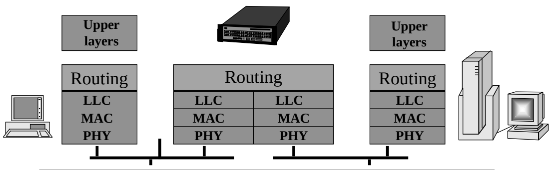
# Purpose of the Network Layer

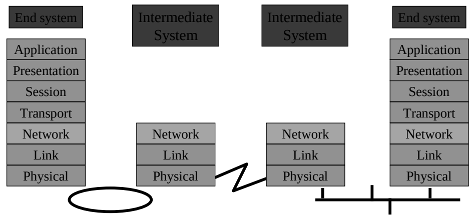
- ❑ Intercconnect systems on all possible cable plants
  - LANs, modems, X.25 links
- ❑ Principle
  - every end system is allocated a network layer (=layer 3) address
  - subnetwork = collection of systems at layer 2
  - intermediate systems forward packets
    - ➪ based on layer 3 address
    - ➪ between subnetworks






# Different Network Layer Exist

- ❑ Different networking architetures exist
  - TCP/IP, NetBIOS, SNA, DECNET, XNS,Apple Talk
- ❑ The OSI model is used as a referencce model for describing all architectures
  - but OSI architecture is not widespeard in real systems






# History of Internet

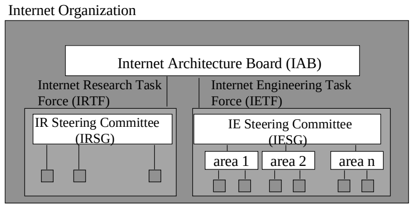
- ❑ **1972** Demonstration of **ARPANET**
  - 50 sites, 20 routers; based on **CO** network layer (**IMP**) and **NCP** (transport)
  - initial work on **TCP**, then **TCP/IP**
- ❑ **1977-79** **TCP/IP** protocols in ARPANET
- ❑ **1980** **UNIX BSD 4.1** includes TCP/IP
- ❑ **1980** construction of an **Internet** using ARPANET as a backbone
- ❑ **1983** **TCP/IP** standard for ARPANET
- ❑ **1983** ARPANET split into **military** and **civil** networks
- ❑ **1987** worldwide **Internet**



# The Internet Organization

- ❑ Coordinates development of Internet standards = (TCP/IP standards)

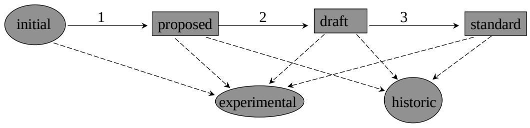
Internet Organization






# Internet Standards

- ❑ RFCs = official Internet documentation:
  - standards, other documents
  - maintained by INTERNIC, shadows at ftp. switch ch
  - INTERNIC manages IP addresses and domain names
  - IANA manages constant names (eg: port 53 for DNS)
- ❑ Internet standardization process:




1:stable specification accepted by IETF
2:two impiementations tested, 6 months
3:field experience for 4 months



# IP Principles

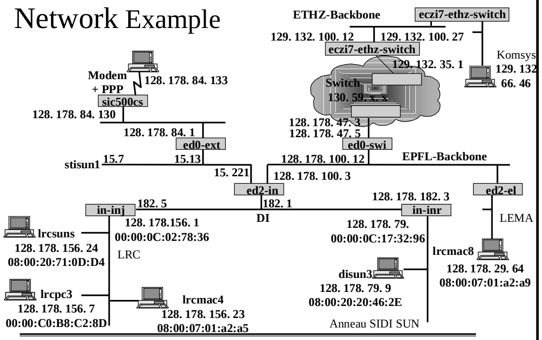
Homogeneous addressing

- ❑ an IP address is unique accross the whole network
- ❑ IP address is the address of the interface
- ❑ one adderss per interface
- ❑ communication between IP hosts require knowledge of IP addresses

Networking by routers

- ❑ Within one network (eg.: Ethernet): direct communication (but using IP addresses)
- ❑ Routers between subnetworks
- ❑ Terminology:
  - host = end system; router = intermediate system
  - subnetwork = one collection of hosts that can communicate directly without routers

# Network Example






# IP Addresses

| | 0 1 2 3 ... 8 | 16 | 24 | 31 |
|---|---|---|---|---|
| class A | 0 | Net Id | Host Id | |
| class B | 10 | Net Id | Host Id | |
| class C | 110 | Net Id | Host Id | |
| class D | 1110 | Multicast address | | |
| class E | 11110 | Reserved address | | |

Examples: 129.132.x.x = ETHZ host; 18.x.x.x = MIT host
193.40.252.x = TTU host; 8.x.x.x = IBM host

| Class | Nr of NW bits | First usable value | Last usable value | Max Nr of networks | Max Nr of hosts on a network |
|---|---|---|---|---|---|
| A | 7 | 1 | 126 | 126 | 16 777 214 |
| B | 14 | 128.1 | 191.254 | 16 382 | 65 534 |
| C | 21 | 192.0.1 | 223.255.254 | 2 097 150 | 254 |
| D | - | 224.0.0.0 | 239.255.255.254 | | |
| E | - | 240.0.0.0 | 255.255.255.254 | | |

- IPv4: 2 100 000 networks; 3 720 000 000 hosts ?



# IP Addresses (2)

- One additional hierarchy level was added (subnetting)

| | 0 1 2 3 ... 8 | 16 | 24 | 31 |
|---|---|---|---|---|
| class A | 0 | Net Id | Subnet Id | Host Id |
| class B | 10 | Net Id | Subnet Id | Host Id |
| class C | 110 | Net Id | Sn Id | Host Id |
| class D | 1110 | Multicast address | | |
| class E | 11110 | Reserved address | | |

Subnet Id - any size, configured by local administrator
Net Id - allocated by INTERNIC, unique world-wide
Host Id - allocated by local administrator
Subnetting used mainly with class A and B addresses



# Subnet Mask

- Exercise: what is **netId** for address 9. 12. 189. 35 ?
- Exercise: what is net and **subnetId** for 128. 178. 156. 24 ?
- Subnet Mask:
  - IP@ string with **hostId** set to **0** and **net:subnet** part set to **1**
  - net:subnet = IP@ **bitwise and** subnetMask

| 10 | Net Id | Subnet Id | Host Id |
|---|---|---|---|

16+ n bits Net:Subnet part

- subnet mask at **TTU=255.255.255.0**
- IP@ requires subnet mask to determine complete ( net:subnet) structure
- Three level hierarchy : net, subnet. host
  - outside of net, no need for knowledge of subnet details
- IP address changes when host moves to another subnet (ex: Ethernet split into 2); compare to bridging



# Reserved IP Addresses

- **127.X.X.X** **- local software loopback test**
- **Network = all 0's** **- 'this network'**
- **Network = all 1's**
- **Host = all 0's** **- particular network**
- **Host = all 1's** **- broadcast to particular network**
- **0.0.0.0** **- as source A - don't know my Addr**
  **- as dest. in router table -DAFAULT**
- **255.255.255.255** **- limited broadcast on subnetwork (not forwarded by routers)**



# IP Unicast Packet Forwarding

- **Case 1: a host route exists for *destAddr* :**
  **for every** entry in routing table (**RT**)
  **if** (destnationaddr = ***destAddr***)
  **then** send to nextHop IPaddr; leave
- **Case 2: *destAddr* is on a directly connected network (= on-link)**
  **for every** physical interface IP address (**PI**) and subnet mask (**SM**)
  **if** (**PI** && **SM** = ***destAddr*** && **SM**)
  **then** send directly to ***destAddr*** ; leave
- **Case 3: a network route exists for *destAddr***
  **for every** entry in routing table
  **if** (destinationaddr && **SM** = ***destAddr*** && **SM**)
  **then** send to ***nextHop*** IP address; leave
- **Case 4: use default route**
  **for every** entry in routing table
  **if** (destinationaddr = **DEFAULT**)
  **then** send to ***nextHop*** IP address; leave

# The IP Routing Algorithm

- ❑ Same at hosts and routers
- ❑ uses the **IP Routing Table**
  - contains information about directly attached systems + next hop routing information
- ❑ Examples:

| At lrcsuns: Next Hop Table | | | Physical Interface Tables | |
|---|---|---|---|---|
| destination@ | subnetMask | nextHop | IP | subnetMask |
| DEFAULT | | 128. 178. 156. 1 | 128. 178. 156. 24 | 255. 255. 255. 0 |

| At in-inj: Next Hop Table | | | Physical Interface Tables | |
|---|---|---|---|---|
| destination@ | subnetMask | nextHop | IP | subnetMask |
| 128.178.156.0<br>DEFAULT | 255. 255. 255. 0 | 128. 178. 182. 5<br>128. 178. 182. 1 | 128. 178. 79. 1<br>128. 178. 182. 3 | 255. 255. 255. 0 |



# Example

- ❑ Fill in the table if an IP packet has to be sent from lrcsuns

| final destination | next hop | case number |
|---|---|---|
| 128. 178. 79. 9<br>128. 178. 156. 23<br>127. 0. 0. 1<br>128. 178. 84. 133<br>129. 132. 1. 45 | | |

- ❑ Fill in the table if an IP packet has to be sent from in-inj



# Example with Netstat

```
Lrcsuns:>netstat -nr

Routing table:
  Destination              Gateway              Flags   Ref   Use     Interface
  127.    0.    0. 1       127.    0.    0.  1   UH      0     11239   lo0
  128. 178. 156. 0         128. 178. 156. 24     U       3     38896   le0
  224.    0.    0. 0       128. 178. 156. 24     U       3         0   le0
  default                  128. 178. 156.  1     UG      0     85883
```



# ARP Packet on Ethernet

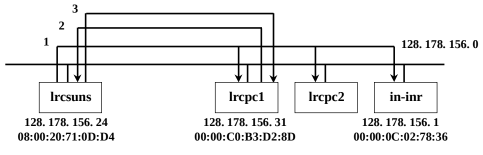
| | 2 | 2 | 1 | 1 | 2 | 6 | 4 | 6 | 4 |
|---|---|---|---|---|---|---|---|---|---|
| Ethernet Header | Hw type | prot type | HLEN | PLEN | Op type | Sender hw @ | Sender IP @ | Target hw @ | Target IP @ |

Hw type = link layer type = 1 for Ethernet
Prot type = 0800 = IP (here)
HLEN = length of "hardware" address
PLEN = length of "protocol" address ( = IP address size)
Op Type = 1: ARP request; 2; ARP reply ;

- ❑ ARP protocol type = 0806
  IP protocol type = 0800
- ❑ ARP addresses learnt from ARP-Requests
- ❑ message chart diagram with MAC and ARP fields for previous page?
- ❑ is an ARP packet an IP packet?
- ❑ ARP requests may be sent at most once per sec for one given target

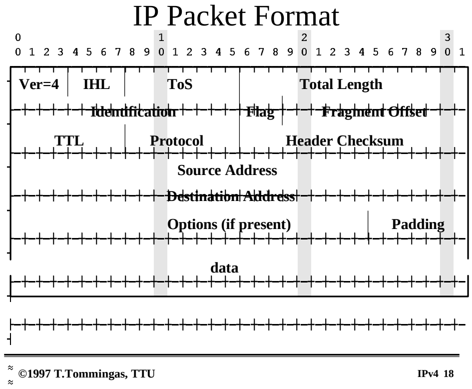


# ARP Protocol (Ethernet, FDDI) (1)






# IP Packet Format



# IPv4 datagram

- ❑ **Version** - *4bits*
  - current version 4
- ❑ **Internet Header Length** (**IHL**) - *4 bits*
  - length of the header in 32-bit words (5 if no options are used)
- ❑ **Type of Service (ToS)** - *8 bits*
  - first 3 bits - **precedence** (8 priorities) some routers ignore that!
  - D - flag    requests low delay  (e.g. interactive trafic with remote echo)
  - T - bit for high throughput
  - R - bit for high reliability
  - C - bit for low cost route (recent RFC)
- ❑ **Total Length** - *16 bits*
  - total length of the datagram (or **fragment**) in octets including the Header
- ❑ **Identification** - *16 bits*
  - an integer value to identify datagrams. All fragments of a datagram carry the same id)



# IPv4 datagram (cont.)

- ❑ **Flags** - *3 bits*   **NU;DF;MF**
  - **NU** - not used
  - **DF - 'Do Not Fragment'**
  - **MF - 'More Flag'** 0 - last fragment of a datagram
- ❑ **Fragment Offset** - *13 bits*
  - indicates the position of the first octet in this fragment in the original message
  - measured in **8-octet** units
- ❑ **Time To Live ( TTL)** - *8 bits*
  - decremented by routers
  - datagramgets discarded when TTL drops to 0
- ❑ **Protocol** - *8 bits*
  - 17 UDP
  - 6 TCP
  - 1 ICMP
  - 8 EGP
  - 89 OSPF



# IPv4 datagram (cont.)

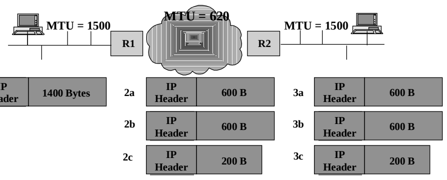
- ❑ **Header Checksum** - *16 bits*
  - protects only the **Header**!
  - recalculated every time the DG passes through a router
  - SUM(1's complement of w16i )
  - when operating over error-detected networks - all 0's
- ❑ **Data**- *variable*
- ❑ **Padding** - *variable*
- ❑ **Options** - *variable (n*8)*
  - Copy (1 bit) - copy Option to all fragments
  - Option Class (2 bits): time stamp - 2, usually 0
  - Option Number (5 bits):
    - ➪ Security (2) (US Defense Intelligence Agency)
    - ➪ Time stamp (4) - each router passed puts the actual time stamp
    - ➪ Loose source route (3)
    - ➪ Record route (7)
    - ➪ Strict source route (9) - ICMP error message if not possible



# IP Fragmentation (1)

IP hosts or routers may have IP datagrams larger than MTU
- ❑ hosts: TCP avoids fragmentation, UDP may require it
- ❑ typical in routers: ex: sic500cs

Fragmentation is performed when IP datagram too large





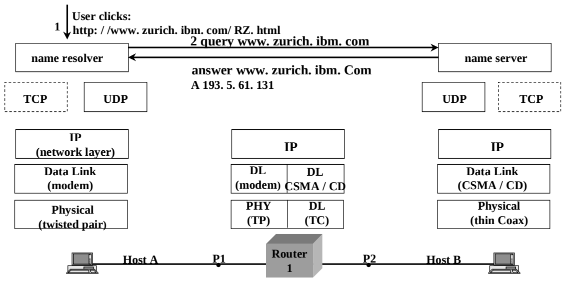

# MTU

- ❑ MTU - Maximum Transmission Unit

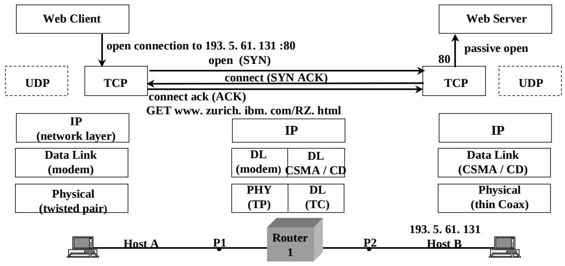
| Network | MTU (bytes) |
|---|---|
| Ethernet | 1500 |
| 802.3 with LLC/SNAP | 1492 |
| Token Ring 4 Mbps | 4464 |
| Token Ring 16 Mbps | 17914 |
| FDDI | 4352 |
| X.25 | 576 |
| Frame Relay | 1600 |
| ATM with AAL5 | 9180 |
| Hyperchannel | 65535 |
| PPP | 296 - 1500 |



# Example: name resolution

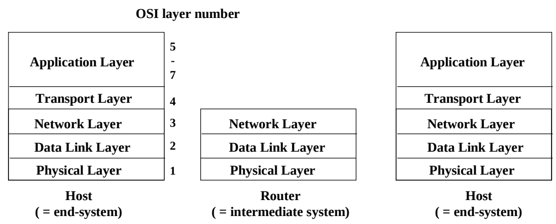


# Example: TPC connection opening



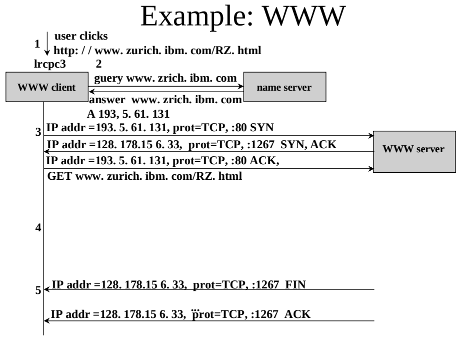

©1997 T.Tommingas, TTU IPv4 25

# The TPC/IP Architecture



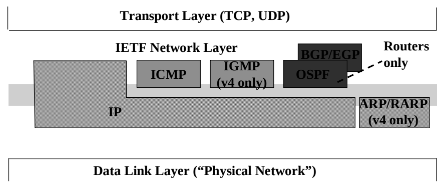

©1997 T.Tommingas, TTU IPv4 26

# Example: WWW




©1997 T.Tommingas, TTU IPv4 27

# IP Network Layer




©1997 T.Tommingas, TTU IPv4 28