TI III: Operating & Communication Systems

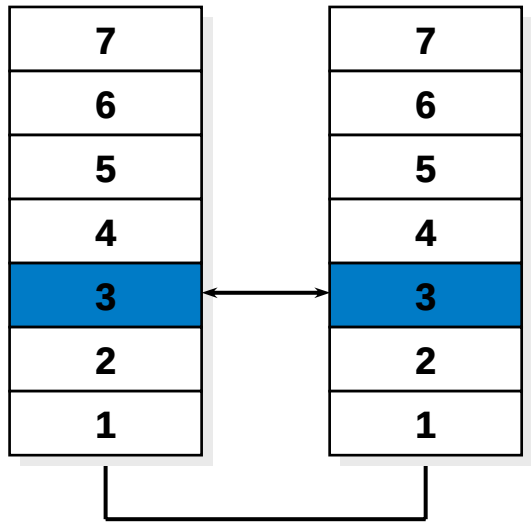
Internetworking

Switches, Routers

Routing

Internet Protocol

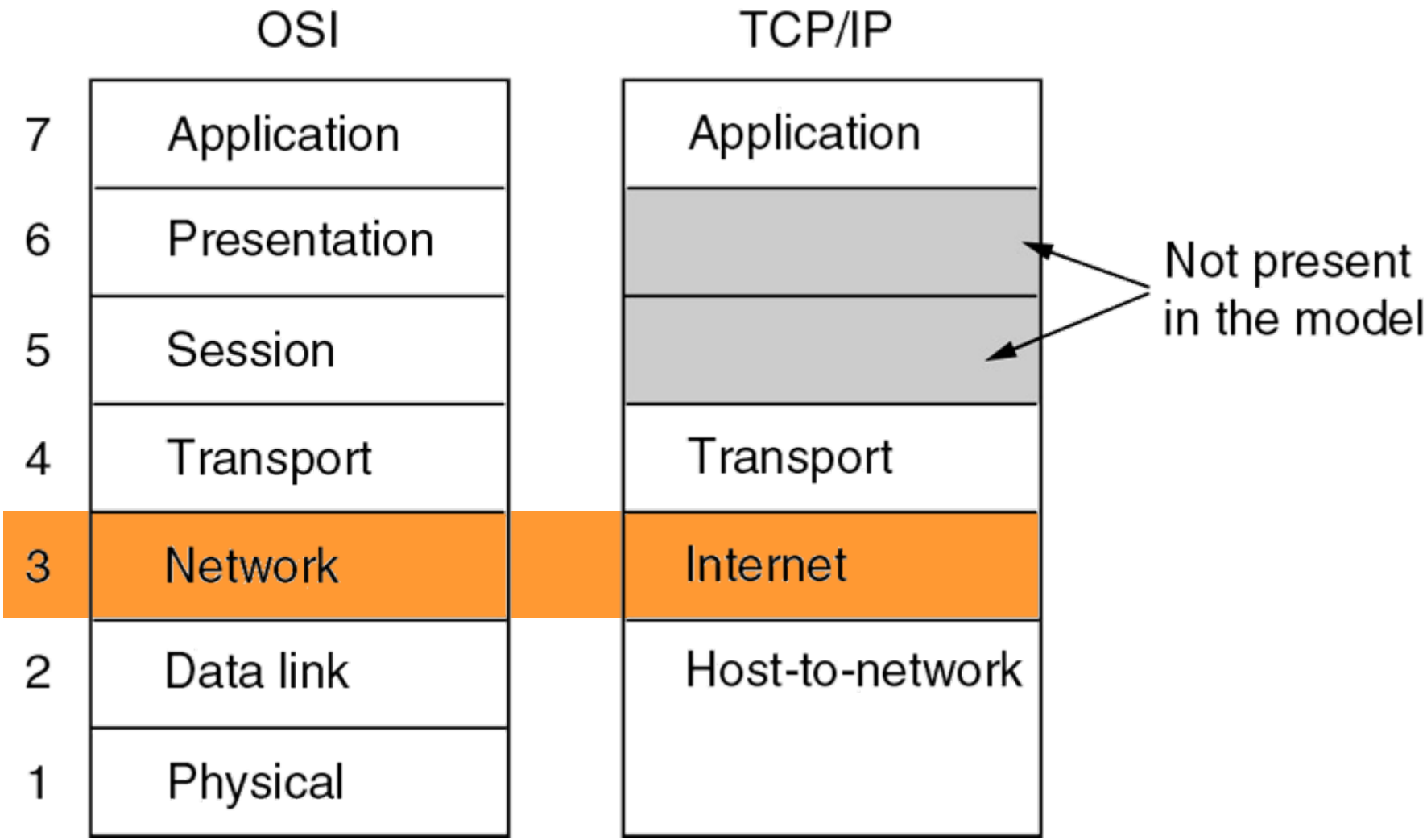
Addressing



Content (2)

8. Networked Computer & the Internet
   - Sockets
   - Internet
   - Layers, Protocols
9. Host-to-Network I
   - Physical Layer
   - Media, Signals
   - Modems
10. Host-to-Network II
    - Data Link Layer
    - Framing, flow control
    - Error detection/ correction
    - PPP
11. Host-to-Network III
    - Topologies
    - Medium Access
    - Local Area Networks
      - Ethernet, WLAN
12. **Internetworking**
    - **Switches, routers**
    - **Routing**
    - **Internet protocol**
    - **Addressing**
13. Transport Protocols
14. Application Support
15. Programming
16. Example

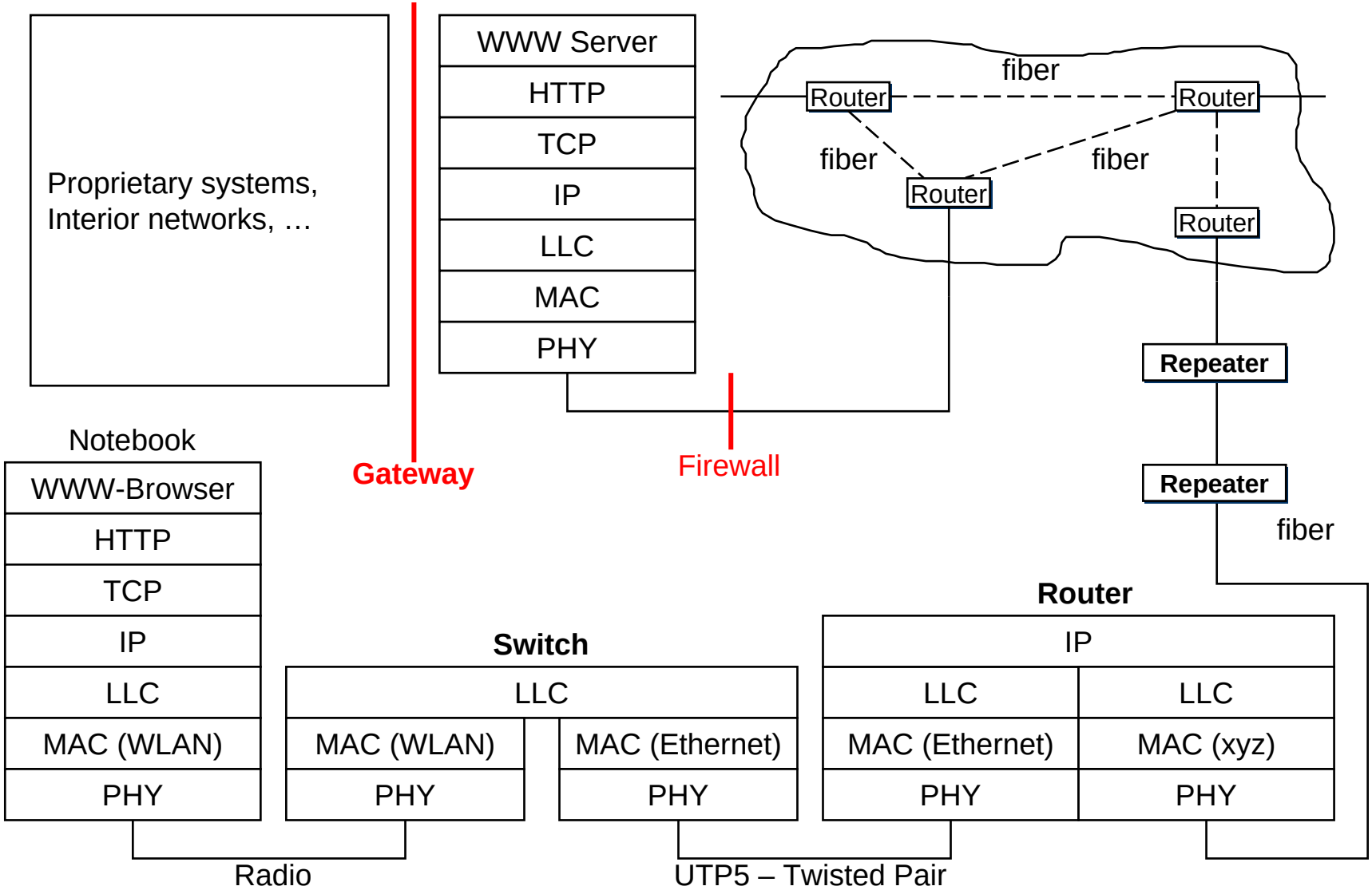
Data Link Layer



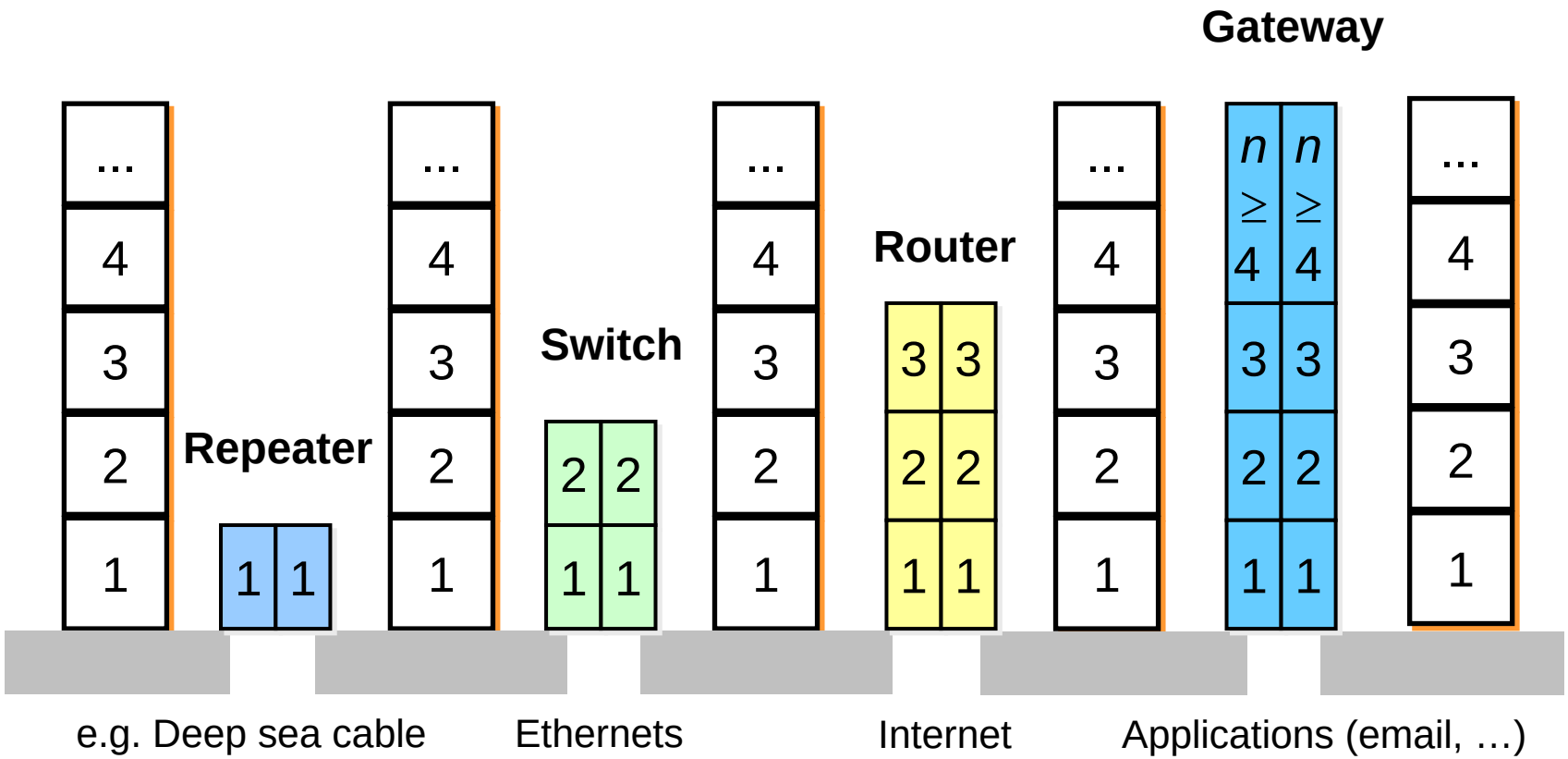
Reasons for multiple networks

- Limited number of users/throughput in a single network
- Historical reasons
  - Different groups started out individually setting up networks
  - Usually heterogeneous
- Geographic distribution of different groups over different buildings, campus, …
  - Impractical/impossible to use a single network because of distance
  - Long round-trip delay will negatively influence performance
- Reliability
  - Don’t put all your eggs into one basket
  - “Babbling idiot” problem (isolation of errors)
- Security
  - Promiscuous operation – contain possible damage
- Political reasons
  - Different authorities, policies, laws…

Internetworking units

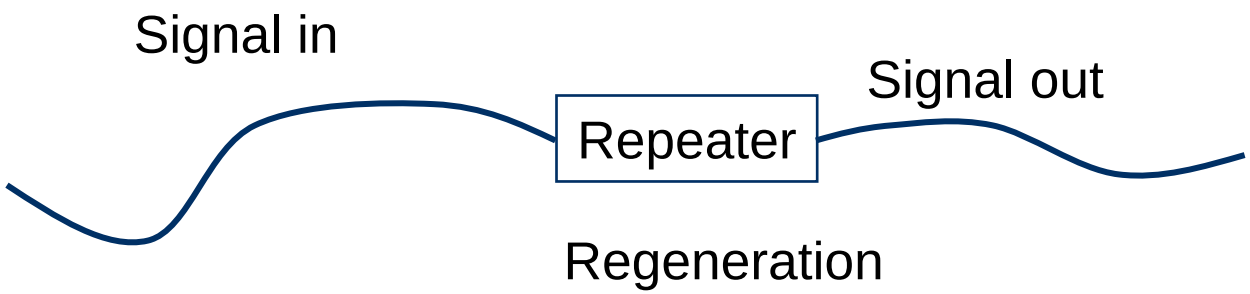


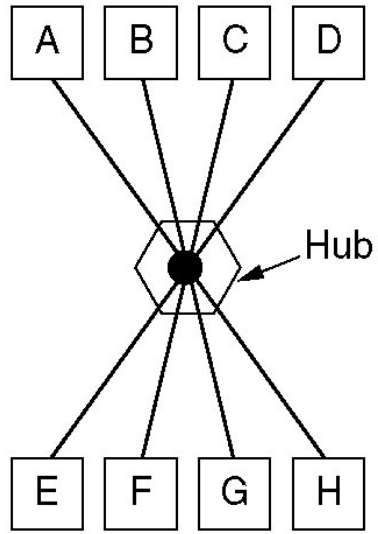
Internetworking units



Repeater/Hub

- Simplest option: Repeater
  - Physical layer device, connected to two or more cables
  - Amplifies signal arriving on either one, puts on the other cable through signal regeneration, combats attenuation
  - Signal has a "meaning" (represents bits) and, thus, can be regenerated, not only amplified (which would also amplify noise)
  - Neither understands nor cares about *content (bits)* of packets





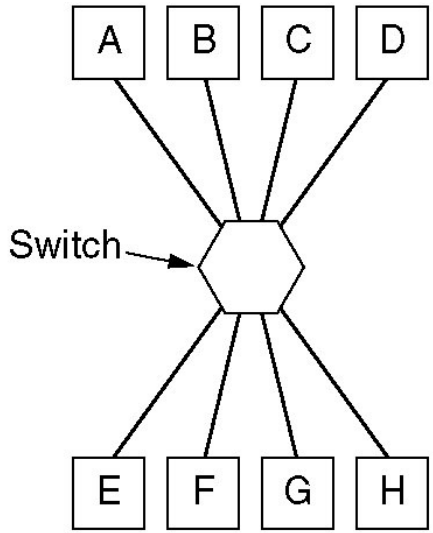
Physical layer solutions not satisfactory

- Physical layer devices – repeater, hub – do not solve the more interesting problems
  - E.g., how to handle load
- Some knowledge of the data link layer structure is necessary
  - To be able to inspect the content of the packets/frames and do something with that knowledge

- Link-layer solutions
  - Bridge & switch
  - Switch: Interconnect several terminals
  - Bridge: Interconnect several networks (of different type)
  - But terms sometimes used interchangeably (separation more historical)

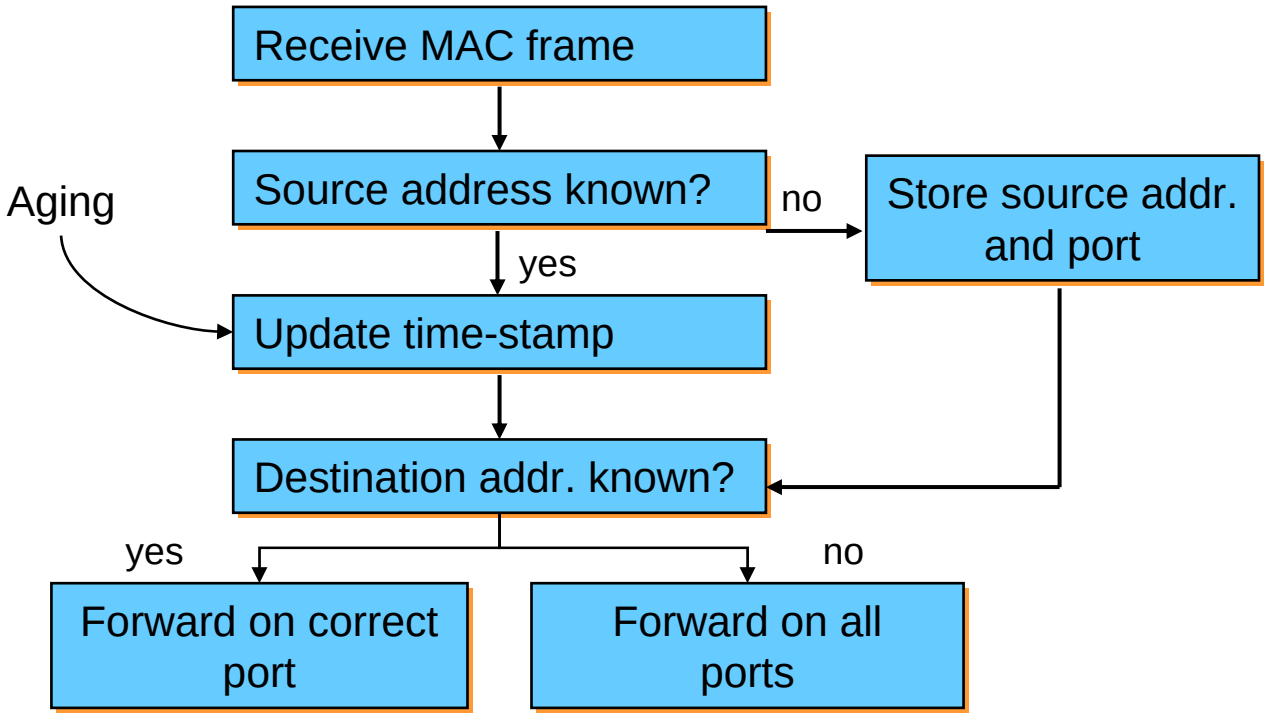
Switch

- Use a switch to connect several terminals or networks
- Switch inspects an arriving packet's MAC addresses and forwards it *only* on the right cable
  - Does not bother the other terminals
  - Needs: buffer, knowledge *where* which terminal is connected
- How to obtain knowledge about directions?
  - Simple: observe *from* where packets come to decide how to reach the sending terminal
  - ***Backward learning***



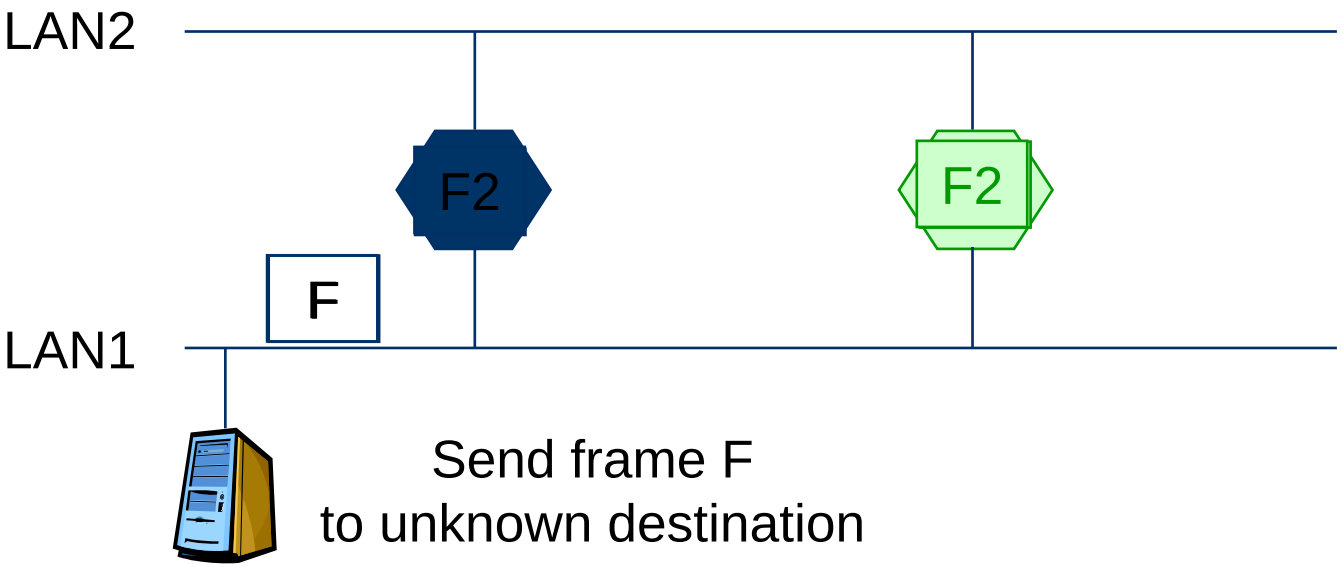
Backward learning – Algorithm

- How does a switch/bridge know initially where a node is?
  - It does NOT know!
    - Option 1: Manual configuration – not nice!
    - Option 2: Do not care – simply forward the data everywhere for an unknown address except to the network where it came from



Flooding by bridges – problems

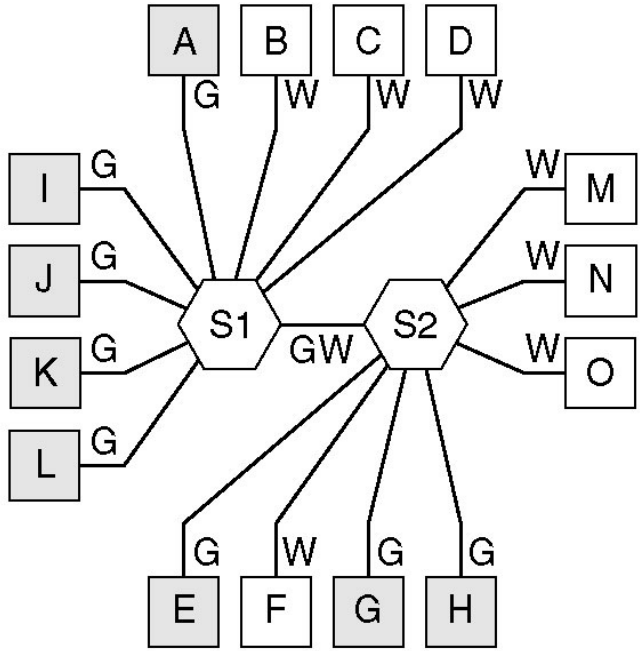
- Previous "backward learning by flooding" is simple, but problematic
- Consider example topology:
  - Second switch/bridge for reliability



- And so on … How to avoid such packet loops?
- Create a logical tree: ***Spanning Tree Protocol***

Further topic in LAN/LAN interconnection: VLAN

- Problem: LANs/switches are geared towards physical proximity of devices
- But: LANs should respect logical proximity
  - Connect devices of working groups together, irrespective where they happen to be located
- Idea: put a virtual LAN on top of an existing physical LAN
- Switches (or bridges) need configuration tables which port belongs to which VLAN
  - Only forward packets to ports of correct VLAN
- Membership of incoming packets determined by port, MAC address or IP address ➔ VLAN mapping
  - Standard: IEEE 802.1Q



Routers

- All devices so far either ignored addresses (repeaters, hubs) or worked on MAC-layer addresses (switches, bridges)

- For interconnection outside a single LAN/connection of LAN, these simple addresses are insufficient
  - Main issue: “flat”, unstructured addresses do not scale
  - A world-wide spanning tree does not really work …

- Need more sophisticated addressing structure and devices that operate on it
  - Routers and routing, e.g. based on IP addresses

Route from a notebook to NASA

Freie Universität Berlin

Traceroute gives current route a packets in the Internet

```
traceroute to www.NASA.GOV (128.183.243.3)

HOP NAME  (IP-address)                    TIME    probe 1 probe 2 probe 3
 0  mobile1.telematik.informatik.uni-karlsruhe.de (129.13.35.123)
 1  i70route1 (129.13.35.244)                      9 ms    9 ms   10 ms
 2  iracs1.ira.uka.de (129.13.1.1)                 2 ms    3 ms    2 ms
 3  Karlsruhe1.BelWue.DE (129.143.59.1)            2 ms    3 ms    2 ms
 4  Uni-Karlsruhe1.WiN-IP.DFN.DE (188.1.5.29)      3 ms    3 ms    3 ms
 5  ZR-Karlsruhe1.WiN-IP.DFN.DE (188.1.5.25)       5 ms    3 ms    2 ms
 6  ZR-Frankfurt1.WiN-IP.DFN.DE (188.1.144.37)     8 ms    7 ms    7 ms
 7  IR-Frankfurt1.WiN-IP.DFN.DE (188.1.144.97)     8 ms   11 ms    9 ms
 8  IR-Perryman1.WiN-IP.DFN.DE (188.1.144.86)    124 ms  126 ms  102 ms
 9  border3.Washington.mci.net (166.48.41.249)   121 ms  123 ms  124 ms
10  core4.Washington.mci.net (204.70.4.105)      123 ms  135 ms  121 ms
11  mae-east4.Washington.mci.net (204.70.1.18)   123 ms  122 ms  121 ms
12  mae-east.nsn.nasa.gov (192.41.177.125)       125 ms  126 ms  126 ms
13  rtr-wan1-ef.gsfc.nasa.gov (192.43.240.33)    127 ms  123 ms  121 ms
14  rtr-600.gsfc.nasa.gov (128.183.251.26)       147 ms  140 ms  130 ms
15  bolero.gsfc.nasa.gov (128.183.243.3)         127 ms  133 ms  129 ms
```

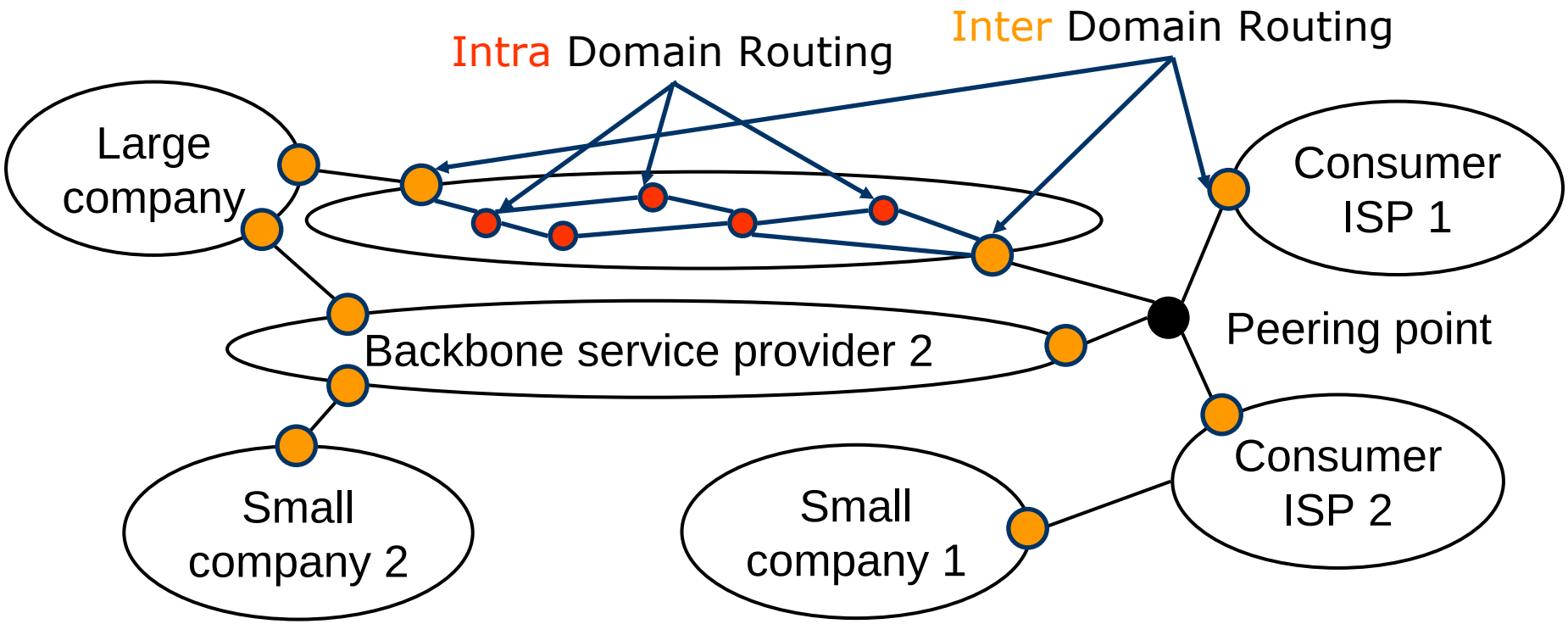
But it is not always that simple…

```
Z:\>tracert www.nasa.gov

Tracing route to www.nasa.gov.speedera.net [213.61.6.3]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  router-114.inf.fu-berlin.de [160.45.114.1]
  2    <1 ms    <1 ms    <1 ms  zedat.router.fu-berlin.de [160.45.252.181]
  3     1 ms    <1 ms    <1 ms  ice.spine.fu-berlin.de [130.133.98.2]
  4     1 ms    <1 ms    <1 ms  ar-fuberlin1.g-win.dfn.de [188.1.33.33]
  5     1 ms    <1 ms    <1 ms  cr-berlin1-po5-0.g-win.dfn.de [188.1.20.5]
  6     9 ms     9 ms     9 ms  cr-frankfurt1-po9-2.g-win.dfn.de [188.1.18.185]
  7    10 ms     9 ms     9 ms  ir-frankfurt2-po3-0.g-win.dfn.de [188.1.80.38]
  8    10 ms     9 ms     9 ms  DECIX.fe0-0-guy-smiley.FFM.router.COLT.net
                                    [80.81.192.61]
  9    10 ms     9 ms     9 ms  ir1.fra.de.colt.net [213.61.46.70]
 10    11 ms    10 ms     9 ms  ge2-2.ar06.fra.DE.COLT-ISC.NET [213.61.63.8]
 11    11 ms    10 ms    10 ms  213.61.4.141
 12    11 ms    10 ms    10 ms  h-213.61.6.3.host.de.colt.net [213.61.6.3]

Trace complete.
```

Not all addresses can be resolved to names (see DNS) and quite often the Internet redirects data in the context of Content Delivery Networks. Some nodes simply don't answer…

The idea of Internet routing

- Routing comprises
  - Updating of routing tables according to a routing algorithm
  - Exchange of routing information using a routing protocol
  - Forwarding of data based on routing tables/addresses



Autonomous systems in the IP world

- Large organizations can own multiple networks that are under a single administrative control
  - Forming an ***autonomous system*** or a ***routing domain***
- Autonomous system form yet another level of aggregating routing information
  - Gives raise to ***inter-*** and ***intra-domain routing***
- Inter-domain routing is hard
  - One organization might not be interested in carrying a competitor's traffic, …
  - Routing metrics of different domains cannot be compared; only *reachability* can be expressed
  - Scale – currently, inter-domain routers have to know about 150,000-200,000 networks

Intra-domain routing: OSPF

- Internet's most prevalent intra-domain (=interior gateway) routing protocol: ***Open Shortest Path First***
- Main properties
  - Open, variety of routing distances, dynamic algorithm
  - Routing based on traffic type (e.g., real-time traffic uses different paths)
  - Load balancing: also put some packets on the 2nd, 3rd best path
  - Hierarchical routing, some security in place, support tunneled routers
- Essential operation: Compute shortest paths on graph abstraction of an autonomous system
  - Link state algorithm

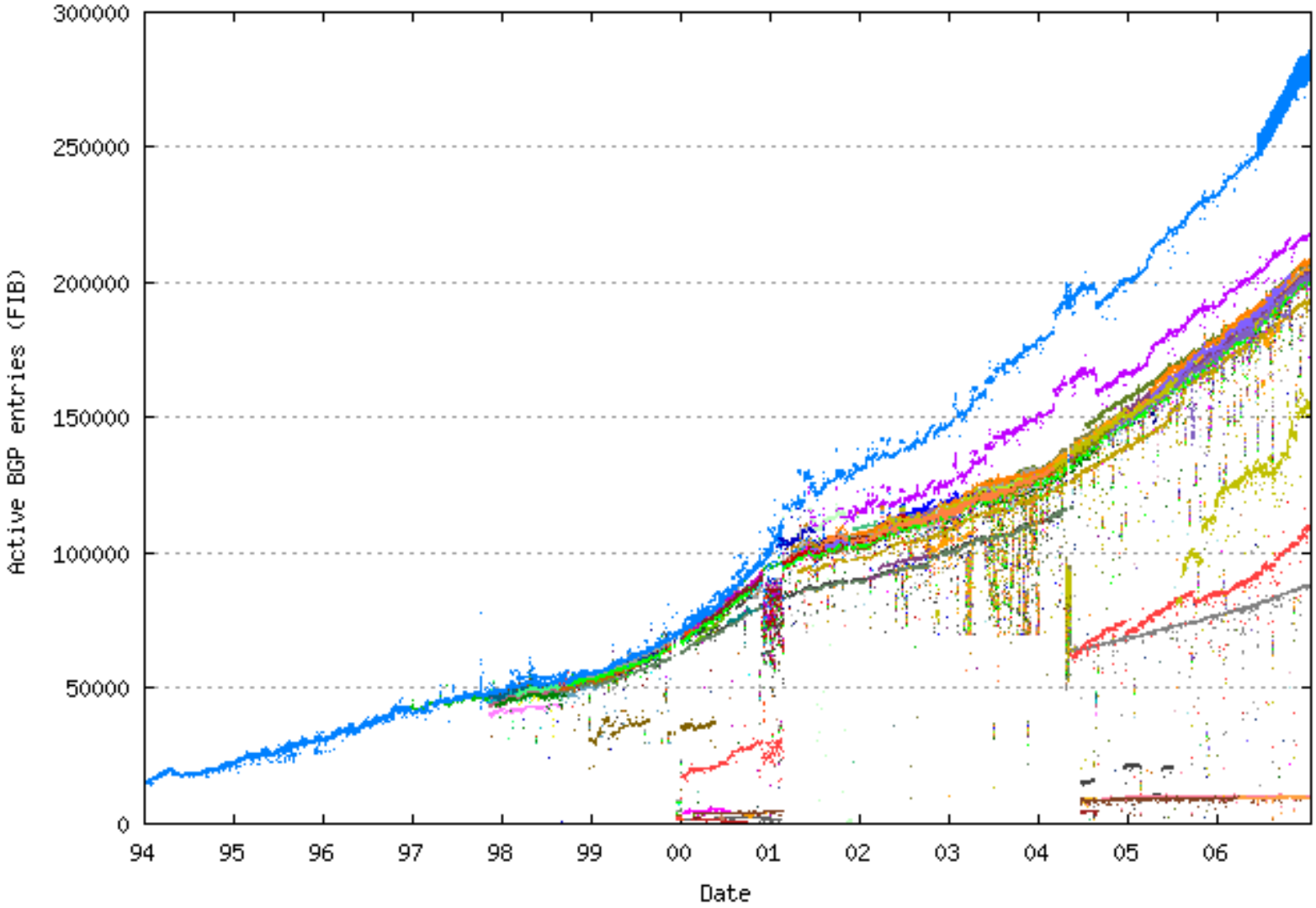
Basic ideas of Link State Routing

- Distributed, adaptive routing
- Algorithm
  - Discovery of new neighbors
    - HELLO packet
  - Measurement of delay / cost to all neighbors
    - ECHO packet measures rount trip time
  - Creation of link state packets containing all learned data
    - Sender, list of neighbors including delay, age
    - Periodic or event triggered (e.g. new neighbors, line failure etc.) update
  - Flooding of packet to all neighbors
    - Flooding, but with enhancements: duplicate removal, deletion of old packets etc.
  - Shortest path calculation to all other routers (e.g. Dijkstra)
    - Computing intensive, optimizations exist

Inter-domain routing: BGPv4

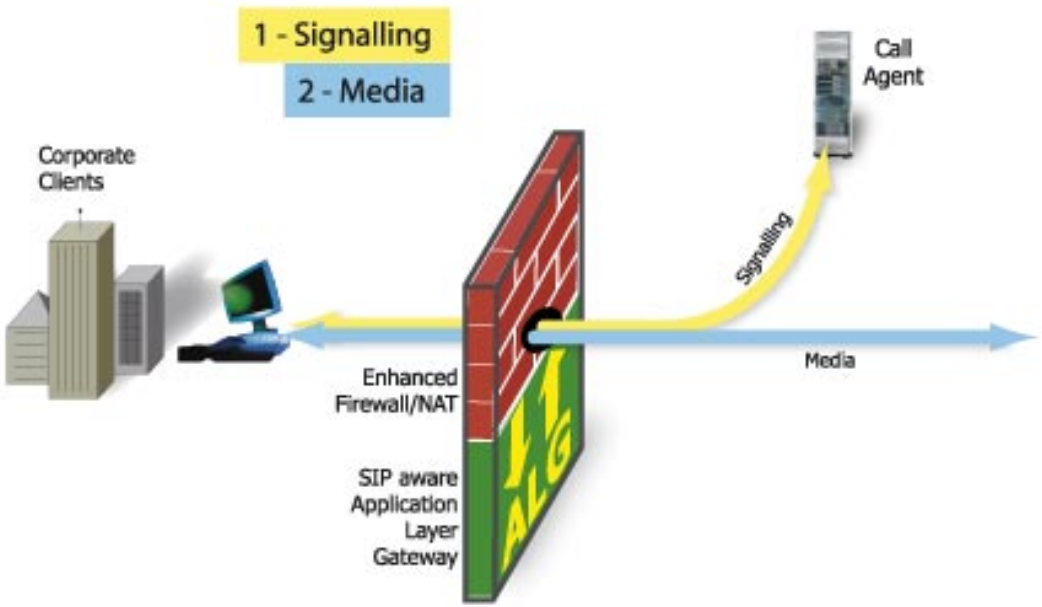
- Routing between domains: ***Border Gateway Protocols (BGP)***
  - Routing complicated by politics, e.g., only route packets for paying customers, …
  - Legal constraints, e.g.: Traffic originating and ending in Canada must not level Canada while in transit!
- BGP's perspective: only autonomous systems and their connections
- Operation: Distance vector protocol
  - But not only keep track of cost via a given neighbor, but store entire paths to destination ASs
  - More robust, solves problems like count to infinity
  - Infernally complicated protocol…

Example: BGP outing table sizes for Internet AS



Gateways

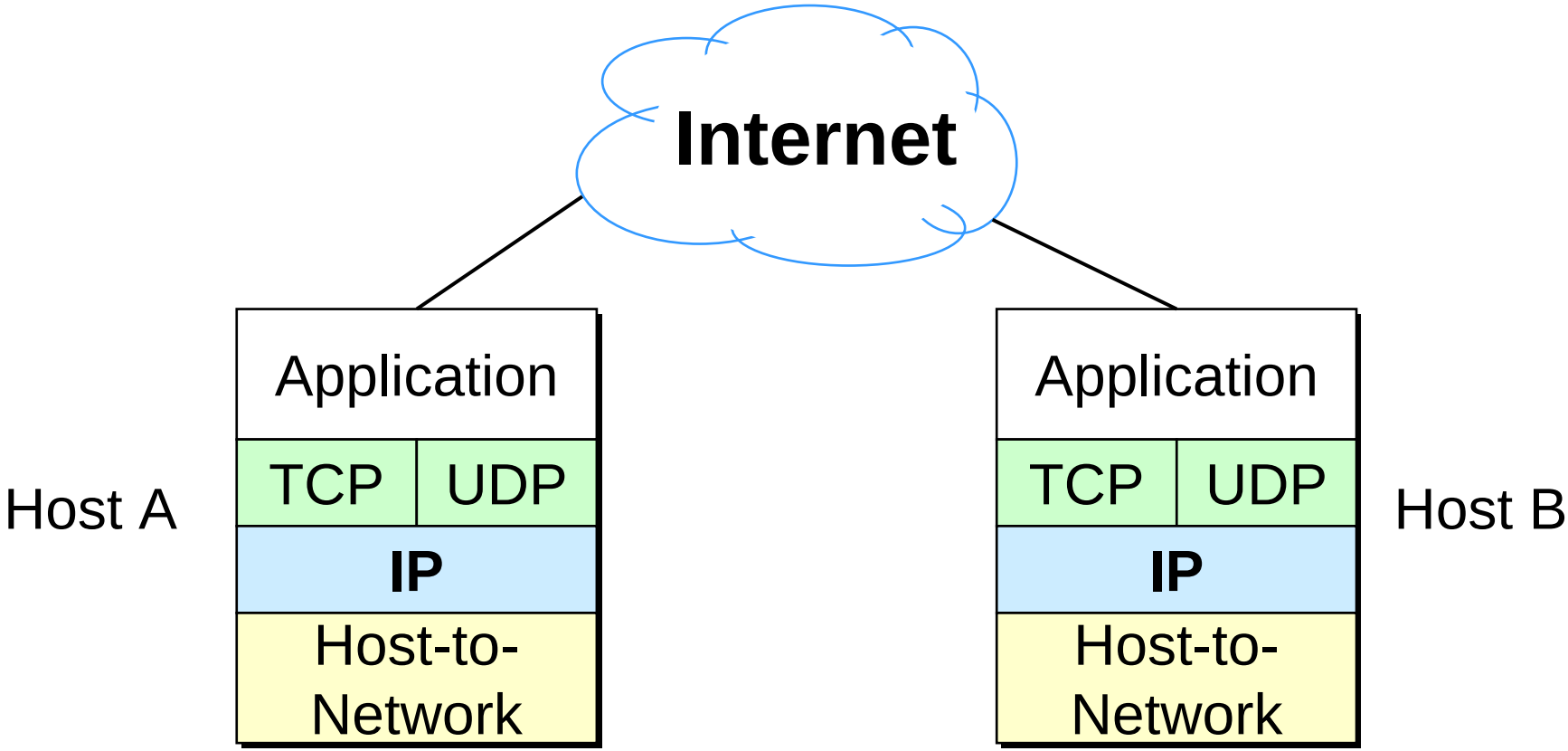
- If even routers will not do, higher-layer interconnection is necessary: ***Gateways***
  - Work at transport level and upwards
  - E.g., application gateways transforming between HTML and WML/HTTP and WAP
  - E.g., transcoding gateways for media content



Conclusion - Interconnections

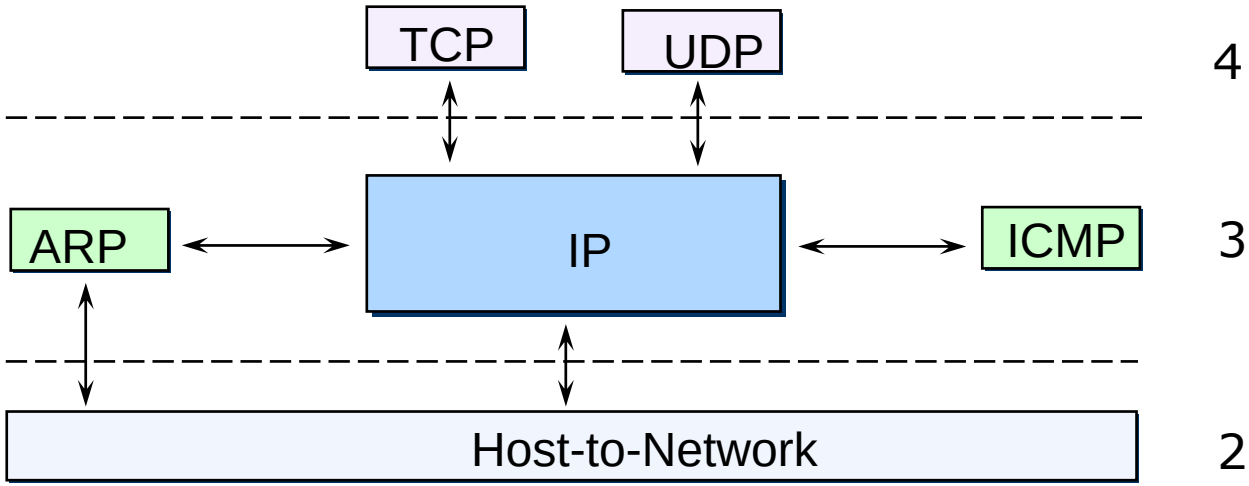
- Single LANs are insufficient to provide communication for all but the simplest installations
- Interconnection of LANs necessary
  - Interconnect on purely physical layer: Repeater, hub
  - Interconnect on data link layer: Bridges, switches
  - Interconnect on network layer: Router
  - Interconnect on higher layer: Gateway
- Problems
  - E.g., redundant bridges can cause traffic floods; need spanning tree algorithm
  - Simple addresses do not scale; need routers

Simplified view of Internet protocols



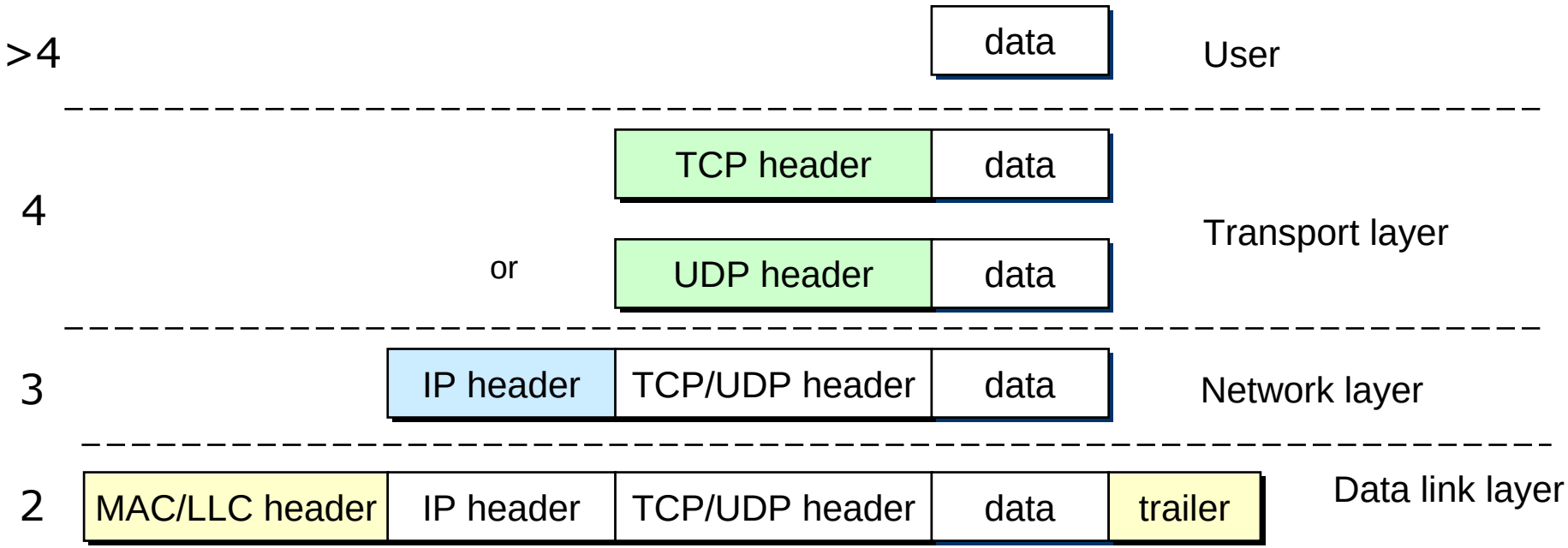
IP and supporting protocols

- Transport protocols (layer 4, TCP or UDP) hand over data together with the IP address of the receiver to IP
- IP may need to ask ARP for the MAC address (layer 2)
- IP hands over data together with the MAC address to layer 2
- IP forwards data to higher layers (TCP or UDP)
- ICMP (Internet Control Message Protocol) can signal problems during transmission



Data encapsulation/decapsulation

- IP forwards data packets through the network to the receiver
- TCP/UDP add ports (=> addresses of processes)
- TCP offers reliable data transmission
- Packets (PDU, protocol data unit) are encapsulated



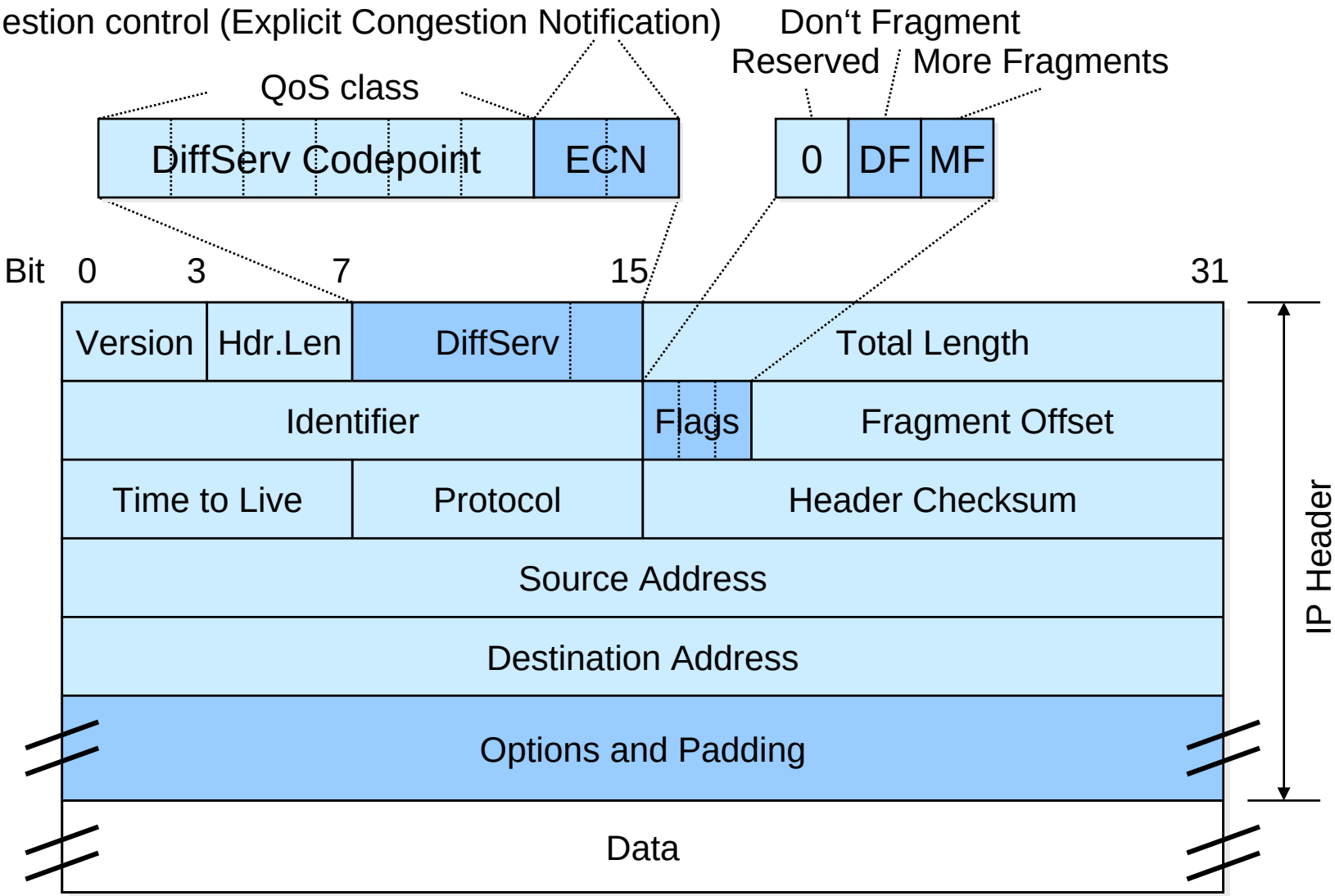
The Internet Protocol – IP

- History
  - Original development with support of the US Department of Defense
  - Already used back in 1969 in the APANET
- Tasks
  - Routing support
  - Check of packet lifetime
  - Segmentation (called fragmentation) and reassembly
  - …
- Development
  - Today IP version 4 is the most used layer 3 protocol
  - Further development started back in the 80s/90s – project IPng (IP next generation) of the IETF (Internet Engineering Task Force)
  - Result in the mid 90s: IPv6, still only rarely used

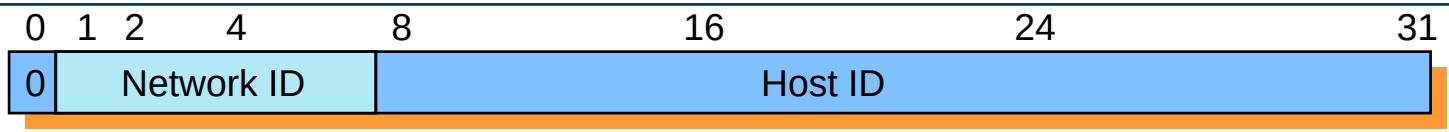
Properties of IP

- Packet oriented
- Connectionless (datagram service)
- Unreliable transmission
  - Datagrams can be lost
  - Datagrams can be duplicated
  - Datagrams can be reordered
  - Datagrams can circle, but solved by Time to Live (TTL) field
  - IP cannot handle layer 2 errors
  - At least there is ICMP to signal errors
- No flow control (yet, first steps taken)
- Used in private and public networks

IPv4 datagram



IP addresses and address classes (classical view)



1. Class A: 128 networks, 16M hosts

1.0.0.0 – 127.255.255.255

2. Class B: 16k networks, 64k hosts

128.0.0.0 – 191.255.255.255

3. Class C: 2M networks, 256 hosts

192.0.0.0 – 223.255.255.255

4. Class D: group communication (Multicast)

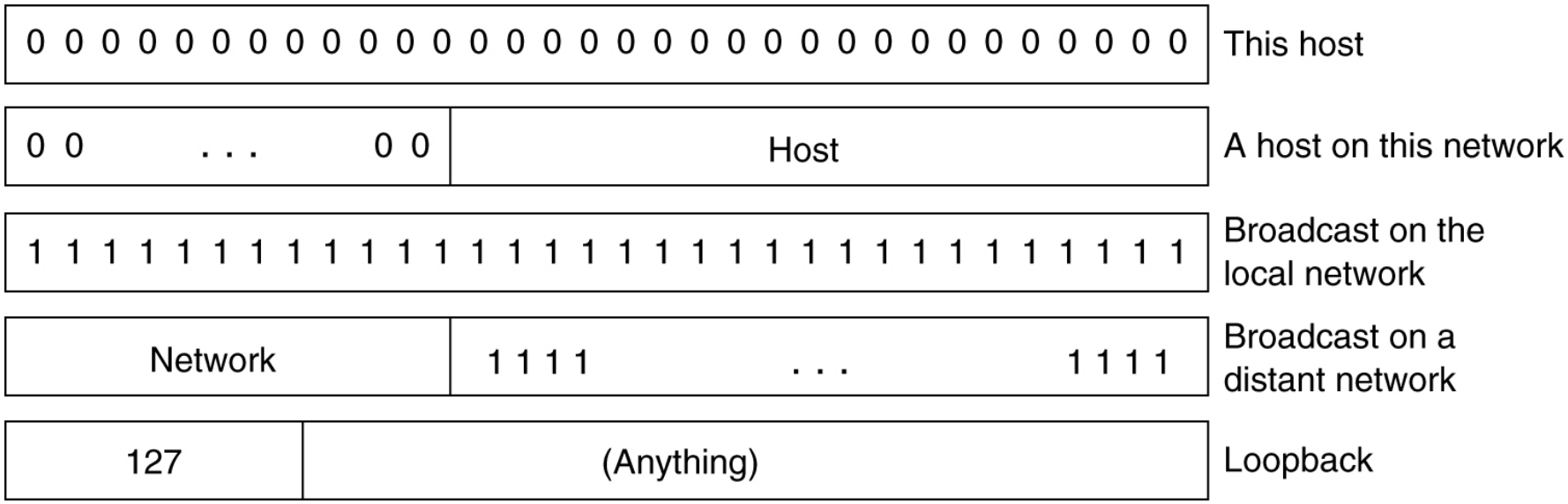
224.0.0.0 – 239.255.255.255

5. Class E: reserved for future use

240.0.0.0 – 255.255.255.255

Some reserved IP addresses

- Some IP addresses are set aside for special uses
  - Not all of the network/host combinations are available
  - So-called "private" IP addresses
    - Used for internal networks (addresses not routable)
    - Example: 10.0.0.1



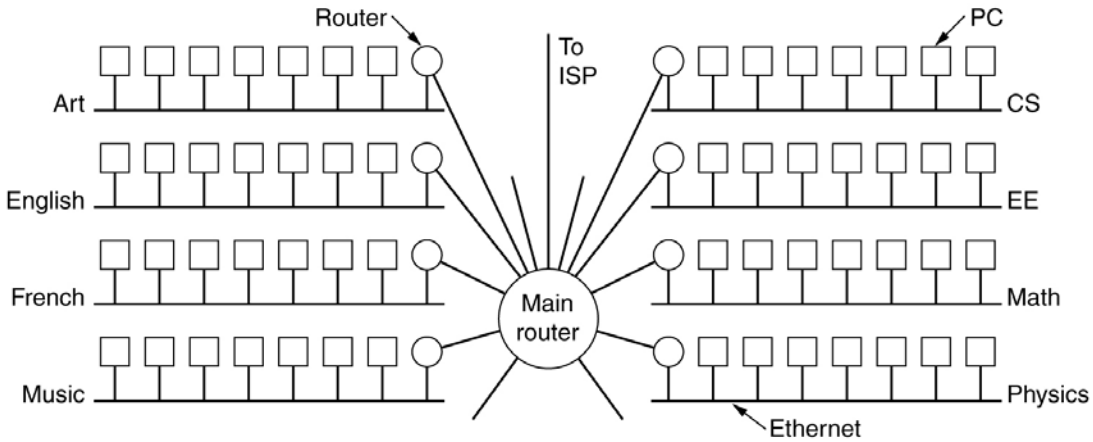
Growing pains…

- How does IP hold up when networks grow?

- Two main problems
  - Class A and B networks can contain MANY hosts, too many for a router to easily deal with (plus: administrative problems in larger networks)
    - Solution: subnetting
  - Network classes waste a lot of addresses
    - Example: Organization with 2000 hosts requires a class B address, wasting 64K-2K ≈ 62.000 host addresses!
    - Solution: Classless addressing ➔ CIDR (Classless Inter Domain Routing, not covered here)

Subnetting

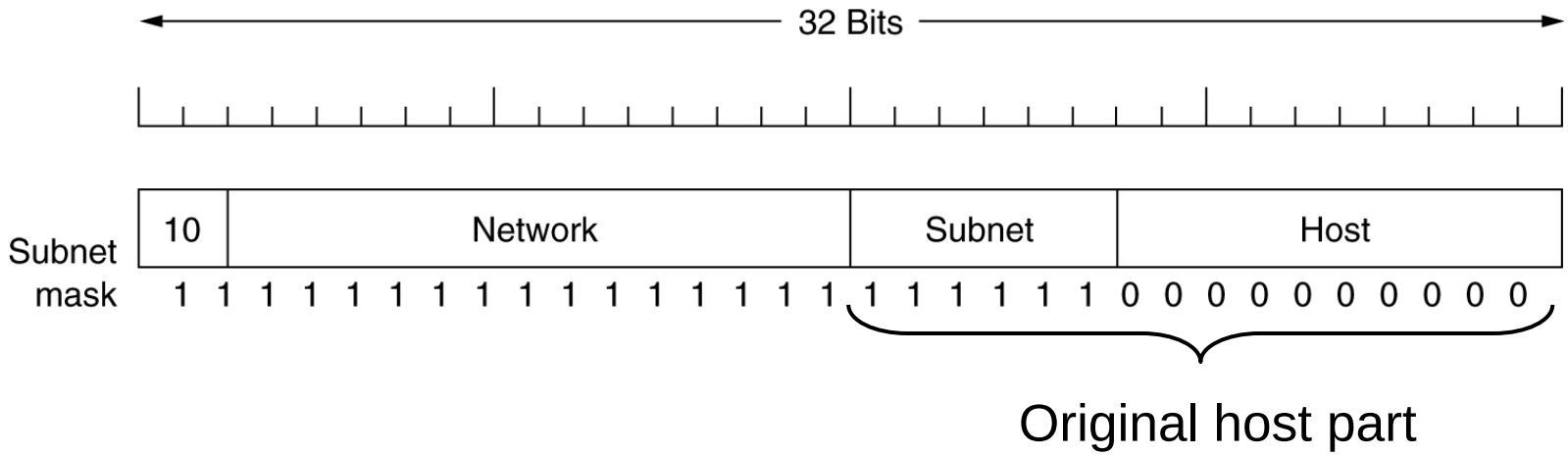
- Suppose an organization has a class B address but is organized into several LANs
  - Example: university with different departments



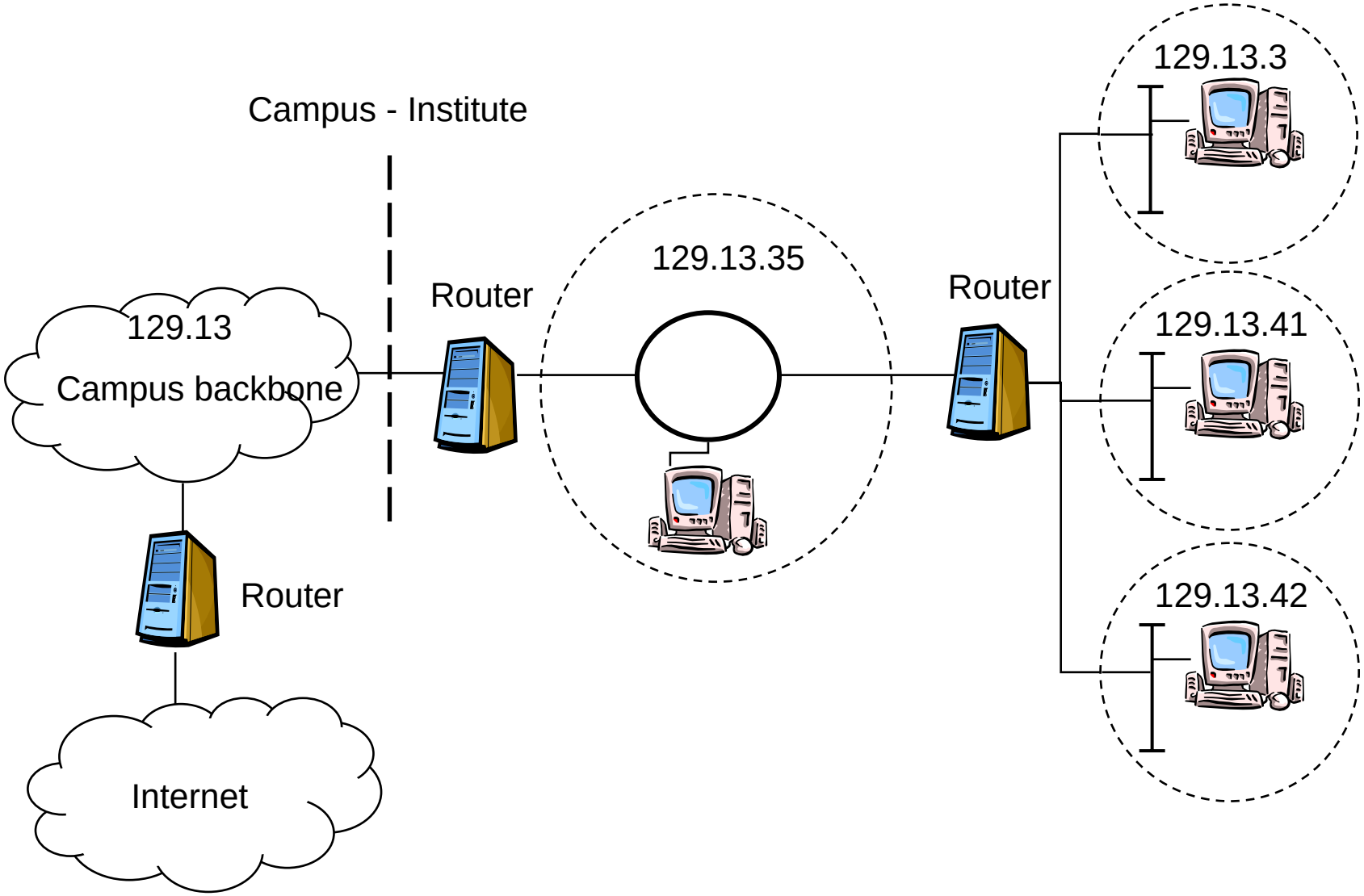
  - Main router does not really want to bother with all the nodes in the individual departments but wants to look only at whole networks
  - Obvious case for hierarchical routing and addressing – how to put hierarchies into existing IP addresses?

Subnetting – Hierarchies in addresses

- Manipulating the class bits to introduce more hierarchy levels is not practical
- Idea: have more hierarchy levels implicitly
  - Introduce a ***subnet***, represented by "borrowing" bits from the host part of the IP address
  - Local router has to know where to apply this split – needs a ***subnet mask***
  - Represented as u.x.y.u***/#bits*** or as bit pattern needed to mask out the host bits

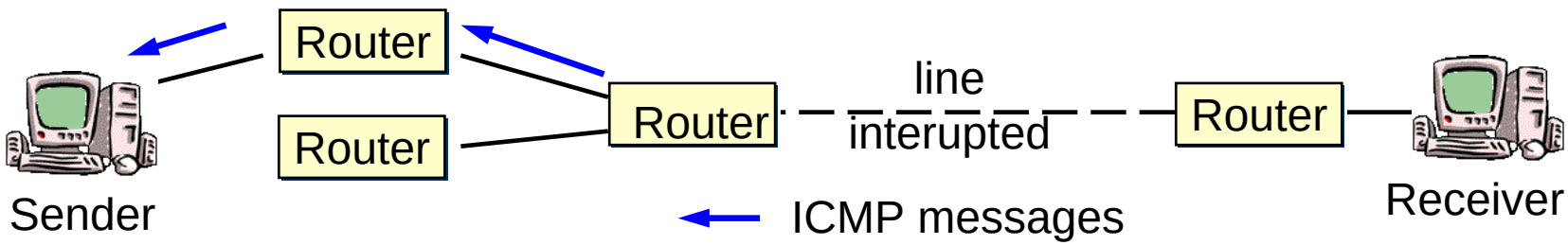


IP subnet example



Controling IP: ICMP

- IP is responsible for (unreliable) data transfer only
- In case of errors or for testing ICMP is used (Internet Control Message Protocol)



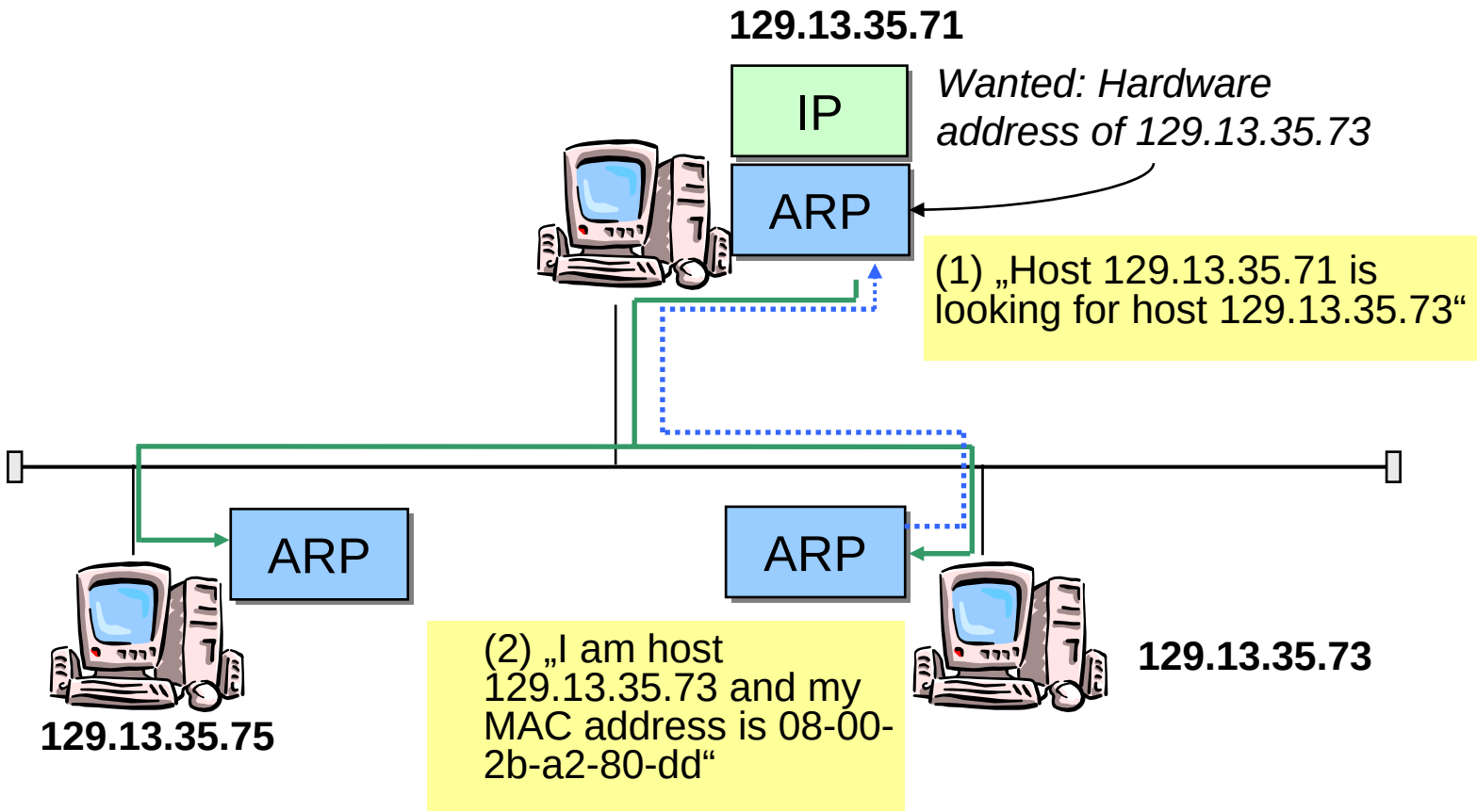
- Examples
  - Destination Unreachable
  - Time Exceeded: Time-to-Live field = 0
  - Echo Request / Reply ("ping").
  - Timestamp Request / Reply

Bridging addressing gap: ARP

- What happens once a packet arrives at its destination network / LAN?
  - How to turn an IP address (which is all that is known about the destination) into a MAC address that corresponds to the MAC address?
- Simple solution: Yell!
  - Broadcast on the LAN, asking which node has IP address x
  - Node answers with its MAC address
  - Router can then address packet to that MAC address

- ***Address Resolution Protocol (ARP)***

ARP – Example


129.13.35.71
IP
ARP
Wanted: Hardware address of 129.13.35.73
(1) „Host 129.13.35.71 is looking for host 129.13.35.73“
ARP
ARP
(2) „I am host 129.13.35.73 and my MAC address is 08-00-2b-a2-80-dd“
129.13.35.73
129.13.35.75

Conclusion Internet Protocol

- Unreliable datagram transfer
- Needs supporting protocols
- Version 4 dominant, version 6 coming (since years…)
- Classical addressing wastes addresses
  - Subnetting, subnet masks
  - Classless addressing, see CIDR
- ICMP for error signaling
- ARP for mapping IP to MAC address