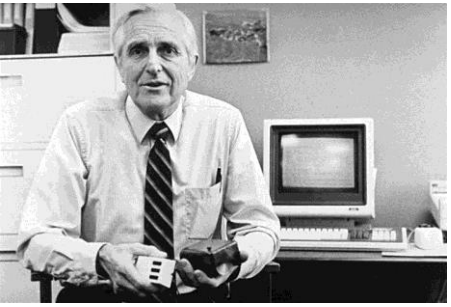
Delft University of Technology

Telecommunications Networking

1. Introduction
2. Local Area Networking
3. Error Control and Retransmission Protocols
4. **Architectural Principles of the Internet**
5. Flow Control in Internet: TCP
6. Routing Algorithms
7. Routing Protocols
8. The principles of ATM
9. Traffic Management in ATM
10. Scheduling
11. Quality of Service
12. Quality of Service routing
13. Peer-to-peer networks

Start of the Internet

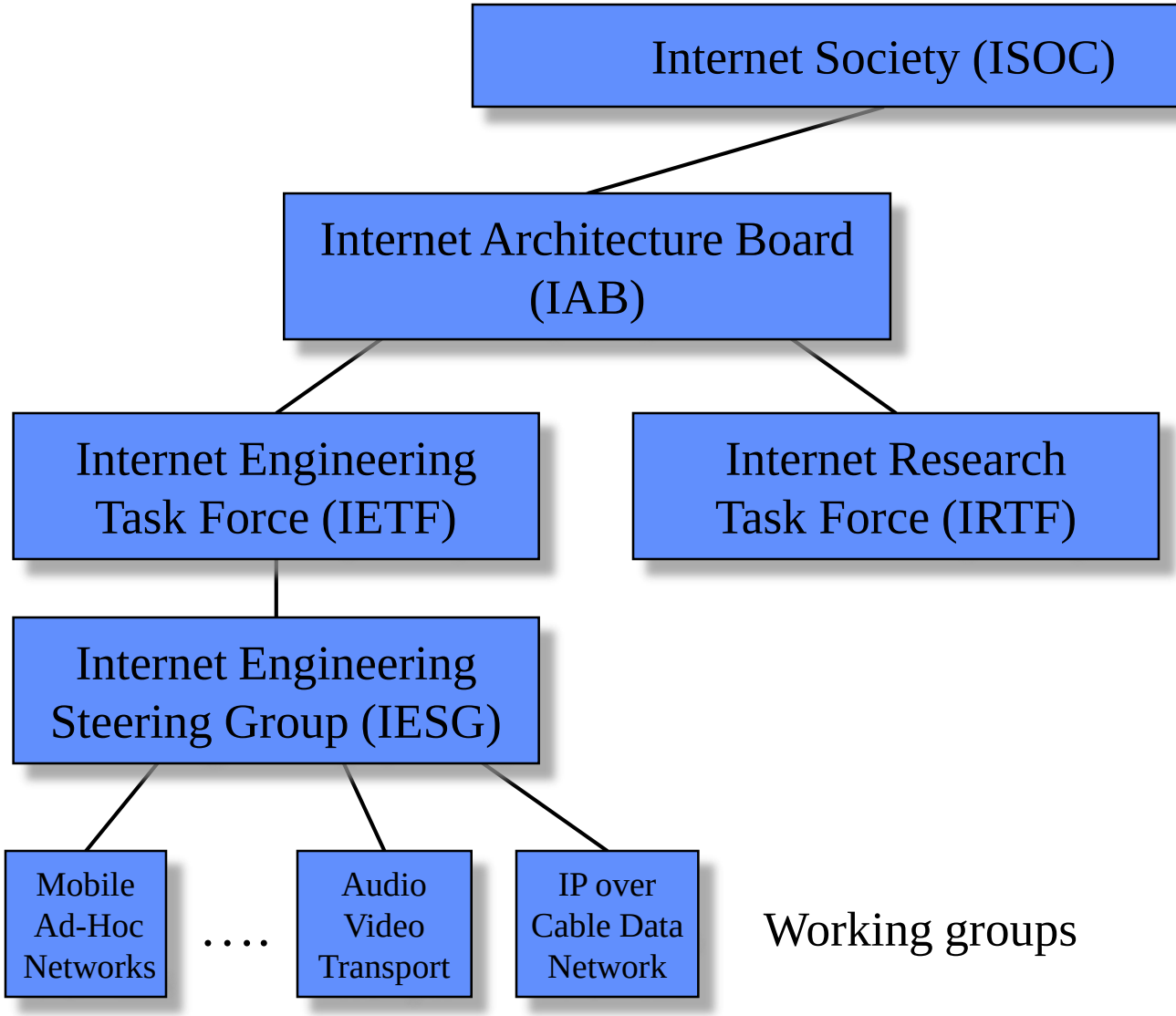


Douglas Engelbart

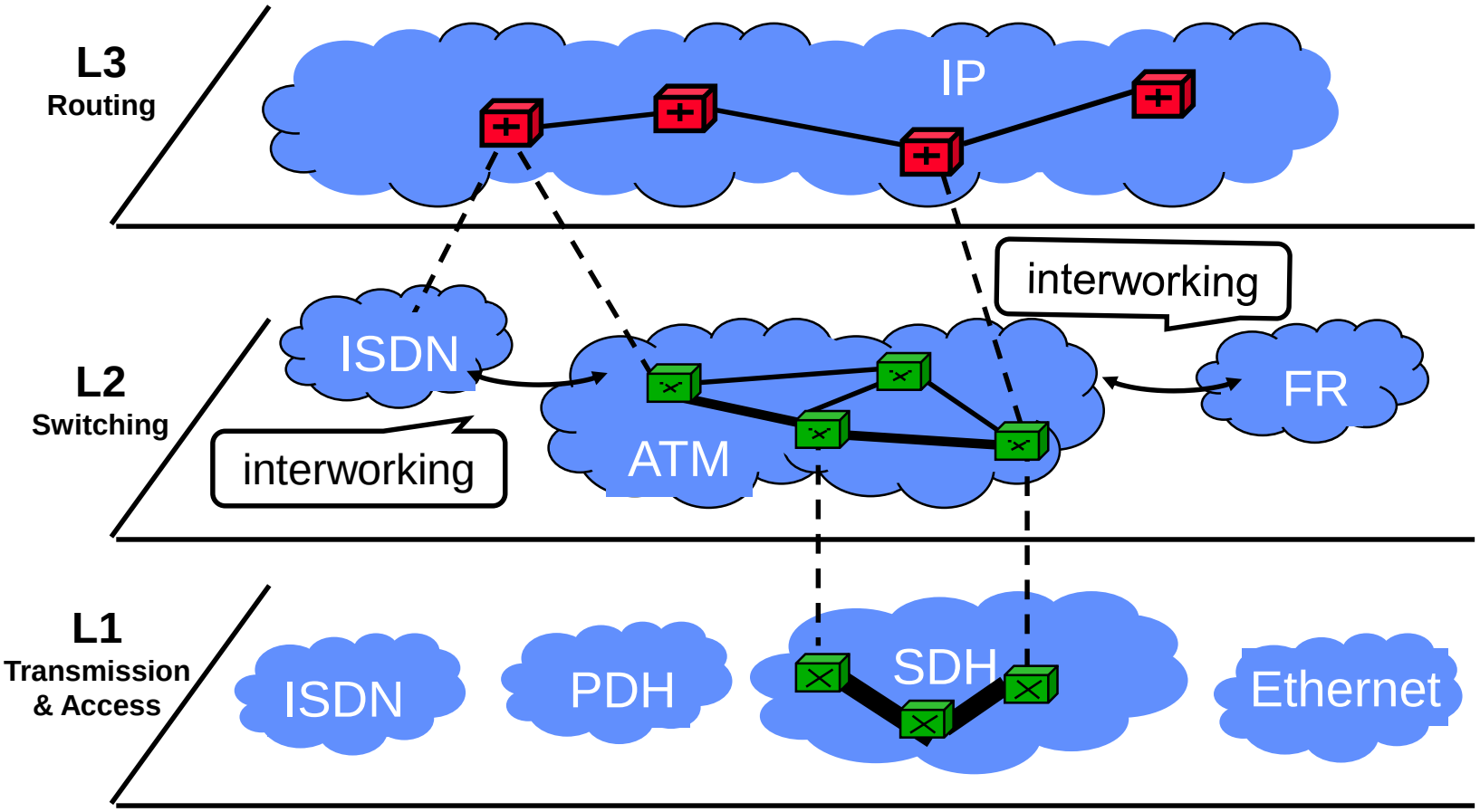
- Login attempt from UCLA to SRI (Stanford)
- L (ok)
- O (ok)
- G (crash)

Leonard Kleinrock

Authorities for Internet Standards



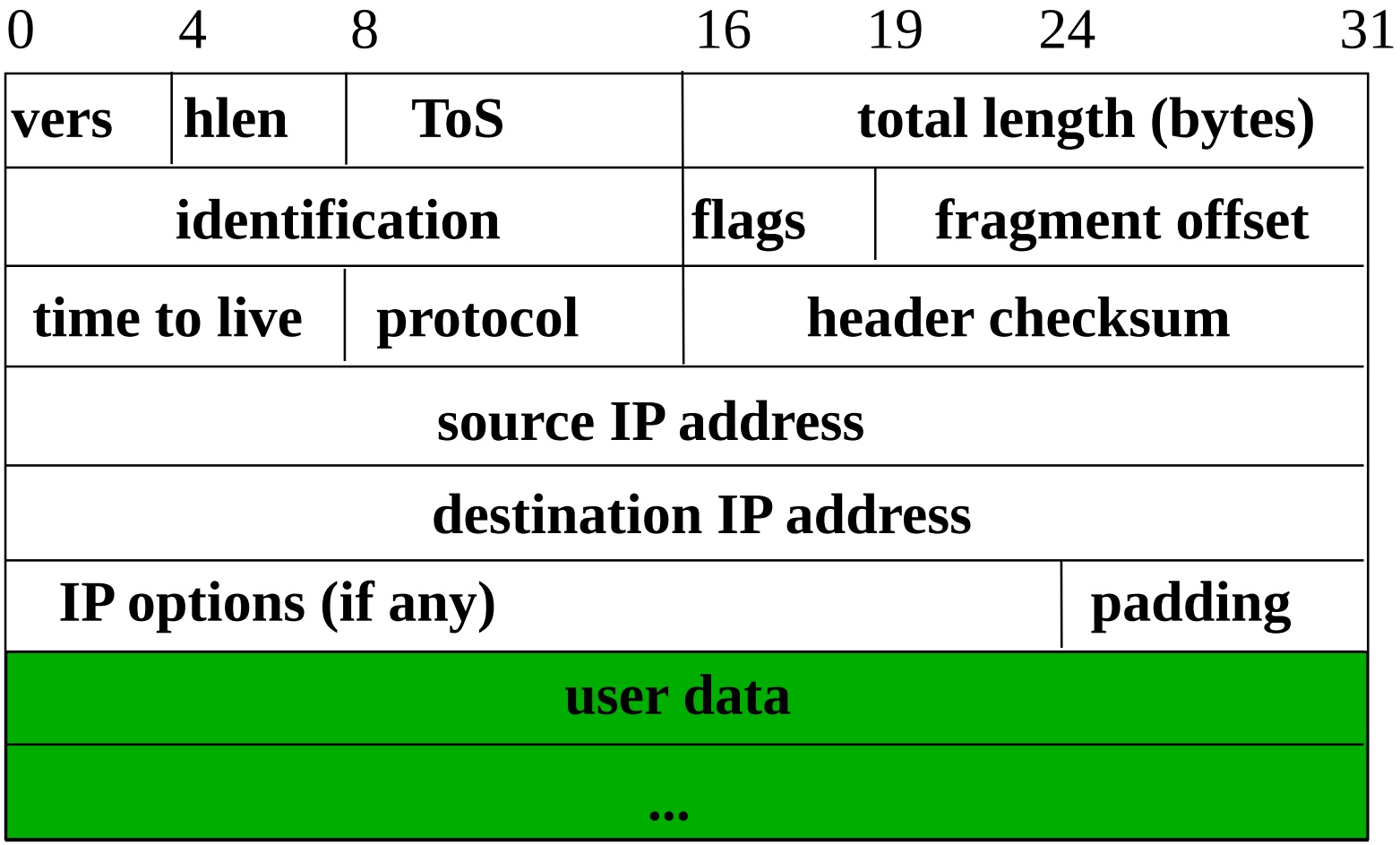
Layered Network Model



IP

- The Internet Protocol (**IP**) defines an unreliable, connectionless, best-effort delivery mechanism:
  - specification of basic unit of transfer (packet)
  - IP software performs the routing function
  - IP includes the rules for unreliable, connectionless, best-effort delivery

IPv4 packet


0
4
8
16
19
24
31
vers
hlen
ToS
total length (bytes)
identification
flags
fragment offset
time to live
protocol
header checksum
source IP address
destination IP address
IP options (if any)
padding
user data
...

ICANN

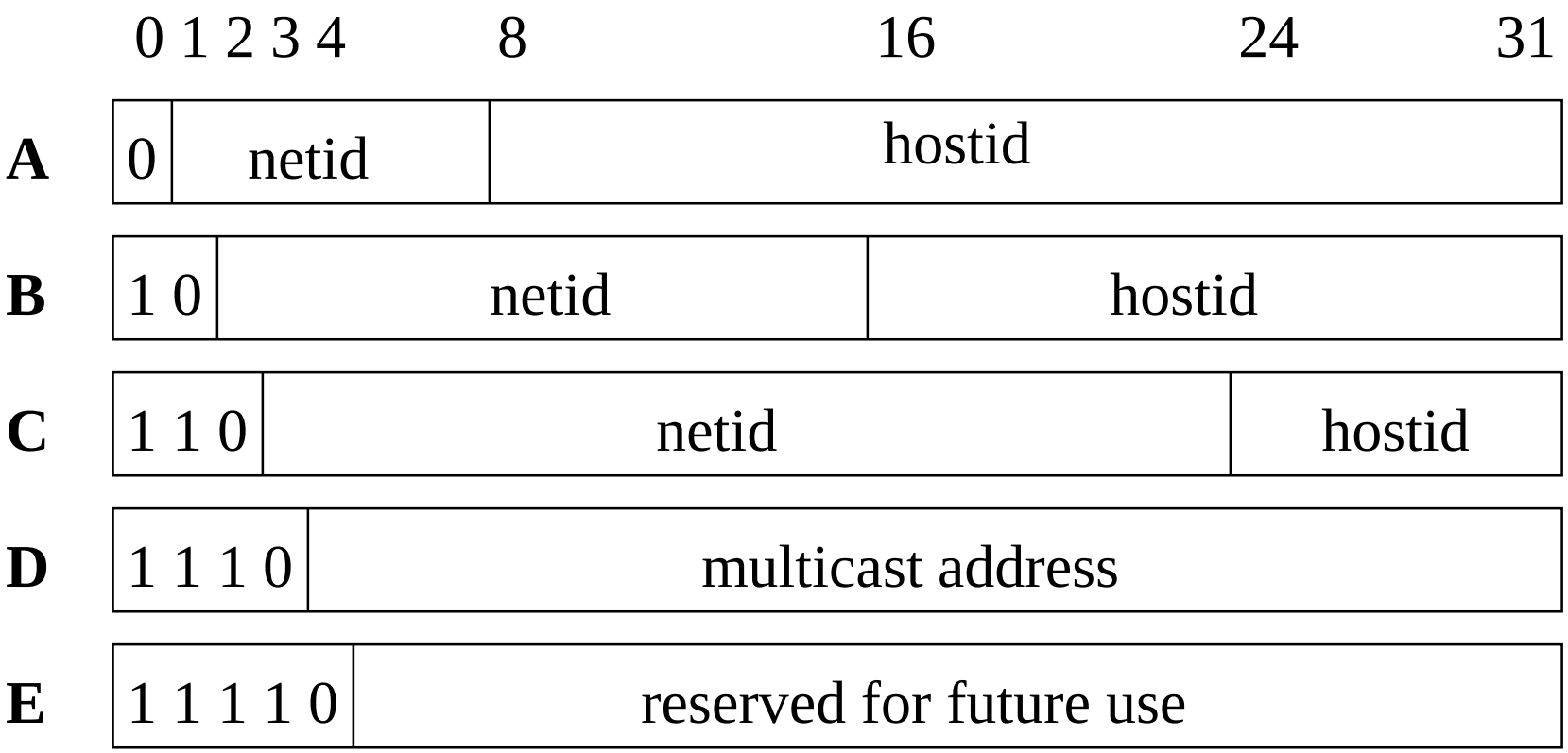
- Internet Registry (IR): IP address space allocation, protocol parameter assignment, DNS management

- Internet Corporation for Assigned Names and Numbers. Regional IRs:
  - ARIN (North America)
  - APNIC (Asia-Pacific)
  - LACNIC (Latin America, Caribbean)
  - RIPE NCC (Europe):
    - Local IRs (national):
      - ISPs:
        - » End-user

Addressing

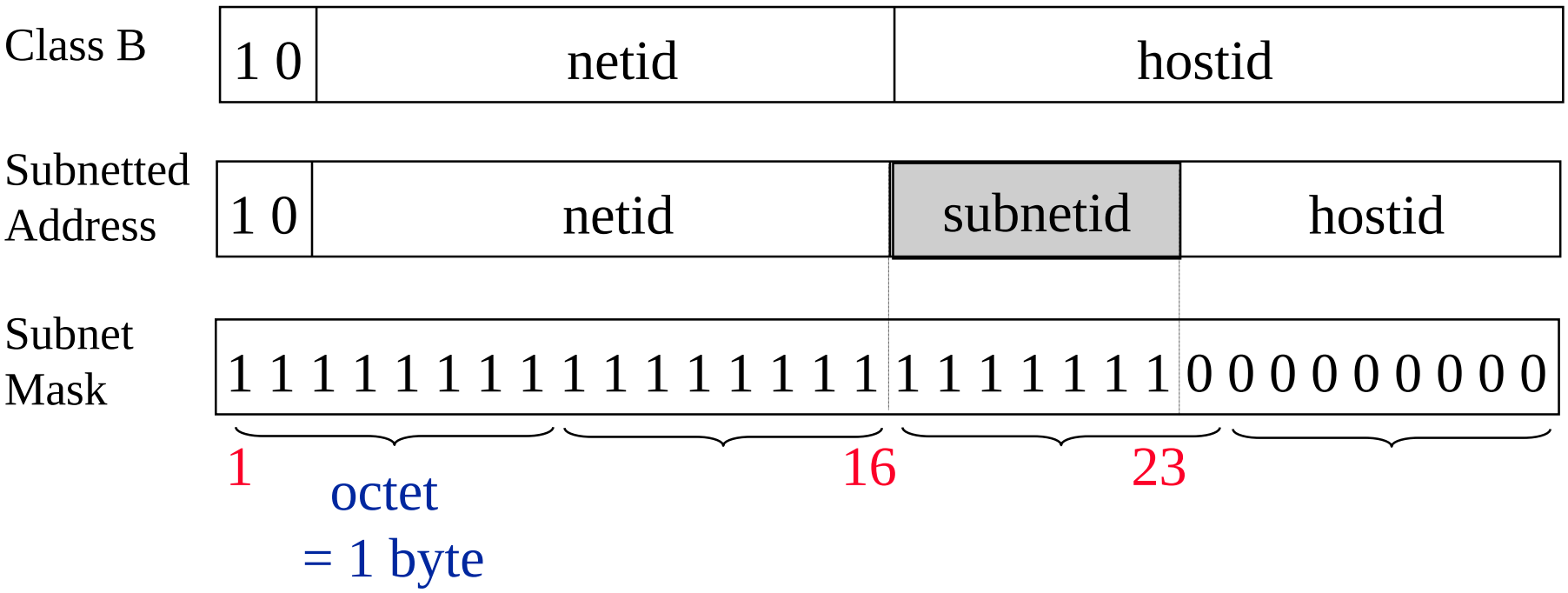
- Geographic (physical) versus logical (mobility)
- 32-bit (4-byte) addresses: *x.y.z.w*
- Netid and hostid
  - Subnet address
  - Classless Interdomain Routing (CIDR)

Class-based Addressing



E.g. 129.82.6.25 belongs to the class B network 129.82.0.0
(129 = 1000 0001)

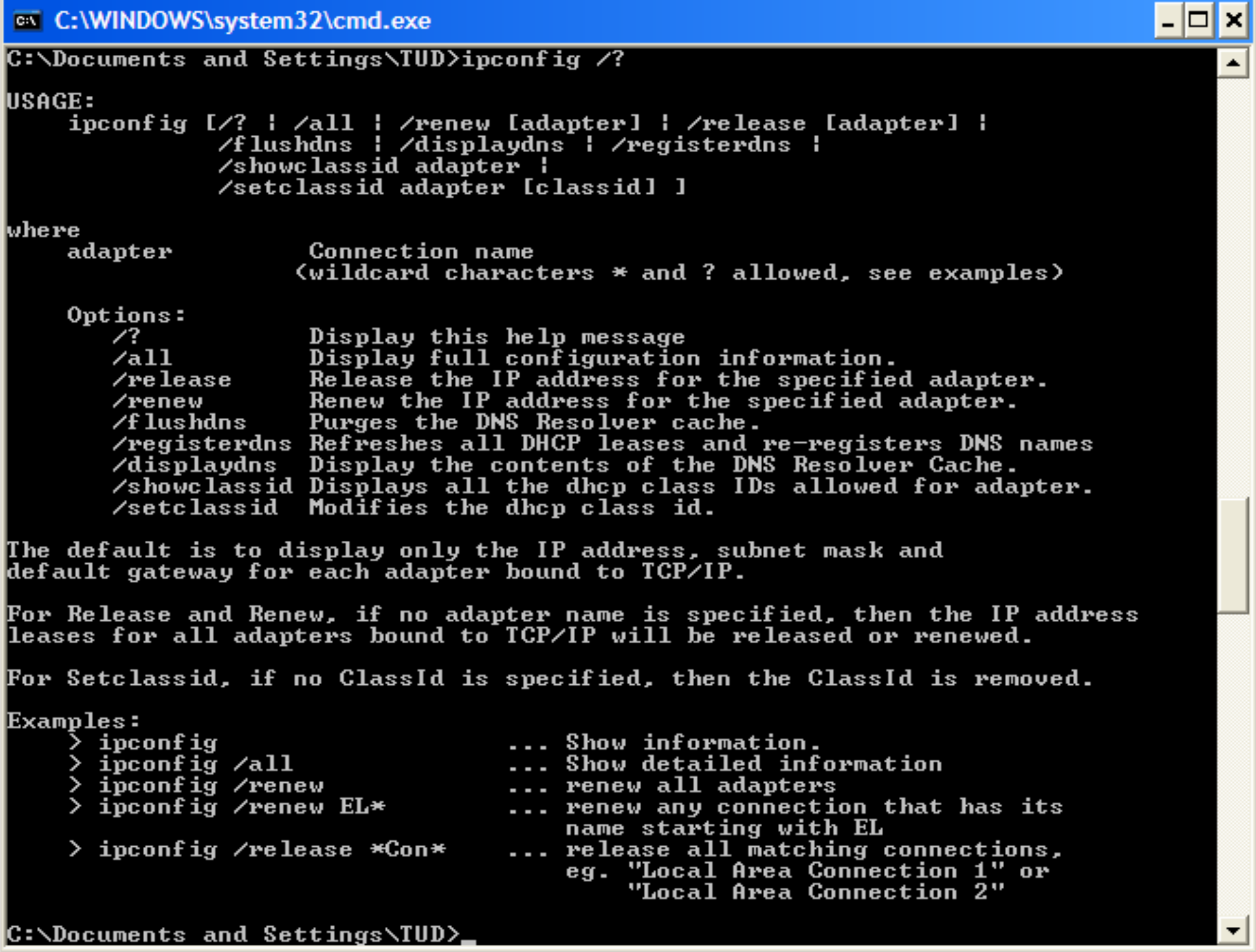
Subnet Addressing



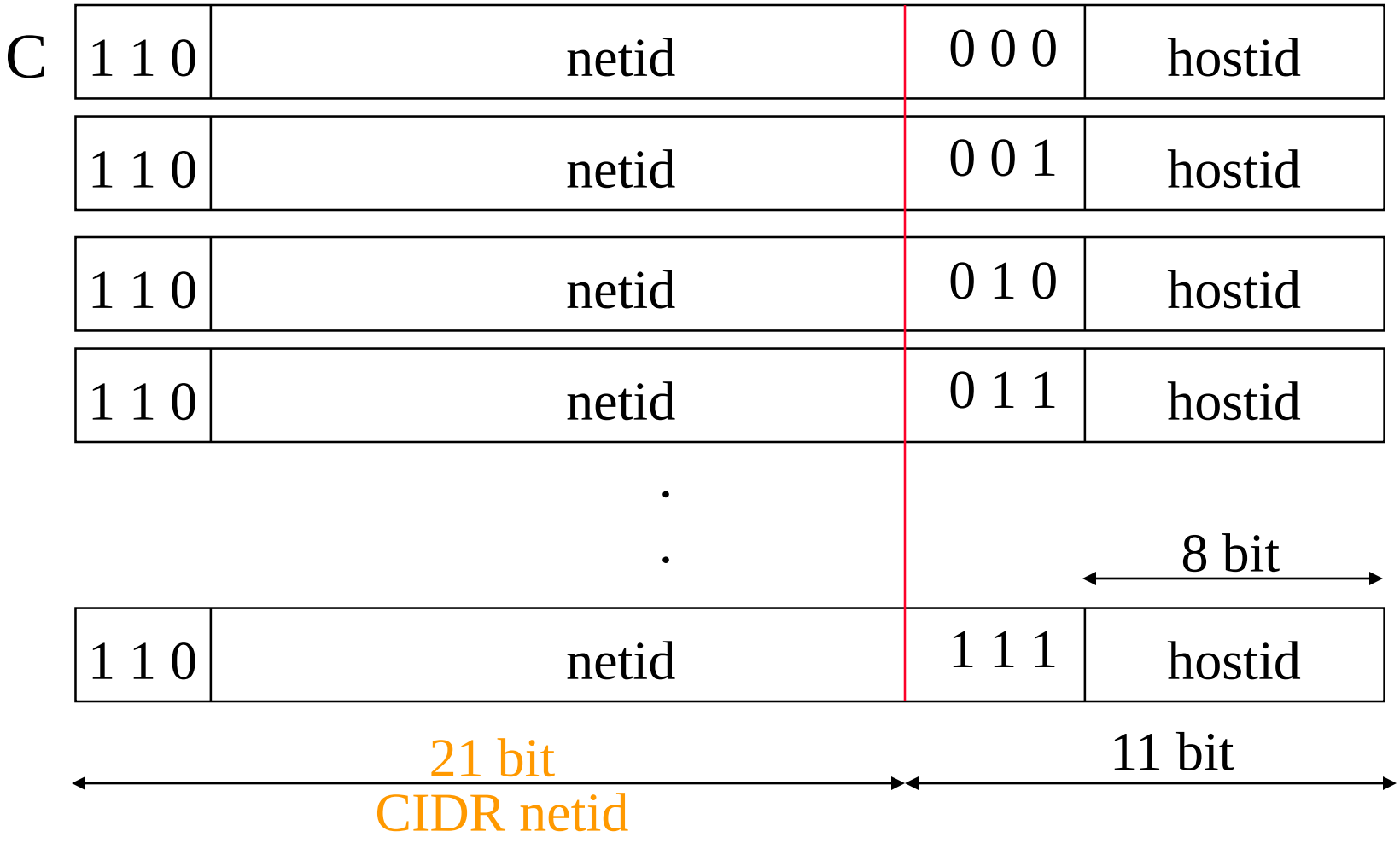
Subnet address = IP address ⊗ subnet mask

Subnet Mask: consists here of 23 one bits = 255.255.254.0

Subnet: *x.y.z.w/v*, where *v* is an integer smaller than 32


C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\TUD>ipconfig /?
USAGE:
ipconfig [/? | /all | /renew [adapter] | /release [adapter] |
/flushdns | /displaydns | /registerdns |
/showclassid adapter |
/setclassid adapter [classid]]
where
adapter Connection name
(wildcard characters * and ? allowed, see examples)
Options:
/? Display this help message
/all Display full configuration information.
/release Release the IP address for the specified adapter.
/renew Renew the IP address for the specified adapter.
/flushdns Purges the DNS Resolver cache.
/registerdns Refreshes all DHCP leases and re-registers DNS names
/displaydns Display the contents of the DNS Resolver Cache.
/showclassid Displays all the dhcp class IDs allowed for adapter.
/setclassid Modifies the dhcp class id.
The default is to display only the IP address, subnet mask and
default gateway for each adapter bound to TCP/IP.
For Release and Renew, if no adapter name is specified, then the IP address
leases for all adapters bound to TCP/IP will be released or renewed.
For Setclassid, if no ClassId is specified, then the ClassId is removed.
Examples:
> ipconfig ... Show information.
> ipconfig /all ... Show detailed information
> ipconfig /renew ... renew all adapters
> ipconfig /renew EL* ... renew any connection that has its
name starting with EL
> ipconfig /release *Con* ... release all matching connections,
eg. "Local Area Connection 1" or
"Local Area Connection 2"
C:\Documents and Settings\TUD>

CIDR


C
1 1 0
netid
0 0 0
hostid
1 1 0
netid
0 0 1
hostid
1 1 0
netid
0 1 0
hostid
1 1 0
netid
0 1 1
hostid
.
.
8 bit
1 1 0
netid
1 1 1
hostid
21 bit
CIDR netid
11 bit

Reserved IP ranges

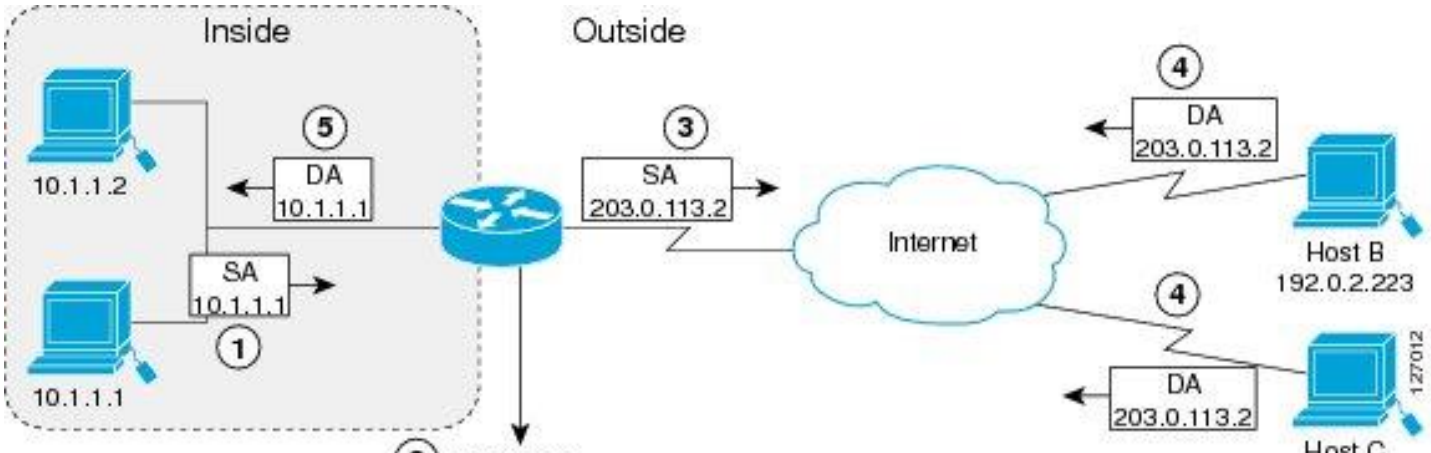
- Class-based addressing loosened into ranges using subnet masks
- Private networks
  - 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16
- Link-local
  - 169.254.0.0/16 (The “the-Internet-is-down” range)
- Multicast
  - 224.0.0.0/4
  - 255.255.255.255/32 (local subnet only)

Network Address Translation

- Remaps private address outgoing flows to public address
- Hides internal network topology and addresses
- Conserves IP address usage
- Incoming connections handled through port mapping
  - Hence problematic with non-TCP or -UDP flows

Network Address Translation

- Outgoing flows automatically added
- Incoming flows need prior existing rules

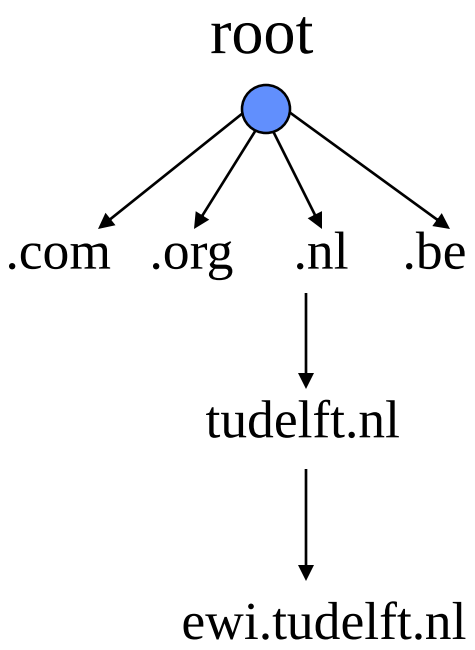


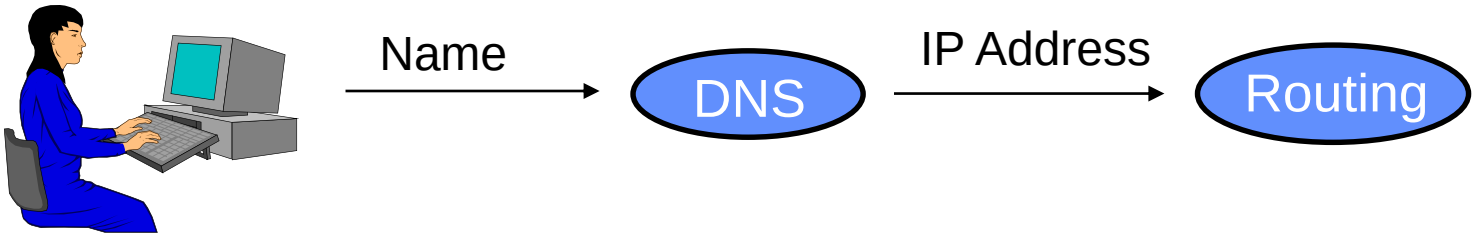
Protocol	Inside Local IP address:port	Inside Global IP address:port	Outside Global IP address:port	Outside Local IP address
TCP	10.1.1.2:1723	203.0.113.2:1723	198.51.100.4:23	198.51.100.4:23
TCP	10.1.1.1:1024	203.0.113.2:1024	192.0.2.223:23	192.0.2.223:23

[http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_nat/configuration/15-mt/nat-15-mt-book/iadnat-addr-consv.html]

Domain Name System (DNS)

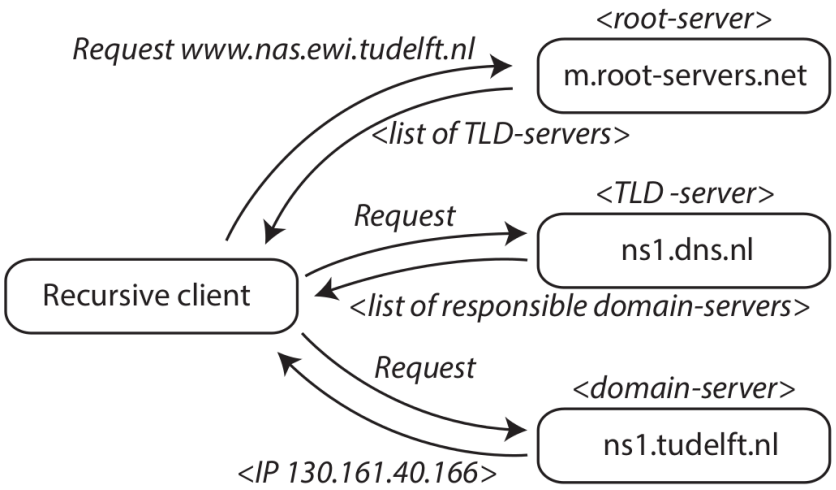
- Humans work best using the name of a host (e.g. ewi.tudelft.nl)
- IP Addresses used to identify a host, and for routing (e.g. 130.161.40.75)
- DNS
  - deals with mapping: Name ⟷ IP Address

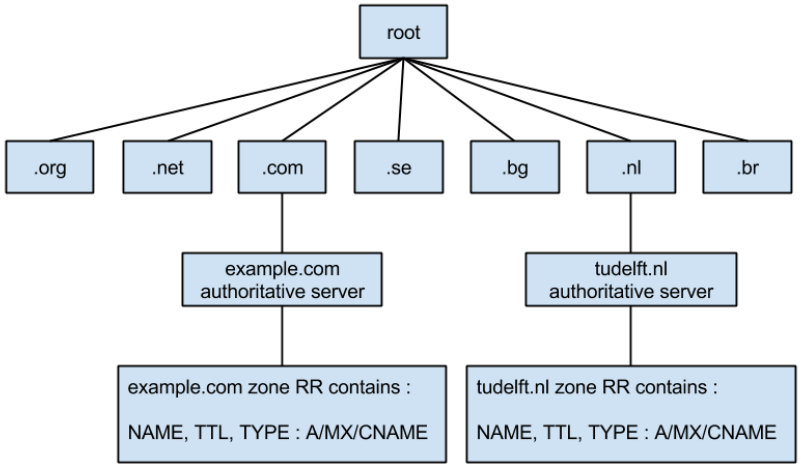


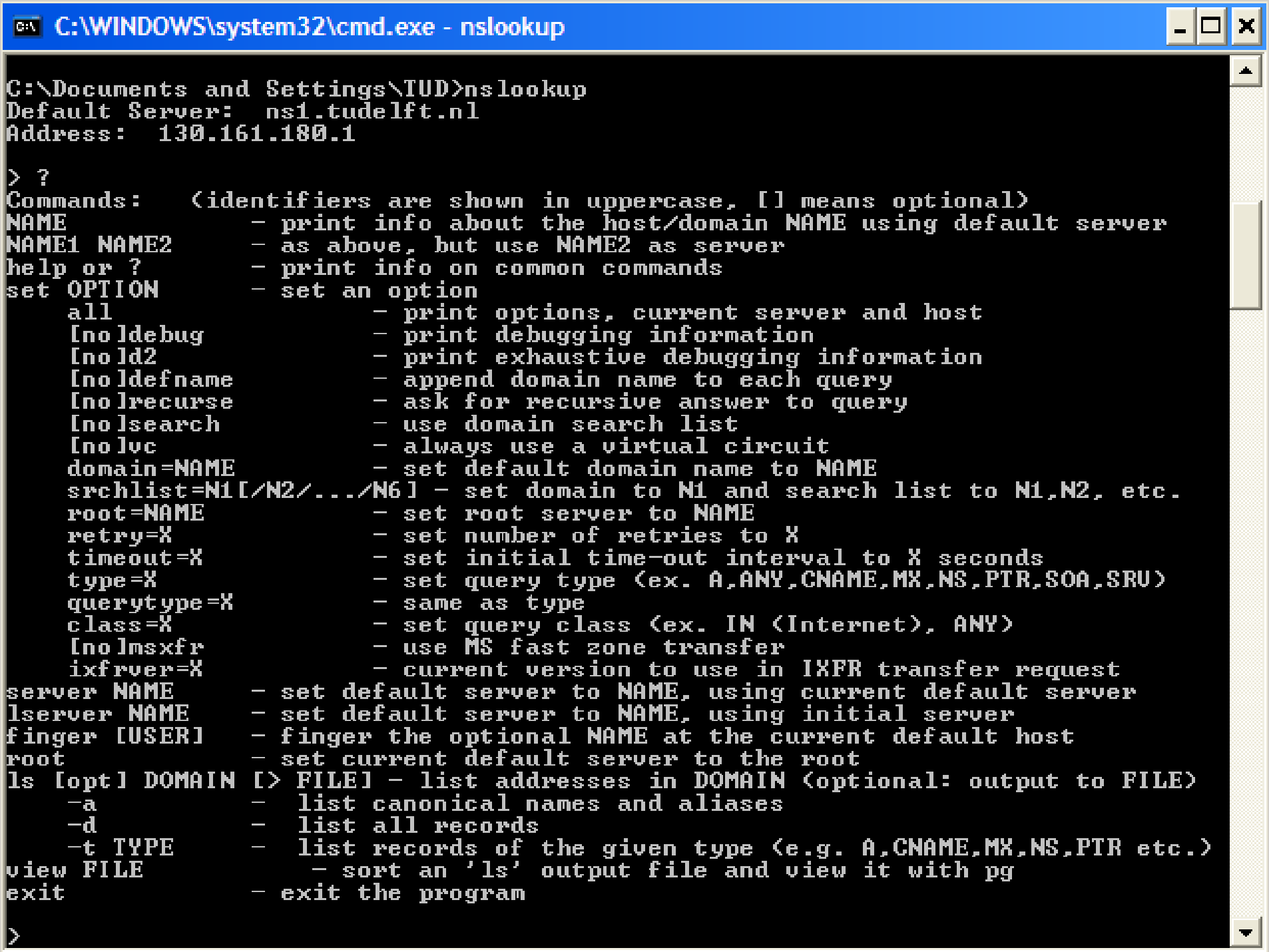


Domain Name System (DNS)

- Hierarchically distributed database
  - Root Domain -> Top Level Domain -> Authoritative domain
- Clients use recursive servers
- High level of caching



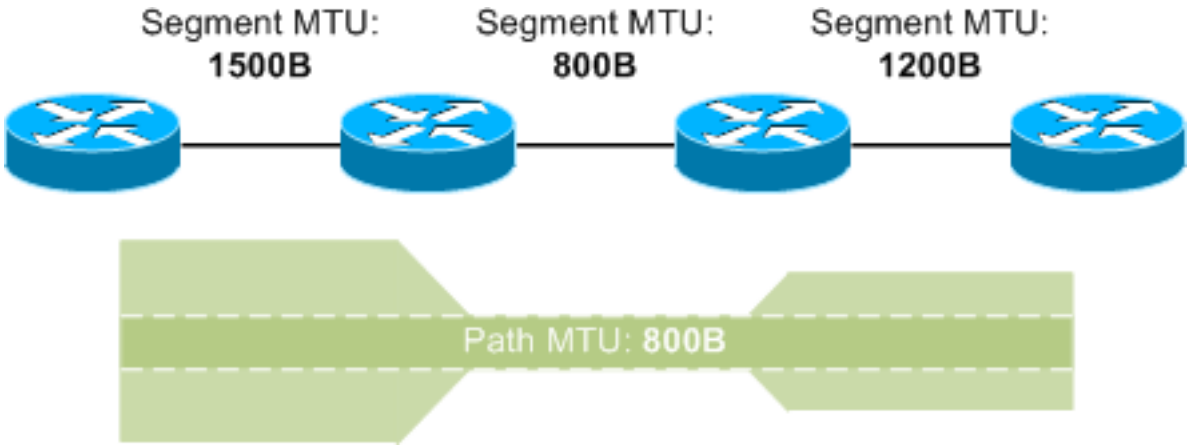



C:\WINDOWS\system32\cmd.exe - nslookup
C:\Documents and Settings\TUD>nslookup
Default Server: ns1.tudelft.nl
Address: 130.161.180.1
> ?
Commands: (identifiers are shown in uppercase, [] means optional)
NAME - print info about the host/domain NAME using default server
NAME1 NAME2 - as above, but use NAME2 as server
help or ? - print info on common commands
set OPTION - set an option
all - print options, current server and host
[no]debug - print debugging information
[no]d2 - print exhaustive debugging information
[no]defname - append domain name to each query
[no]recurse - ask for recursive answer to query
[no]search - use domain search list
[no]vc - always use a virtual circuit
domain=NAME - set default domain name to NAME
srchlist=N1[/N2/.../N6] - set domain to N1 and search list to N1,N2, etc.
root=NAME - set root server to NAME
retry=X - set number of retries to X
timeout=X - set initial time-out interval to X seconds
type=X - set query type (ex. A,ANY,CNAME,MX,NS,PTR,SOA,SRV)
querytype=X - same as type
class=X - set query class (ex. IN (Internet), ANY)
[no]msxfr - use MS fast zone transfer
ixfrver=X - current version to use in IXFR transfer request
server NAME - set default server to NAME, using current default server
lserver NAME - set default server to NAME, using initial server
finger [USER] - finger the optional NAME at the current default host
root - set current default server to the root
ls [opt] DOMAIN [> FILE] - list addresses in DOMAIN (optional: output to FILE)
-a - list canonical names and aliases
-d - list all records
-t TYPE - list records of the given type (e.g. A,CNAME,MX,NS,PTR etc.)
view FILE - sort an 'ls' output file and view it with pg
exit - exit the program
>

Maximum Transmission Unit

- Not all links support the same fragment size

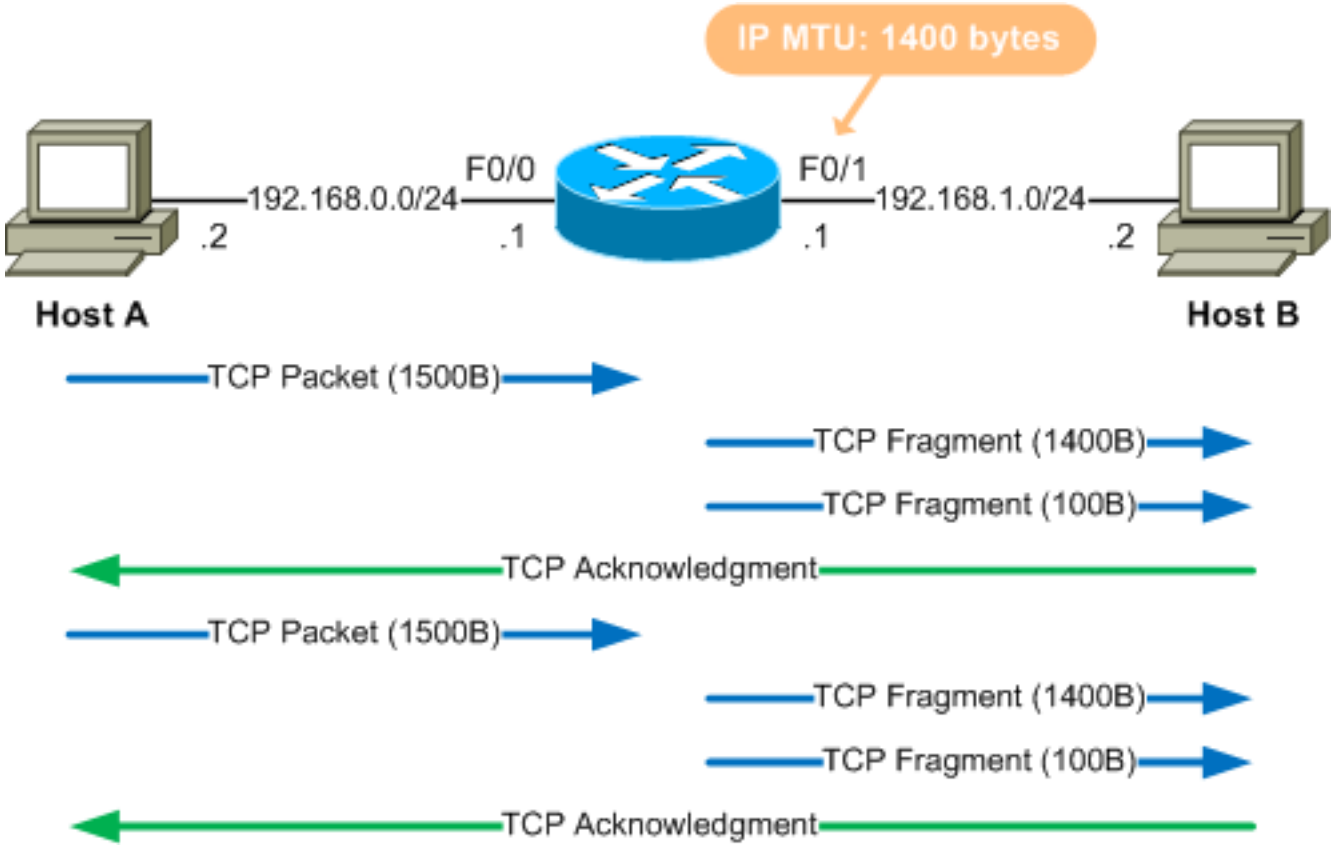
- Connectionless and layered nature prevents learning



[http://packetlife.net/blog/2008/aug/18/path-mtu-discovery/]

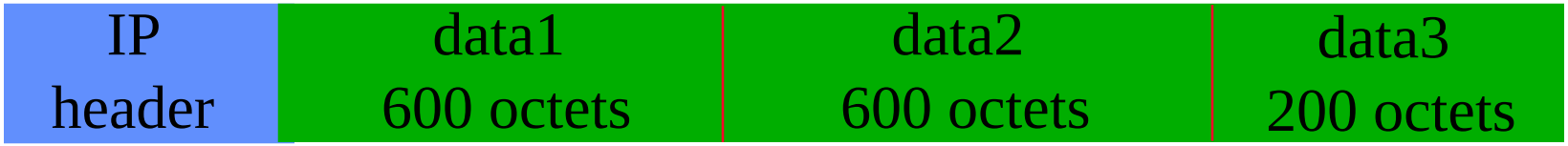
Fragmentation

- Instead, IPv4 fragments packets



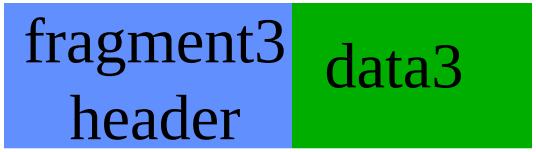
[http://packetlife.net/blog/2008/aug/18/path-mtu-discovery/]

Fragmentation



Fragment1: offset 0

Fragment2: offset 600



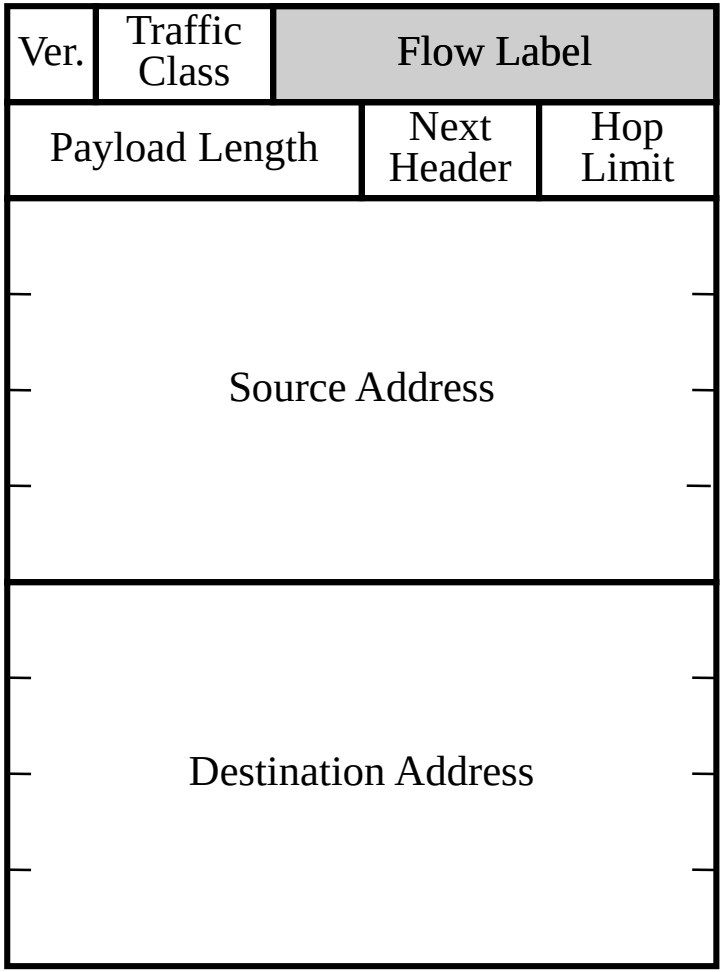
Fragment3: offset 1200

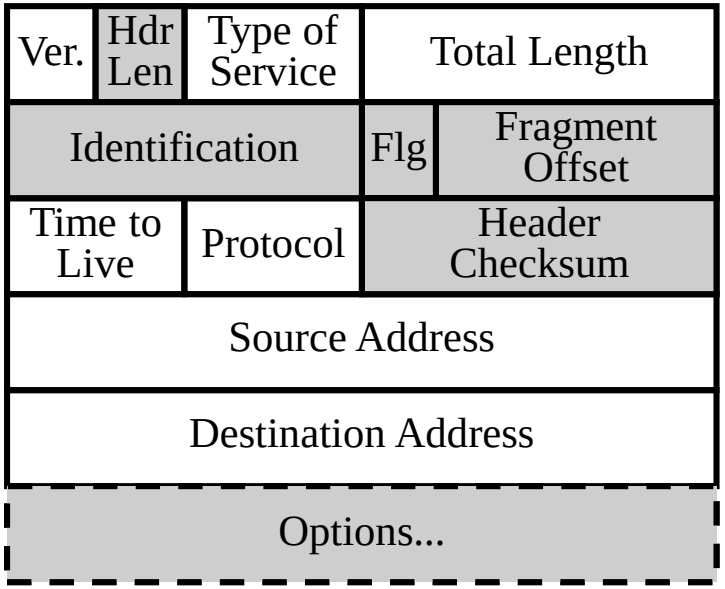
IPv6

- IPv6 is not a simple derivative of IPv4, but a definite improvement
- Four major changes over IPv4:
  1. *extended addressing capability*:
     128 bits versus 32 bits in IPv4
  2. *a fixed format to all headers*
     no option element, no header length field, but extension headers
  3. *no header checksum*
     diminish cost of processing, other layers check & correct for errors
  4. *no hop-by-hop fragmentation*
     unit of transmission = unit of control: use *path MTU discovery*
  5. *no Network Address Translation*

IPv6 Header Format

Header comparison IPv6-IPv4

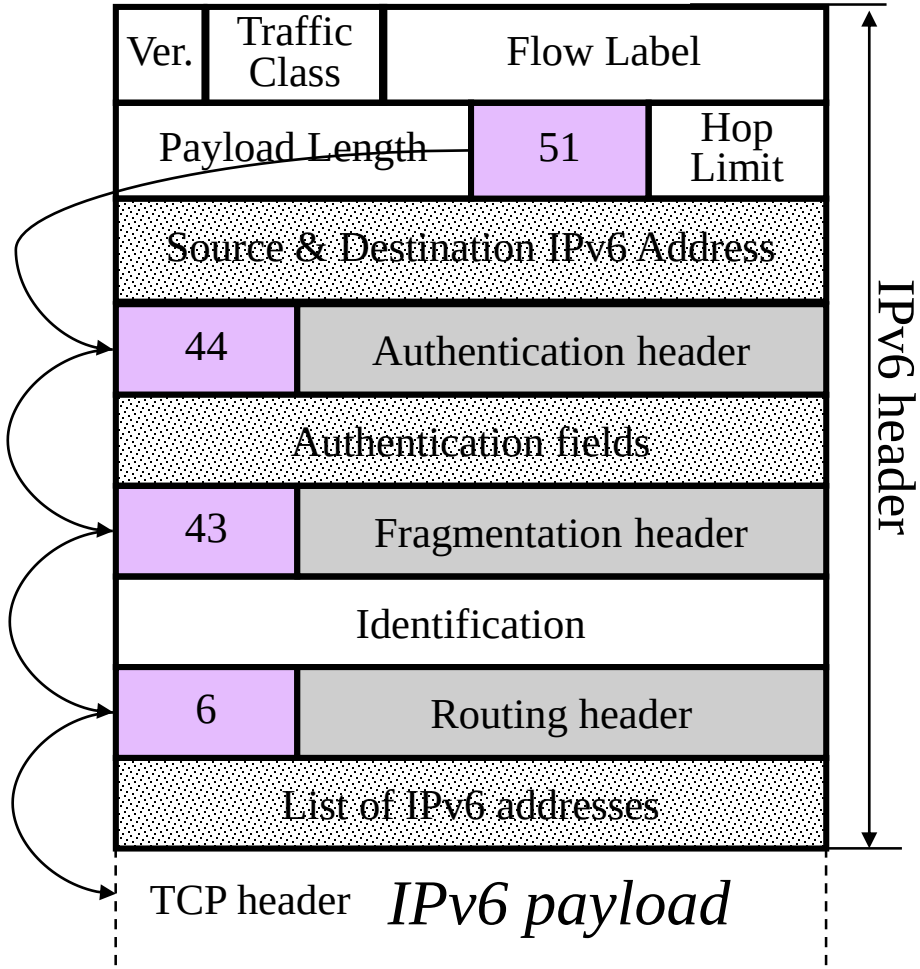




shaded fields have no equivalent in the other version

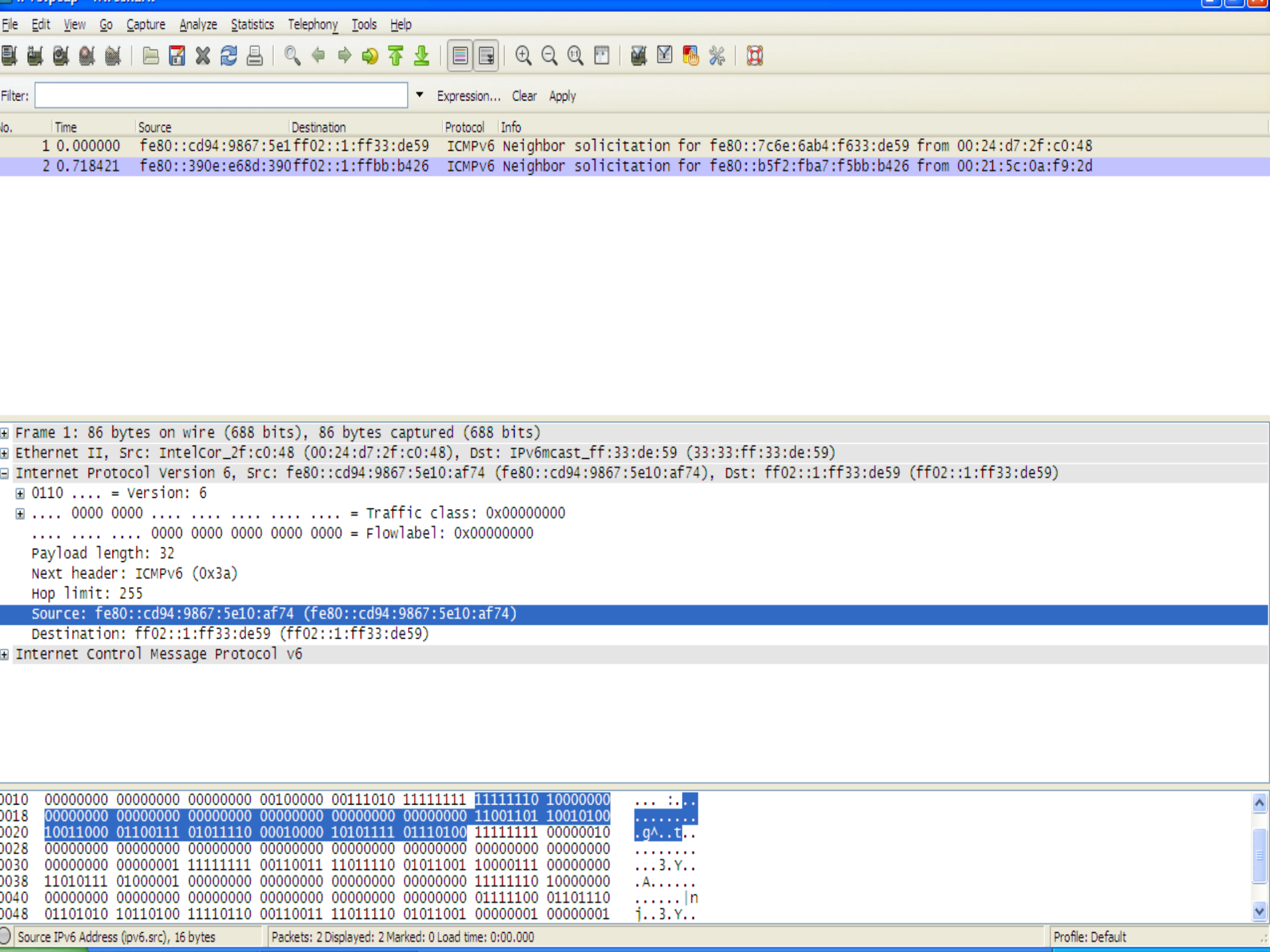
IPv6 header is twice as long (40 bytes) as IPv4 header without options (20 bytes)

Extension Header



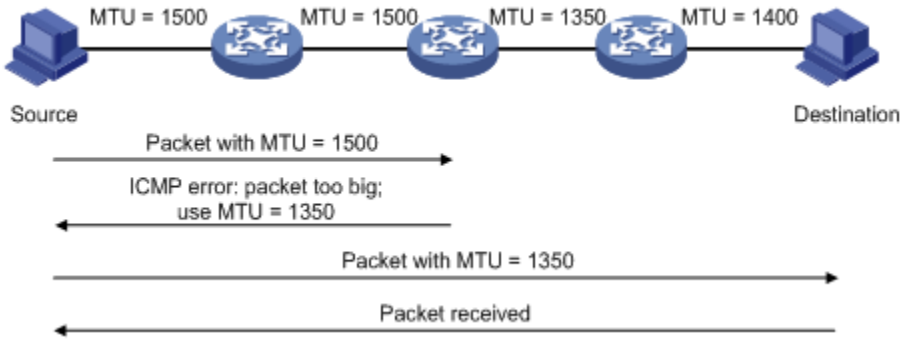
IPv6 Addressing

- Text Representations:
  - x_1:x_2:x_3:x_4:x_5:x_6:x_7:x_8 where x_k hexadecimal value of 16-bit. e.g. 1080:0:0:0:0:0:200C:417A
  - "zero" compressed notation: e.g.1080::200C:417A
  - Mixed IPv4 and IPv6: x:x:x:x:x:x:d.d.d.d where *d* decimal value of 8-bit IPv4 address
    e.g. 0:0:0:0:0:0:13.1.68.3 = ::13.1.68.3
- Three types:
  - unicast, anycast and multicast (no broadcast)
  - multicast: FF:x_2:x_3:x_4:x_5:x_6:x_7:x_8 (1/256 of address space)
  - All types are assigned to interfaces, not nodes


File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 fe80::cd94:9867:5e1 ff02::1:ff33:de59 ICMPv6 Neighbor solicitation for fe80::7c6e:6ab4:f633:de59 from 00:24:d7:2f:c0:48
2 0.718421 fe80::390e:e68d:390 ff02::1:ffbb:b426 ICMPv6 Neighbor solicitation for fe80::b5f2:fba7:f5bb:b426 from 00:21:5c:0a:f9:2d
Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
Ethernet II, Src: IntelCor_2f:c0:48 (00:24:d7:2f:c0:48), Dst: IPv6mcast_ff:33:de:59 (33:33:ff:33:de:59)
Internet Protocol Version 6, Src: fe80::cd94:9867:5e10:af74 (fe80::cd94:9867:5e10:af74), Dst: ff02::1:ff33:de59 (ff02::1:ff33:de59)
0110 .... = Version: 6
.... 0000 0000 .... = Traffic class: 0x00000000
.... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
Payload length: 32
Next header: ICMPv6 (0x3a)
Hop limit: 255
Source: fe80::cd94:9867:5e10:af74 (fe80::cd94:9867:5e10:af74)
Destination: ff02::1:ff33:de59 (ff02::1:ff33:de59)
Internet Control Message Protocol v6
0010 00000000 00000000 00000000 00100000 00111010 11111111 11111110 10000000 ... :...
0018 00000000 00000000 00000000 00000000 00000000 00000000 11001101 10010100 ....
0020 10011000 01100111 01011110 00010000 10101111 01110100 11111111 00000010 .g^..t..
0028 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 ....
0030 00000000 00000001 11111111 00110011 11011110 01011001 10000111 00000000 ...3.Y..
0038 11010111 01000001 00000000 00000000 00000000 00000000 11111110 10000000 .A......
0040 00000000 00000000 00000000 00000000 00000000 00000000 01111100 01101110 ......|n
0048 01101010 10110100 11110110 00110011 11011110 01011001 00000001 00000001 j..3.Y..
Source IPv6 Address (ipv6.src), 16 bytes
Packets: 2 Displayed: 2 Marked: 0 Load time: 0:00.000
Profile: Default

Path MTU Discovery

- IPv6 does not fragment like IPv4
- Relies on ICMP to find MTU
- Creates blackholes when blocked



[http://bitsandoctets.blogspot.nl/2012/05/mtu-and-mss-why-should-we-care.html]

ICMP

- Internet control message protocol (ICMP)
  - two modes: query (echo, timestamp, etc...) and error
  - Diagnosing error conditions sent back to the source when a router destroys a packet
  - Not to enhance reliability, but merely to provide feedback about network problems.

Ping

- Ping (via ICMP *query* mode ‘echo’)
- Ping –f –l <size in bytes> address
  - f: sets don’t fragment flag
  - l: sets payload size

Traceroute

- Traceroute (via ICMP *error* mode)
- Windows: tracert address
- Returns the path to the address
- Sends multiple UDP packets with increasing TTL
- Artefacts:
  - ICMP messages may be discarded
  - UDP packets might traverse multiple paths
  - Interface and no router addresses

Questions Ch. 4

- Give, in x.y.z.w. notation, the subnet mask that belongs to the address 130.140.150.160/20.
- Give three differences between the headers of IPv4 and IPv6 and explain the motivation behind these differences.

Break exercise

- Use *ping* to find RTTs,
- And *tracert* to find the paths to:

1. Your gateway router
2. Common destinations, such as
    1. Google website
    2. TU Delft website
    3. Major Dutch news site
    4. Major foreign (transatlantic) news site
3. Each other's laptop