, the first "cyberbank"
    - Shopping malls arrive on the Internet
  - ATM (Asynchronous Transfer Mode, 145Mbps) Backbone installed in NSFNET.

  Backbones: 145Mbps (ATM) NSFNET, many private networks + satellite and wireless interconnections

  Hosts: 3,864,000

- **1995**:
  - NSF ends free access to the NSF-Backbone, from Mai 1995 only via private Internet service providers
  - Compuserve, America Online, Prodigy begin to provide Internet access

  Backbones: 145Mbps (ATM) NSFNET (private), many interconnected private nets running at 56Kbps,1.544Mbps, 45Mpbs, and155Mpbs + satellite and wireless interconnections

  Hosts: 6,642,000

Internet history (9):

- **1996**:
  - MCI upgrades Internet backbone speed from 155Mbps to (effectively) 622Mbps
  - Restrictions on Internet use around the world:
    - China: requires users and ISPs to register with the police
    - Germany: cuts off access to some newsgroups carried on Compuserve
    - Saudi Arabia: confines Internet access to universities and hospitals
    - Singapore: requires political and religious content providers to register with the state
    - New Zealand: classifies computer disks as "publications" that can be censored and seized
- **1998**:
  - Internet users get to be judges in a performance by 12 world champion ice skaters on 27 March, marking the first time a television sport show's outcome is determined by its viewers
  - US Postal Service allowing electronic postal stamps to be purchased and downloaded for printing from the Web.

  Backbones: many backbones from independent ISPs (e.g. MCI, AT&T, Sprint, UUnet, BBN planet, ANS etc.), plans for Gigabit-backbones

  Hosts: ~ 40,000,000

Internet history (10):

- **1999**:
  - First Internet Bank of Indiana, the first full-service bank available only on the Net
  - MCI/Worldcom begins upgrading the US backbone to 2.5Gbps
  - business.com is sold for US$7.5million (it was purchased in 1997 for US$150,000)
  - Technologies of the Year: E-Trade, Online Banking, MP3
  - Emerging Technologies: Net-Cell Phones, Thin Computing, Embedded Computing
- Viruses of the Year: Melissa (March), ExploreZip (June)

- **2000**
  - The US timekeeper (USNO) and a few other time services around the world report the new year as 19100 on 1 Jan
  - massive denial of service attacks against Yahoo, Amazon, and eBay
  - Web size estimates by NEC-RI and Inktomi surpass 1 billion indexable pages
  - The European Commission contracts with a consortium of 30 national research networks for the development of **Géant**, Europe's new gigabit research network
  - Technologies of the Year: ASP, NAPSTER?, Wireless devices?, IPV6?
  - Viruses of the Year: Love Letter (May)

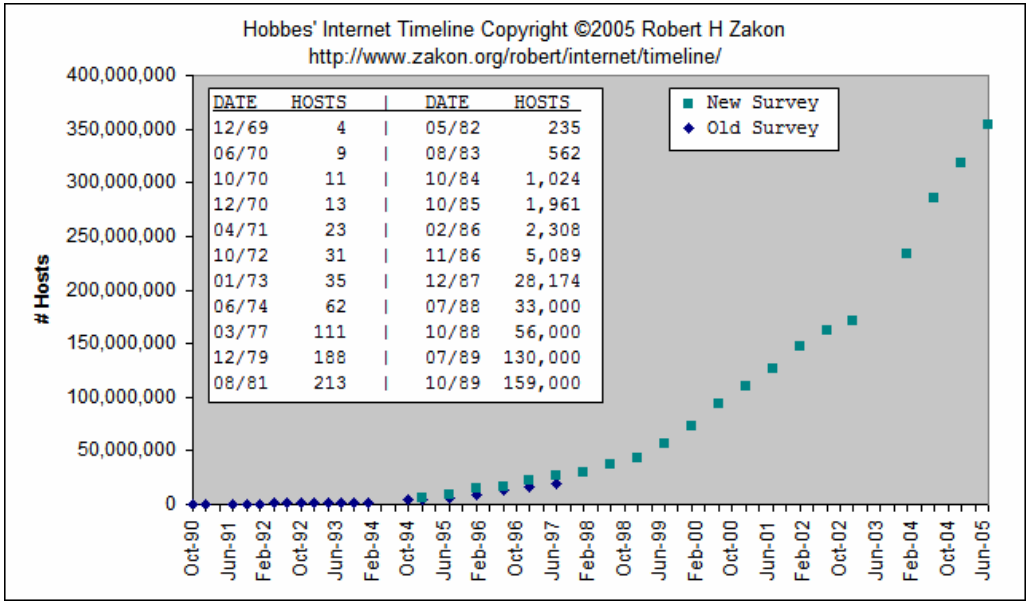
Internet history (11):

- **2001**
  - Forwarding email in Australia becomes illegal with the passing of the Digital Agenda Act, as it is seen as a technical infringement of personal copyright (4 Mar)
  - Napster keeps finding itself embroiled in litigation and is eventually forced to suspend service; though it expects to come back as a subscription service
  - SETI@Home launches on 17 May and within four weeks its distributed Internet clients provide more computing power than the most powerful supercomputer of its time (:par:)
  - European Council finalizes an international cybercrime treaty on 22 June and adopts it on 9 November. This is the first treaty addressing criminal offenses committed over the Internet.
  - .biz and .info are added to the root server on 27 June with registrations beginning in July
  - Afghanistan's Taliban bans Internet access country-wide, including from Government offices, in an attempt to control content (13 Jul)
  - Code Red worm and Sircam virus infiltrate thousands of web servers and email accounts, respectively, causing a spike in Internet bandwidth usage and security breaches (July)
  - Géant goes operational in November
- **2002**
  - A distributed denial of service (DDoS) attack struck the 13 DNS root servers knocking out all but 5 (21-23 Oct). Amidst national security concerns, VeriSign hastens a planned relocation of one of its two DNS root servers

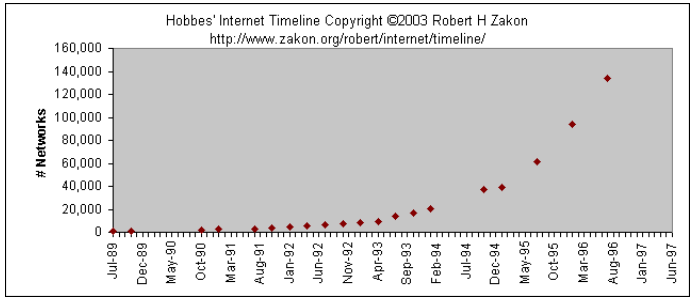
Internet history (12):

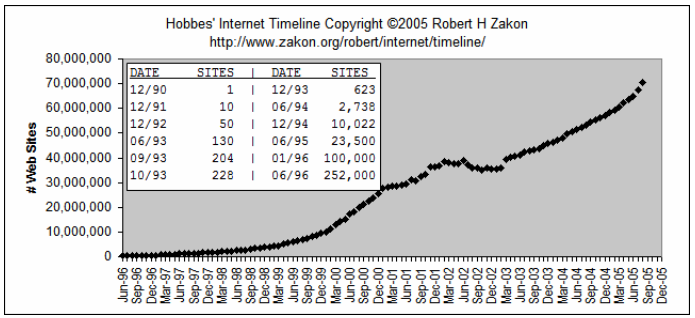
- **2003**
  - The SQL Slammer worm causes one of the largest and fastest spreading DDoS attacks ever. Taking roughly 10 minutes to spread worldwide, the worm took down 5 of the 13 DNS root servers along with tens of thousands of other servers, and impacted a multitude of systems ranging from (bank) ATM systems to air traffic control to emergency (911) systems (25 Jan). This is followed in August by the Sobig.F virus (19 Aug), the fastest spreading virus ever, and the Blaster (MSBlast) worm (11 Aug), another one of the most destructive worms ever
  - The French Ministry of Culture bans the use of the word "e-mail" by government ministries, and adopts the use of the more French sounding "courriel" (Jul)
  - Little GLORIAD (Global Ring Network for Advanced Application Development) starts operations (22 Dec), consisting of a networked ring across the northern hemisphere with connections in Chicago, Amsterdam, Moscow, Novosibirsk, Zabajkal'sk, Manzhouli, Beijing, and Hong Kong. This is the first-ever fiber network connections across the Russia-China border
- **2004**
  - For the first time, there are more instances of DNS root servers outside the US
  - Network Solutions begins offering 100 year domain registration (24 Mar)
  - CERNET2, the first backbone IPv6 network in China, is launched by the China Education and Research Network (CERN) connecting 25 universities in 20 cities at speeds of 1-10Gbps (27 Dec)

Internet growth: # hosts



Internet growth: # networks, # web servers





Hosts = a computer system with registered ip address

Networks = registered class A/B/C addresses

Domains = registered domain name (with name server record)

Sites = # of web servers (one host may have multiple sites by using different domains or port#)

Reasons for the tremendous growth of the internet:

- **growth of bandwidth in wide area networks**
  (within few years from 1,5 Mb/s to 155 Mb/s to 10 Gb/s to …,
  important for network backbones)
- **improved transmission performance over telephone lines**
  (important for end user access to the internet)
  - **ISDN = *Integrated Services Digital Network***
    - 64 kilobits per second (in Germany), all-digital end-to-end channel
    - a set of communications standards allowing a single wire or optical fibre to carry voice, digital network services and video
  - **ADSL = *Asymmetric Digital Subscriber Line*** *(offered as **T-DSL**)*
    - bandwidth available for downstream connection is significantly larger than for upstream (1.5 - 20 Mb/s versus 16 - 640 Kb/ s)
    - well suited for web browsing and client-server applications as well as for some emerging applications such as video on demand
    - can carry digital data, analog voice and broadcast MPEG2 video in a variety of implementations to meet customer needs
- **standardised, platform independent interfaces to the WorldWideWeb**
  (WWW-Browser: Mosaic, Netscape, Firefox, Internet-Explorer; programming language Java)

Remarks on this “internet history”

- Mainly the American perspective.
- Different authors use slightly different numbers and facts.
- The Internet and WorldWideWeb evolved in universities and research institutes, it took long before the industry recognised the advantages and the economic potential of the internet.
- Meanwhile Gigabit networks are operational and there are experiments with Terabit networks; this requires very powerful network nodes being capable of supporting Gigabit (and even Terabit) traffic.

Remarks on the German Internet history:

- “Deutsches Forschungsnetz” (DFN-Verein) since mid 80s, based on X.25
- “Wissenschaftsnetz” (WIN) installed end of 1996, lines at 35 Mbps, before that mainly 2 Mb lines
- “Gigabit-Wissenschaftsnetz” (G-Win) installed in 2002/2003, lines at 10 Gb and 2.5 Gb/s

⇒ Germany has been far behind the USA with respect to information infrastructure, but now it has one of the most powerful network worldwide.

- While the USA had an “information superhighway research program”, one of the main concerns of German politicians in the 90s has been the threat of sex and crime on the Internet.
- In 1984, the first Email has been received in Germany at the University of Karlsruhe
- From 1994-1998, the University of Karlsruhe administrates all German domains
- Ca. 40% of all German websites are hosted at Karlsruhe
- Since Fall 2000, wireless networks (waveLAN) have been installed at German universities (“notebook university”-projects 2002/2003)

⇒ German universities meanwhile have perfect network infrastructure for using mobile devices for teaching and learning, even top universities in North America do not provide better service.

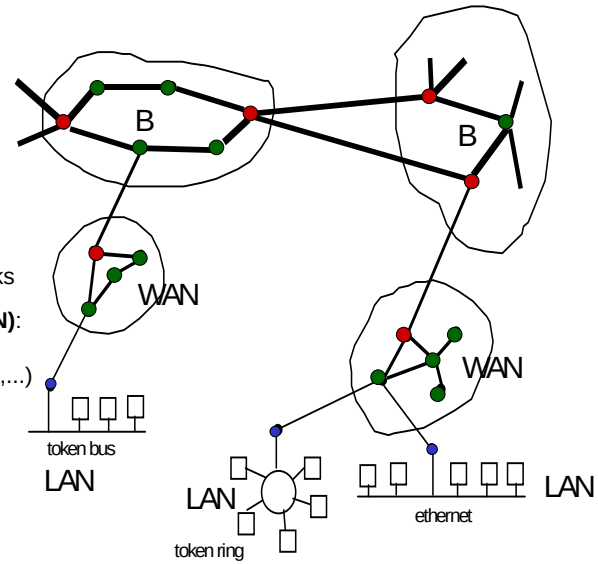
Internet services:

- Telnet: virtual terminal for remote log in
- E-mail: electronic mail service for simple and efficient sending of messages (+ attached documents)
- FTP: File Transfer Protocol, allows for simple transport of files over the network
- Newsgroup (Usenet): international bulletin board
- WWW: distributed collection of multimedia documents connected by hyperlinks, web browsers allow to "surf" in the net, i.e. to follow hyperlinks; essential characteristics:
  - client-server architecture
  - HTML (Hypertext Markup Language): document description language
  - HTTP (Hypertext Transfer Protocol): protocol for exchange of data (linked documents) between client and server
  - URL (Uniform Resource Locator): unique addressing of documents
- Intranet: company-wide network based on Internet technology

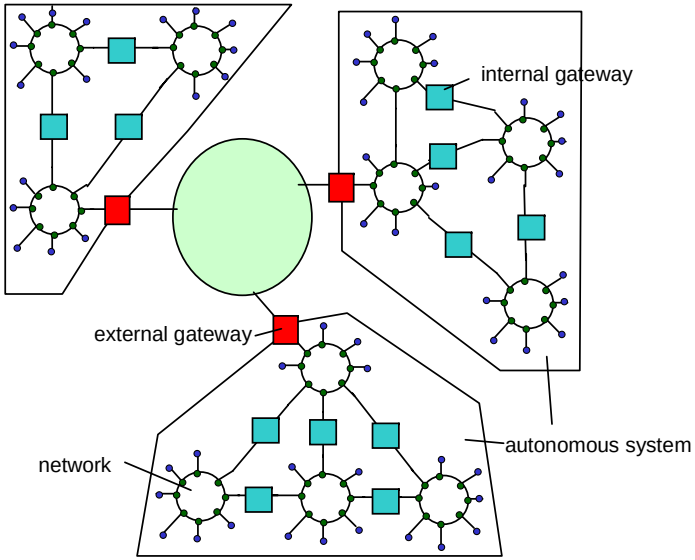
2.2 Internet Technology

structure of the Internet:

- **Backbone**: high bandwidth “internet kernel” several backbones connected by high capacity and high speed cables
- **Wide Area Network (WAN)**: national (D, F, GB,...) and regional (Bayern, Baden Württ., ...) networks
- **Metropolitan Area Network (MAN)**: (small)regional networks (Ruhr region, Berlin, NewYork City,...)
- **Local Area Network (LAN)**: Institut AIFB, Badenwerk,...
- interconnected via
  - network access points,
  - routers, and
  - gateways



different view of the Internet :

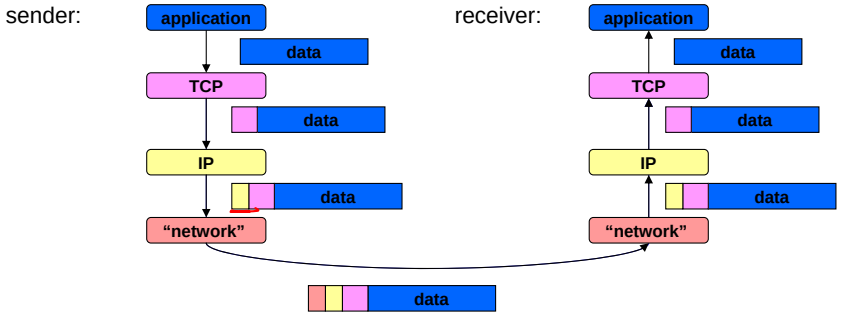


Communication in the Internet :

- standard architecture for communication between computers:
  OSI network layer model of the ISO, in the Internet reduced to

– application layer	TELNET, FTP, SMTP, HTTP, spec. application,...
– transport layer	TCP, UDP
– network layer	IP, ICMP
– physical &data link layer	NSFNET, SATNET, WIN, LAN, radio,...

How does it work?

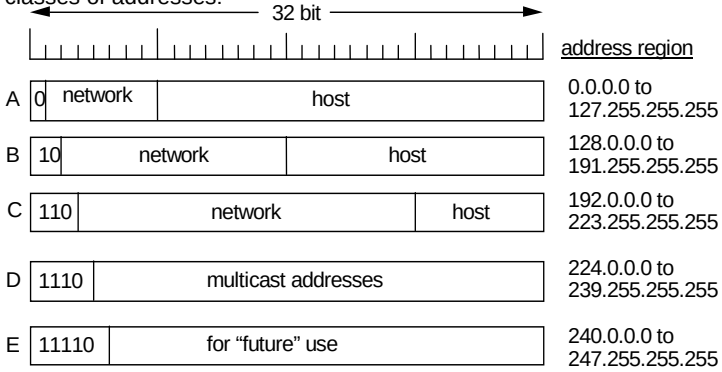


IP - the Internet Protocol
(network layer = Vermittlungsschicht)

- Defines an **addressing format** in the Internet
- assures that **routers** know what to do with datagrams
- provides an **unreliable** and **connectionless** service for the transport layer
- the next layer above (transport layer) is responsible for reliability

addressing format:

- IP-address has 32 bit (4 decimal blocks of numbers) e.g.: 129.13.124.60
- addresses for host computers (or devices) and networks
- different classes of addresses:



IP-addresses (original version):

There are up to

- 126 networks with up to 16 million hosts each (class A)
- 16.382 networks with up to 64 k hosts each (class B)
- 2 million networks (e.g. LANs) with up to 254 hosts each (class C)

and **multicast** (class D) to send data to several host addresses simultaneously.

- each IP address is transformed into a physical address (e.g. an Ethernet address) by the Address Resolution Protocol (**ARP**)
- number of addresses (in particular class B) not sufficient
- meanwhile, class E addresses are also used (as "overflow address space")

⇒ new Classless Inter Domain Routing Scheme (CIDR)

new IP addressing system in IPv6 *(→ later section)*

The Domain Name System (DNS)

IP addresses hard to handle (memorise) for humans

⇒ textual addressing system DNS

DNS built in the reverse order of IP addresses: (host.network.area)

name server transforms DNS addresses into IP-addresses

example:

– aifbquadriga.aifb.uni-karlsruhe.de corresponds to 129.13.124.56

standard names for domains

– .edu = educational institutions
– .com = commercial companies
– .gov= government authorities
– .org = organisations
– ….

and names for countries

– .de = Germany
– .uk = Great Britain (United Kingdom)
– .fr = France
– .au = Australia
– .at = Austria
– ….

new names: .aero, .biz, .coop, .info, .museum, .name, .pro

Nowadays, private domain names are easily available from many service providers and, officially but more expensive, from internic

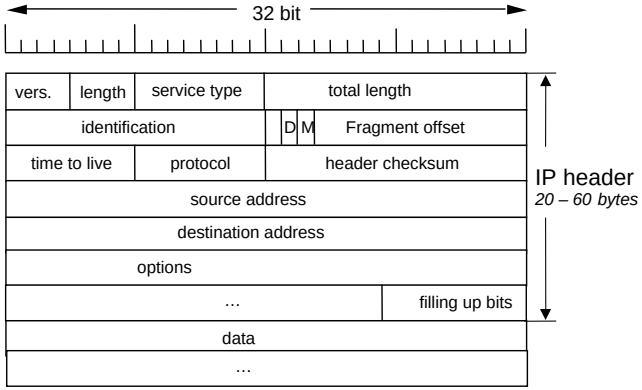
IP protocol - more details:

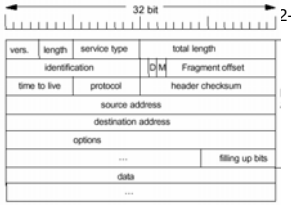
IP manages the sending of units of information through the Internet.

unit of information = datagram with header and data part

(division of a message into units is done by the TCP)

structure of an IP datagram:





Meaning of IP header fields:

vers. : version number of the IP protocol used

length : length of the header in number of 32-bit words(at least 5, maximally 15
⇒ header consists of maximally 60 bytes, i.e. option part maximally 40 bytes (often too small!!)

service type: specification of the desired quality of service
(priority, reliability, throughput, and delay; not used by current routers)

total length: length of the datagram in #bytes
(maximally 65k bytes; currently sufficient, but not in GigaBit-networks)

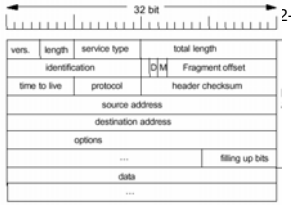
identification: identifies the datagram
(relevant for fragmented datagrams, all fragments of a datagram have the same identification)

D : "Don't fragment"= datagram must not be fragmented

M: "More fragments"= always set, except for the last fragment.

fragment offset: Position of the fragment in the datagram (in multiples of 8 bytes)

time to live: remaining life time of the datagram (fragment),
(should be time in sec.(max. 255), actually counts #hops)



Meaning of IP header fields: *(cont.)*

protocol : determines the protocol of the transport layer that should be used (TCP, UDP,...)

header check sum: one's complement of the sum of the 16 bit (half-)words in the header (without this field) using one's complement arithmetic.
(has to be recomputed at every hop, sum of all the fields must be 0 upon arrival)

addresses: IP addresses of source and destination

options : used for optional information, e.g.

security: secrecy level of the datagram

strict routing: specifies completely the path to follow

loose routing: specifies some routers on the path

record route: makes each router append its IP address

timestamp: makes each router append its IP address and timestamp

(options field length is always a multiple of 4 bytes, if necessary achieved by padding)

the routing options cannot be used really, due to the size of the Internet

Basic Router Algorithm:

```
procedure ROUTING(datagram);
begin
  dest:= DestinationAddress(datagram);
  NetworkAddress:= NetworkAddress (dest);
  if NetworkAddress in direct_NetworkAddresses then
                Send (datagram, ARP(dest))
  else if       datagram.option=Strict_Routing then
      Send (datagram, datagram.option.MyRoute)
  else if NetworkAddress in Routing_Table then
      Send (Datagram, Route[NetworkAddress])
  else if Default_Path then
      Send(datagram, default_router)
  else report_error;
end
```

Remark: The relevant information in the above code is the fact that a router has several sources of information for its routing decision. Real routers may use a different ordering for the different cases indicated above. An important part is the routing table.

How do we build routing tables?

Routing Tables

Determined by different routing algorithms :

- within autonomous systems: mainly OSPF
- between autonomous systems: e.g. EGP or BGP

more detailed:

- **OSPF= Open Shortest Path First:**
  - by local exchange of messages every router determines its own view of the network topology, including information on cost factors (delay, throughput, reliability,...)
  - every Router determines shortest path tree to all other routers (maybe different trees for different types of service) *(next section will give more information on routing)*
- **EGP= Exterior Gateway Protocol:**
  - allows to establish and test connections between exterior gateways
  - essentially an accessibility protocol
- **BGP= Border Gateway Protocol:**
  - fixes routing constraints like
    - no transit over autonomous system X
    - no Pentagon messages via Iraq
    - messages from Germany to France not via USA
    - messages from IBM to IBM not via Microsoft

Standard routing algorithms

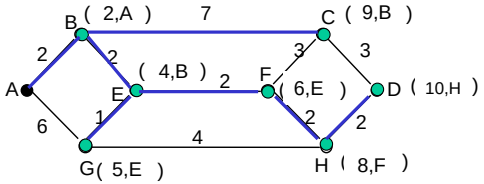
(following Tanenbaum: Computer Networks)

a) static routing

Shortest Path Routing: *Always send packet on shortest path from source X to destination Y.*

Prerequisite: Knowledge of network topology.
Computation of shortest paths.

- Use **Dijkstra's Algorithm**:
  - *Explore the network wavelike starting at node A and determine at every visited node the shortest path to A and the next node on this path. Move the wave further at the (border-)node having the shortest distance to A.*
- example:



run time: $O(\#nodes^2)$ (with naive data structure)
$O(\#edges + \#nodes \cdot \log \#nodes)$ (with Fibonacci-Heap)

- possible edge weights: #edges, edge/path delay, edge/path reliability

Flooding:

Send the packet to all direct neighbours you did not get it from.

⇒ produces large numbers of packets

- superfluous packets should disappear at some time (controlled by timer)
- important for applications, where packets have to be delivered in any case, even if links are destroyed (e.g. military applications)
- suitable for broadcast or multicast.
  (cf. echo-broadcast algorithm in "distributed algorithms")

- Alternative (more efficient but insecure):
  - **selective flooding**, i.e.
    send packets only over those links leading approximately into the correct direction

b) dynamic routing (also „adaptive routing“)

Distance vector routing:

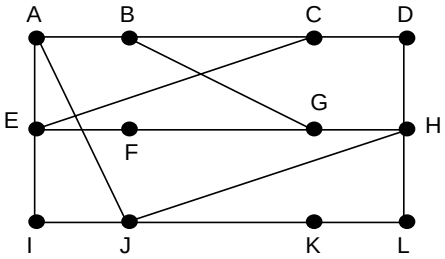
Every router sends the packet over the link, which - according to current local knowledge - belongs to the shortest path
(also called *distributed Bellman-Ford-Routing*, *Ford-Fulkerson-Routing*, *RIP*)

Method:

- Every router maintains distance table with 2 entries per destination:
  - shortest known (estimated) distance to destination
    (#routers/hops, delay, throughput,...)
  - link corresponding to this distance.
- Every router knows (an estimate of) the distance to its direct neighbours.
- Every router sends regularly its current distance vector to each direct neighbour (distances only, not links).
- On receipt of a distance table from neighbour X the router checks, if it should correct its own entries:
  - increase entry, if all neighbours report larger distances
  - decrease entry, if a neighbour reports a smaller distance

  (i.e. in every router A compute for every destination Z
  $d_{A,Z} = \min\{d_{A,X} + d_{X,Z} \mid X \text{ direct neighbour of } A\}$)

Example for distance vector routing:



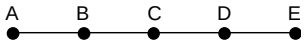
- *distance ≈ delay*
- current estimated delay to direct neighbours of node J : *(in ms)*
- A: 8 I:10 H: 12 K: 6
- node J receives the following distance vectors:

from \ to	A	I	H	K
A	0	24	20	21
B	12	36	31	28
C	25	18	19	36
D	40	27	8	24
E	14	7	30	22
F	23	20	19	40
G	18	31	6	31
H	17	20	0	19
I	21	0	14	22
J	9	11	7	10
K	24	22	22	0
L	29	33	9	9

⇒

Problems with distance vector routing:

- good news spreads fast
  - i.e.: the knowledge about fast links is quickly distributed over the network (#updates bounded by diameter of the network)
- bad news spreads slowly
  - i.e.: the knowledge about link failures is distributed extremely slowly
- example:
  - simple network:



  - development of estimated distances (#links) to A on failure at A or on link AB:

from B	C	D	E	
1	2	3	4	initially
3	2	3	4	after 1 exchange
3	4	3	4	after 2 exchanges
5	4	5	4	after 3 exchanges
5	6	5	6	after 4 exchanges
7	6	7	6	after 5 exchanges
	…			
8	8	8	8	(8 = longest path +1)

This is the **“count-to-infinity“ problem.**
⇒ distance vector routing not suitable for "dynamic networks"
(not used any more in the Internet but in other networks)

Link-State Routing:

Compute shortest paths based on regularly exchanged local knowledge of the topology.

Method:
Every Router must
1. Detect its neighbours and their network addresses.
2. Measure the delay (cost) to the neighbours.
3. Compile the results of (1) and (2) into a suitable packet.
4. Send this packet to all other routers.
5. Compute the shortest path to all other routers.

More detailed:
1. *Detect neighbours* :
   - Every router sends ”hello”-packet on every outgoing link.
   - On receipt of a “hello”-packet every router sends its address to the sender.
   - ”hello”-packets are sent regularly to document the router’s aliveness .
2. *Measure delay to neighbour X* :
   - Send “echo”-packet to X.
   - X immediately sends echo-packet back.

   ⇒ delay = (time to X and back) / 2

   (it may be reasonable to send the echo-packet several times and then compute the average delay)

Problems with measuring delay:

- Measured delay contains
  - propagation time on link to X
  - time from receipt to resending in X (“handling time" in X)

  but: still reasonable, since every packet has to be handled by every router anyway.

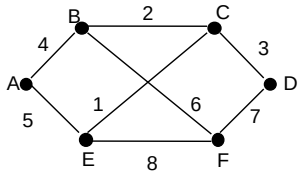
- Should one measure without load or with load?
  - without load: measured delay corresponds to ideal situation (which will not happen!)
  - with load: measured delay is influenced by current routing table (small delay leads to higher load, large delay leads to smaller load, i.e. routing table will change often!)

  ⇒ probably better, to measure without load and to distribute packet traffic depending on link performance, but in practice, we will always measure with some load!

Step 3: *Building Link State Packets*

- Link state packets consist of
  - local link data (list of neighbours and costs)
  - router identification
  - administrative data (sequence number, age)
- Example:

subnetwork:



- Link state packets for this subnetwork:

A	
seq.no.	
age	
B	4
E	5

B	
seq.no.	
age	
A	4
C	2
F	6

C	
seq.no.	
age	
B	2
D	3
E	1

D	
seq.no.	
age	
C	3
F	7

E	
seq.no.	
age	
A	5
C	1
F	8

F	
seq.no.	
age	
B	6
D	7
E	8

Step 4: *Distributing Link State Packets*

- New link state packets are sent by flooding with acknowledgement („echo-flooding“)
- Distinction between old and new link state packets by sequence numbers and age.
- Too old packets are deleted.
- Several measures to reduce communication overhead.

Step 5: *Computing shortest paths:*

- As soon as the information on the network topology is complete, every router can compute the shortest paths to all other routers (wrt the chosen metric), using Dijkstra’s algorithm.

Advantages of Link State Routing:

- Routers can react fast to changes in network topology.
- No count-to-infinity problem.

c) Hierarchical Routing

- Growth of network leads to very large routing tables.
- therefore:
  - Separation of network into regions
  - Every router knows only the internal structure of its own region
  - Every region has designated router for "inter regional" packet traffic.
  - For very large networks possibly multilevel hierarchy

  ⇒ significantly smaller routing tables

  - but: hierarchical routing may lead to longer “shortest paths” than "flat" routing.
- Algorithm OSPF uses link state routing in connection with hierarchical routing, broadcast, and multicast.
- Complete specification of OSPF (July 1997) in RFC2178:
  - `http://www.cis.ohio-state.edu/htbin/rfc/rfc2178.html` (about 200 (!!) pages)

Further tasks of network layer:

- *Congestion control*, i.e.
  - Monitor the system to determine, whether routers ”overflow" and packets get lost
  - Reroute traffic, to avoid congestion.
- *Routing for mobile hosts:*
  - Even if host R moves around in the network, it should be reachable via its IP-Address.
  - Realised by having R register with the foreign router which informs the ”home router”.
  - Whenever the ”home router" gets data for R, it sends the data to the foreign router and informs the sender about the current address (which remains valid for this transfer of data only!)

New Internet Protocol IPv6: *(see RFC 2460 and http://www.ipv6.org/)*

goals:

- support billions of hosts, even in case of inefficient assignment of addresses
  - motivation: in our future there will be a tremendous number of hosts on the Internet (every TV, handy, ...)
- Reducing the size of routing tables.
- Simpler protocol to speed up the traffic.
- Higher security than with today’s IPv4 (by authentication and integrity protection).
- More emphasis on types of service, especially for real time applications.
- Better support of multicasting.
- Improved possibilities for moving around without change of address (*mobile IP*).
- Larger flexibility for future changes of the protocol.
- Support of old and new protocols in coexistence for several years.

The new addressing format of IPv6

- 16 byte = 128 bit = 8 groups of hexadecimal digits, separated by colons:
  - e.g. 8000:0000:0000:0000:0123:4567:89AB:CDEF

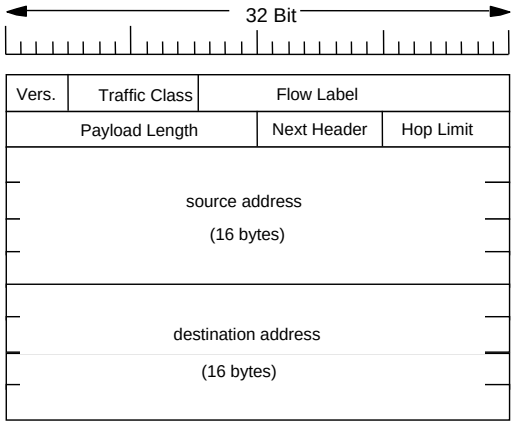
  ⇒ $2^{128} \approx 3 \cdot 10^{38}$ different addresses
  (= 340 282 366 920 938 463 463 374 607 431 768 211 456)
  i.e. about 10^{24} addresses per square meter of the earth’s surface!
- Address space will not be used efficiently.

Currently planned separation by prefixes:

- 80 leading zeroes: old IPv4 addresses ::128.13.64.5
- 0000 001 OSI-NSAP addresses
- 0000 010 Novell NetWare IPX-Adresses
- 010 addresses for service providers
- 100 addresses for geographic regions
- 1111 1110 10 link specific local addresses
- 1111 1110 11 site specific local addresses
- 1111 1111 Multicast
  (after the prefix 4 bit flagfield, 4 bit scope field, 112 bit group identifier;
  possible scopes: site, organisation, planet (value 14, value 15 reserved for extensions to other planets, solar systems, galaxies...)
- further type of addressing:
  - *Anycasting*, i.e. the destination is an arbitrary member of the addressed group

The new header format of IPv6:

- *Version*:
  IPv4 or IPv6
- *Traffic Class* : (also *Priority*)
  specifying information for quality of service, not fixed so far
- *Flow Label:*
  labeling of pseudoconnections (i.e. packet sequences) for flow control
- *Payload Length*:
  #bytes after the 40 byte long header
- *Next Header*:
  information on following headers (called extension headers), or to which transport protocol the packet has to be passed.
- *Hop Limit* :
  corresponds to “time to live” in the IPv4 header



possible extension headers (in this order):

- *Hop-by-hop options* : various information for routers that every router has to check (e.g. indicating *Jumbograms,* i.e. extra long datagrams)

the following options are checked at the destination address only:

- *Routing* : Definition of complete or partial routes
- *Fragmentation*: Management of datagram fragments
- *Authentication*: verification of the sender's identity
- *Encryption security data*: Information on the encrypted contents (key,...)
- *Options for destination*: additional information for the destination

⇒ – no header check sums
– simple standard header with corresponding shorter handling delay
– larger flexibility by extension headers
– improved security by authentication and encryption options

transformation procedure (plan):

- start with isolated IPv6 - islands
- communication between islands via "tunnels"
- gradual merger of islands
- estimated time for transformation: about one decade

• Start of IPv6 has been delayed

some of several reasons:

– use of "Classless InterDomain Routing"-protocol (CIDR) and the practice of "lending" addresses (see RFC1519 and RFC2008)

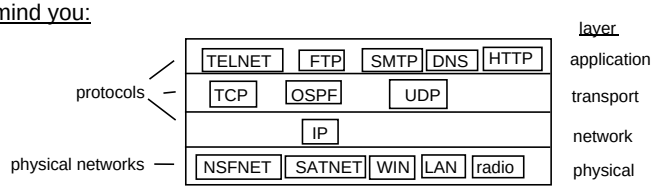
– Introduction of DHCP-Servers, providing "dynamic" IP-Adresses:

- Dynamic Host Configuration Protocol provides IP-Adresses (and other information) "on demand" for hosts connecting to a network.

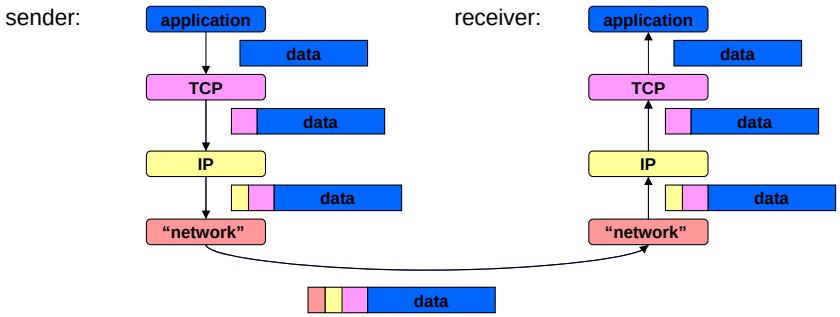
- Transition to IPv6 is happening but it remains a controversial issue, some people (service providers) prefer to stick to extended versions of IPv4

TCP - the Transmission Control Protocol *(transport layer)*

- to remind you:



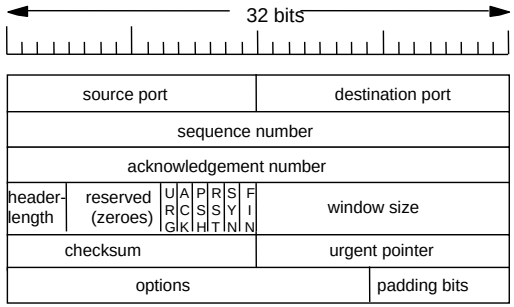
- Application sends data (stream of bytes) from sender to receiver using TCP/IP.
- Every layer adds a header with layer specific information (sender) and removes it again (receiver):



Properties of TCP/IP

- IP provides connectionless, unreliable service
- TCP takes care of establishing a (logical) connection and of providing the necessary reliability
- TCP gets byte stream ("message") from the application
- messages are divided into segments depending on the network's (or the logical connection's) link capacities (maximally 64k bytes, often around 1.500 bytes)
- TCP takes care of flow control (#packets / s)
- every segment gets a header; segment and header together form a datagram.

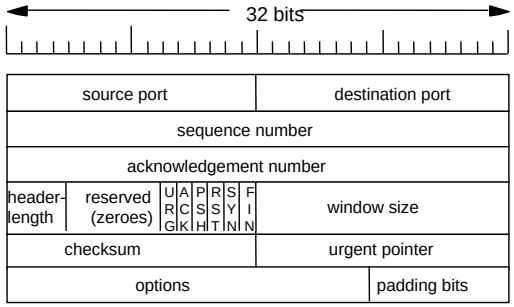
TCP-header format :



- ***source port, destination port*:**
  the two ports are the end points of the logical connection ("sockets"), together with IP-addresses of source and destination they identify the connection
  rsp. the two processes associated with this connection.
  Port numbers are managed by IANA – Internet Assigned Numbers Authority
  http://www.iana.org/assignments/port-numbers
  - ”well known ports" (< 1023): e.g. for ECHO (7), FTP (21), TELNET (23), SMTP (25), TIME (37), FINGER (79), HTTP (80),…
  - “Registered Ports” (1024 – 49151): mostly registered with IANA for specific application programs
  - “Dynamic and/or Private Ports” (49152 – 65535): free choice for application program

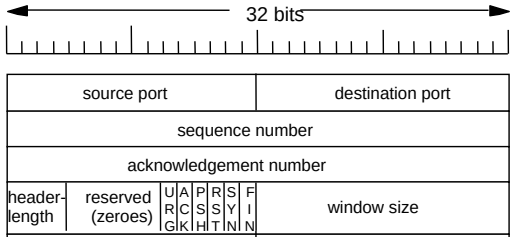
meaning of header fields :

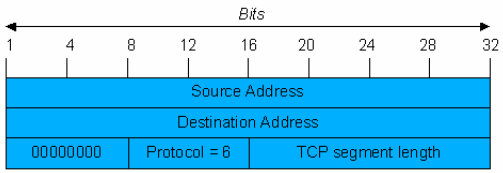
- *sequence number*:
  all bytes of a message (byte stream of this TCP-connection) are numbered; it is the number of the first data byte within this datagram.
- *acknowledgement number*:
  dest. host sends x+1 to acknowledge correct receipt of all bytes up to number x.
- *Header length*:
  #words (4 bytes) in the header, at least 5 (rsp. start of data part of the datagram)
- *Flags*:
  - URG:
    urgent pointer is pointing to data, which has to be read immediately (corresponds to an interrupt)
  - ACK:
    acknowledgement number is valid
  - PSH:
    immediate delivery of data requested (important for interactive applications)
  - RST:
    reset the connection in case of faults
  - SYN: ("synchronize")
    used during connection build-up
  - FIN: ("final")
    indicates the final datagram, used for closing a connection



meaning of header fields :

- *window size*:
  used for flow control *(see below)*
- *checksum*:
  1’s complement of the 16 bit sum of all 16 bit fields of the datagram (header (with checksum 0) + data) and of the ”pseudo header" (which is not sent by TCP but contained in the IP header).
  Allows the receiver to check for correct transmission of the datagram.
- *Options*:
  additional functions not contained in the standard header, e.g.
  - specification of the maximally accepted segment size (every host must accept at least 536 bytes of data, i.e. 576 bytes including the TCP/IP header, maximally 65k bytes).
    - small values (e.g. "1") would lead to bad utilisation (maximally 1/40 of the available bandwidth!),
    - large values lead to increased fault rate due to fragmentation
  - setting a scaling factor for the window size (e.g. increase to 32-bit values)
  - request for specific segments by receiver to avoid redundant resending of data if only some datagrams are missing
    (NAK= negative acknowledgement)

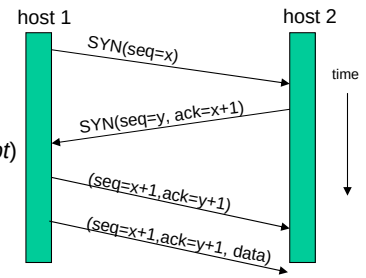




TCP - connection management

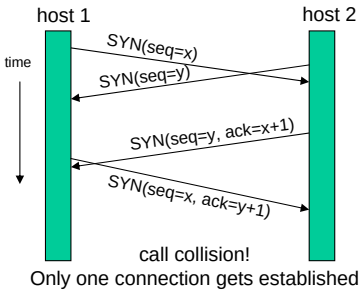
a) connection establishment

- server waits for incoming connection requests
- client (sender) executes CONNECT primitive with the following parameters
  - IP address of server (receiver)
  - desired port for this connection
  - maximally handled segment size
  - optionally more user data (password,...)
- TCP establishes connection using a *three-way-handshake protocol*:
  - host1 sends datagram with (adequately determined) sequence number x and SYN=1 (*connection request)*
  - host2 sends datagram with (adequately determined) sequence number y , SYN=1, ACK=1, and ack-number x+1 (*connection accept*)
  - host1 acknowledges the acknowledgement *(connection established)*
  - after this, host1 can start sending data



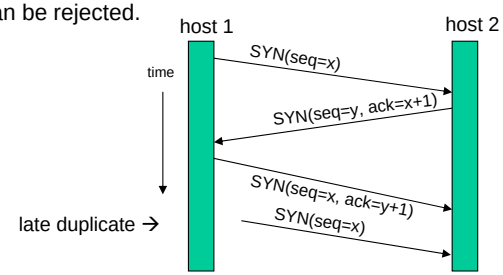
Fault handling, collision control

- When sending a datagram, a timer is started (actually, several timers are used). "Time out" leads to appropriate actions, e.g. retransmission or connection closure.
- If two connections are requested simultaneously, only one gets established:



Only one connection gets established

- Since both hosts remember the datagrams they sent (in particular the sequence numbers x and y) , late duplicates cannot lead to incorrect connections,
  i.e. erroneous acknowledgements can be detected and the duplicate connection request can be rejected.



(b) connection closure

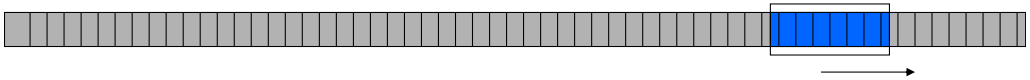
- If a host does not intend to send more data, it sends a FIN datagram.

⇒ one direction of the connection is closed, the other continues to exist until the other host sends a FIN datagram or a time out occurs.

c) sending data

- TCP has to guarantee fault free transmission of data

  ⇒ received datagrams have to be acknowledged.
- Use of a "sliding window" (*sliding window protocol*):



  – window size f determines how many datagrams may be sent, before the sender has to wait for an acknowledgement.

  – f is negotiated dynamically between sender and receiver depending on the receiver's buffer capacity ("maxwinsize") and the network's current "transport performance".

  – The acknowledgement field always contains the maximal received sequence number (+1), up to which all bytes have been received correctly; the acknowledgement is done in the next available (or reasonable) datagram ("piggybacking").

  – If a segment with sequence number x does not get acknowledged in time, all segments starting with byte x are retransmitted

⇒ even if only the first segment is missing, all others have to be resent, too.
(may be resolved by NAK-Option)

Dynamic flow control:

- Loss of too many datagrams may be caused by too large window size.
  (e.g. if a subnet on the path from sender to receiver has a too small capacity, congestion can lead to datagram losses)

  ⇒ the window size could be reduced (e.g. halved) to achieve better transmission quality ; this could be repeated until the quality is acceptable.

- Currently used method: **slow start**:
  – Start with a small window (w=1 segment) and a threshold value t=maxwinsize.
  – Increase (double) w, if w ≤ t and all sent segments got acknowledged in time.
  – Increase w linearly, if t < w <maxwinsize and all sent segments got acknowledged (different linear functions are used).
  – If there are faults, set t:=w / 2 (i.e. half the last acceptable window size) and start again with w=1 .

  ⇒ data flow adapts dynamically to the network capacity available for this connection.

This is also called "congestion control" (**slow start** + **congestion avoidance**).

Problems with TCP:

- Loss of few segments within the (sliding) window can lead to retransmission of many segments (unless negative acknowledgements are used).
- Slow start may lead to frequent changes in throughput

⇒ **This is unacceptable for multimedia applications, in particular for transmission of audio signals.**

- Flow control is based on transmission properties **valid for transmission on wires** (i.e. loss of datagrams due to congestion)
  – **but**: on radio- or satellite connections losses occur due to transmission faults which generally do not depend on traffic

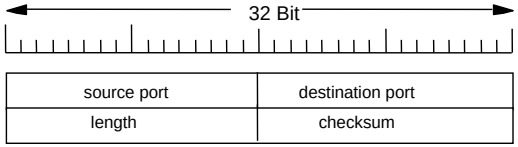
  ⇒ dynamic flow control leads to even worse transmission rates, instead, one should retransmit datagrams as quickly as possible !
- **TCP does not know about the type of interconnections, therefore it cannot decide for the most reasonable method of flow control.**

further transport protocol: UDP = User Datagram Protocol

- provides connectionless, unreliable transport service

  ⇒ ***IP service gets extended up to the application layer***

- UDP is used for the transmission of single datagrams
  (essential for many client/server applications with short requests and replies)
- **UDP-Header:**



- *source port, destination port:* same meaning as in TCP, but the assigned functions differ
- *length*: total length of the datagram (header + data), i.e. at least 8.
- *checksum*: same meaning as with TCP, i.e. 1's complement of the 16-bit fields in UDP-header, data part and pseudo header
- UDP is mainly used to grant application programs direct access to the IP layer.

  ⇒ if UDP is used, the application program has to take care of reliability, flow control etc.

ATM – Asynchronous Transfer Mode

- Connection oriented ("virtual channels", "virtual circuits")
- fixed cell size (cell = unit of transmission):
  - 53 bytes = 5 bytes header plus 48 bytes data
- sending and receiving in the same sequence on virtual channels

  but: if cells from the same byte stream are sent over different virtual channels, their order is not guaranteed

service classes:

- CBR: Constant Bit Rate
- RT-VBR: Variable Bit Rate, Real Time (for video conferencing)
- NRT: Variable Bit Rate, NonReal Time
- ABR: Available Bit Rate
- UBR: Unspecified Bit Rate (background data transfer)

service quality:

- class and quality of service are determined before establishing a connection. If the virtual channel cannot support this class or quality, the channel is not established.

typical transmission rates:

- 155 Mb/s and 622 Mb/s
- extremely important for:
  - HDTV (High Definition Television)
  - Video-on-demand, audio-on-demand
- GigaBit technology may make expensive ATM technology obsolete! ?