Networking

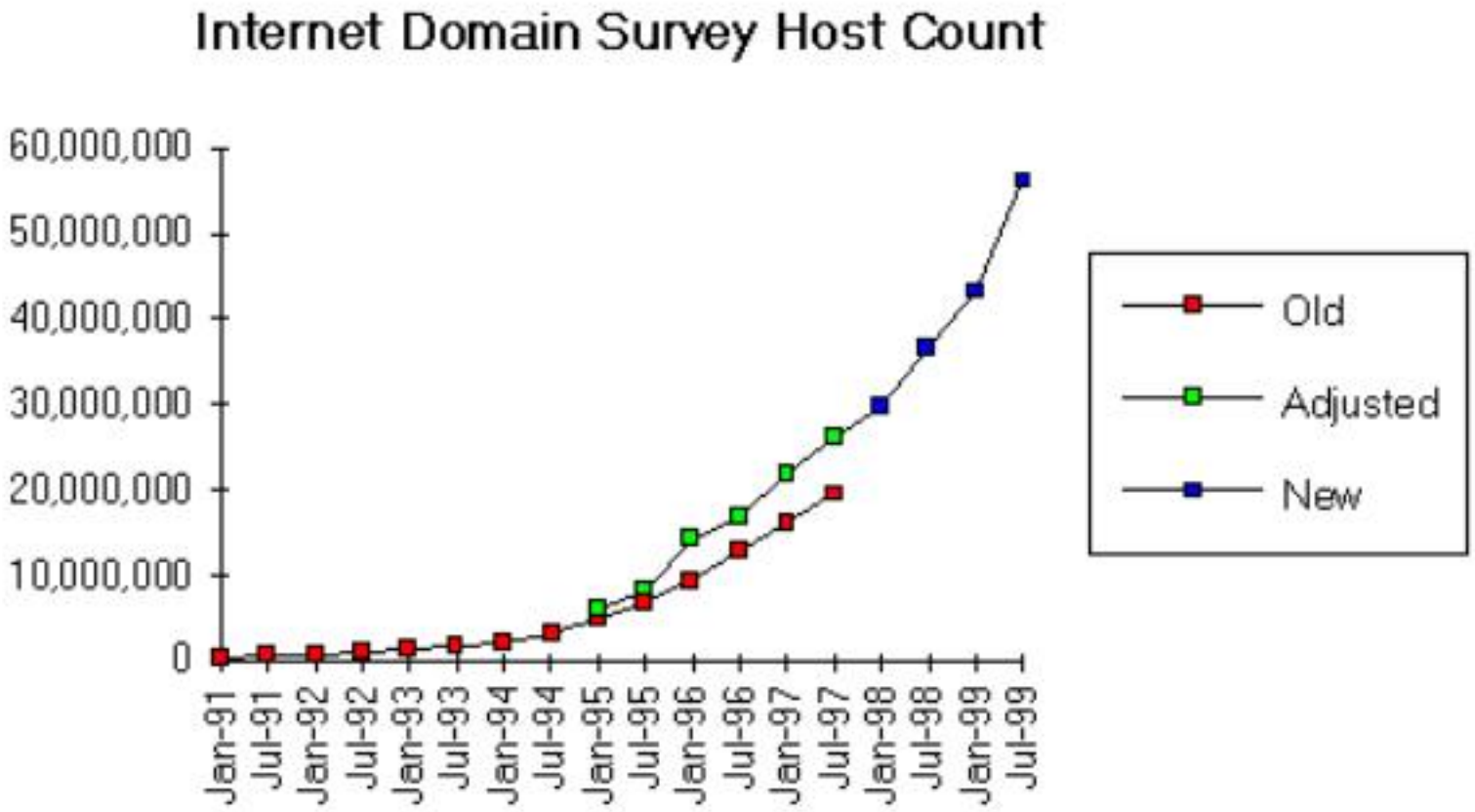
Stavros Tripakis (stavros@eecs)

Introductory lecture for EE290T

Outline

- The Internet: history, challenges.
- Layered architecture of networks.
- Application layer: Web, E-mail, Domain Name Service (DNS).
- Transport layer: TCP/UDP, flow control.
- Network layer: IP, routing, fragmentation/reassembly.
- Link layer: LANs (Ethernet, Token-rings, Wireless.

Internet Grows Exponentially

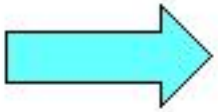


Source: Internet Software Consortium (http://www.isc.org/)

Who benefits from the Internet ?

- Users:
  - Companies (production cycle, marketing, ...).
  - Subscribers (communication, information, shopping, entertainment, ...).
- Providers:
  - ISPs, name-administration companies, PTTs.
  - Computer vendors.
  - Network-specialized technology vendors (ethernet cards, IP routers, ATM switches, ...).

Who benefits from the Internet ?



The Internet is both
a **product** and a **tool**.

Similar *bearer services*: postal, telephone.

Goal:

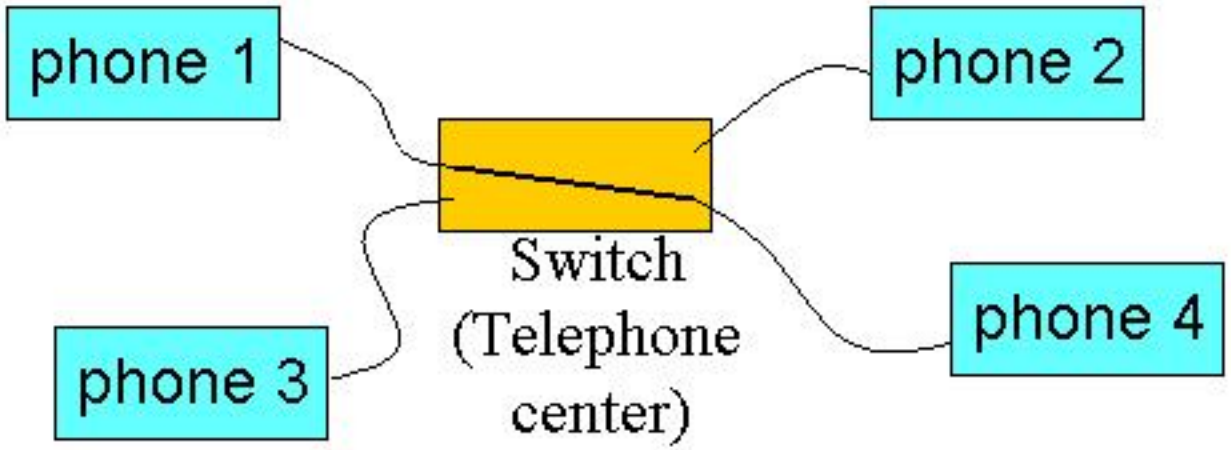
Make the Internet a **useful tool**.

Key requirements:

- **Interoperability.**
- **Diversity / Extensibility.**
- **Scalability.**
- **Performance (Cost-effectiveness).**

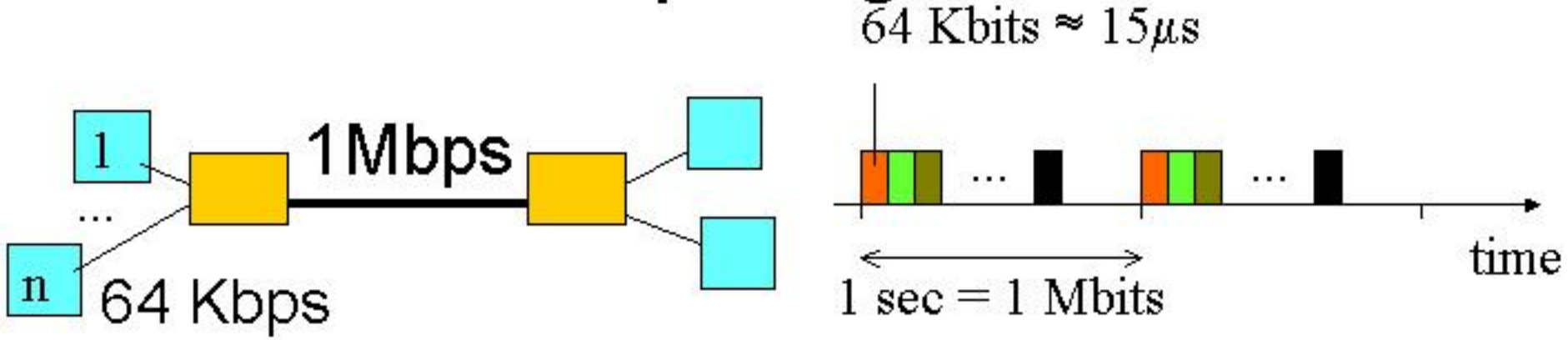
The telephone network: a brief history.

- 1890: analog, switching manual:



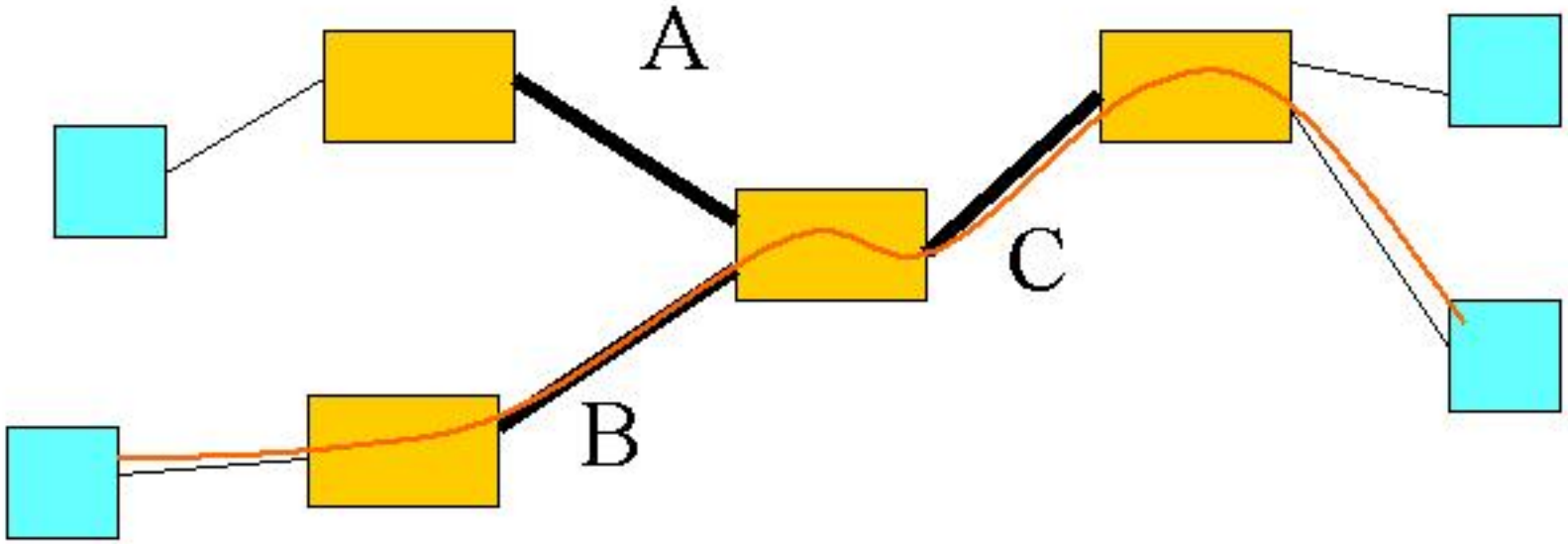
The telephone network: a brief history.

- Today:
  - Digital: voice → bit stream (64 Kbps).
  - Switches = computers.
  - Better channel utilization by **time-division multiplexing**:



The telephone network: a brief history.

- **Circuit switched** network:
  - each connection gets 64Kbps end-to-end
  - reservation fixed for the whole transmission

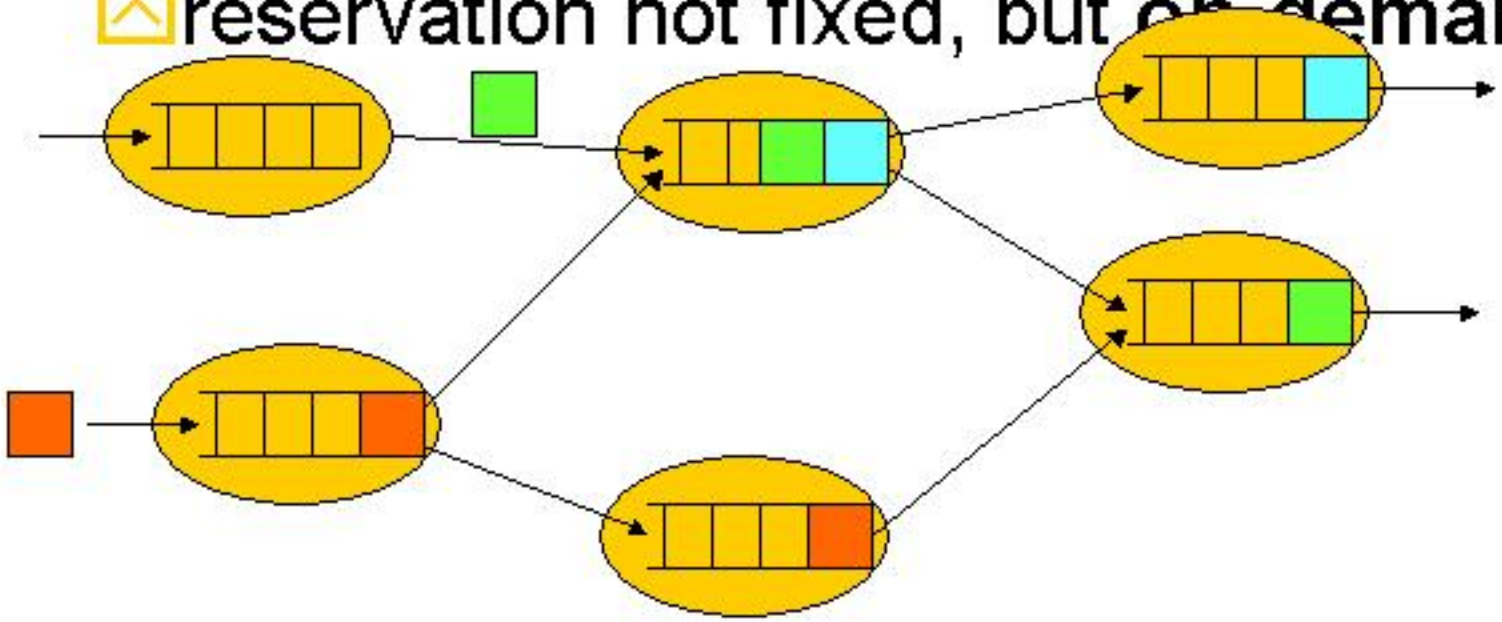


The telephone network: properties.

- Interoperability : good.
- Scalability : good.
- Cost-effectiveness : OK.
- Diversity : limited (constant-bit-rate).
- Extensibility : very limited.

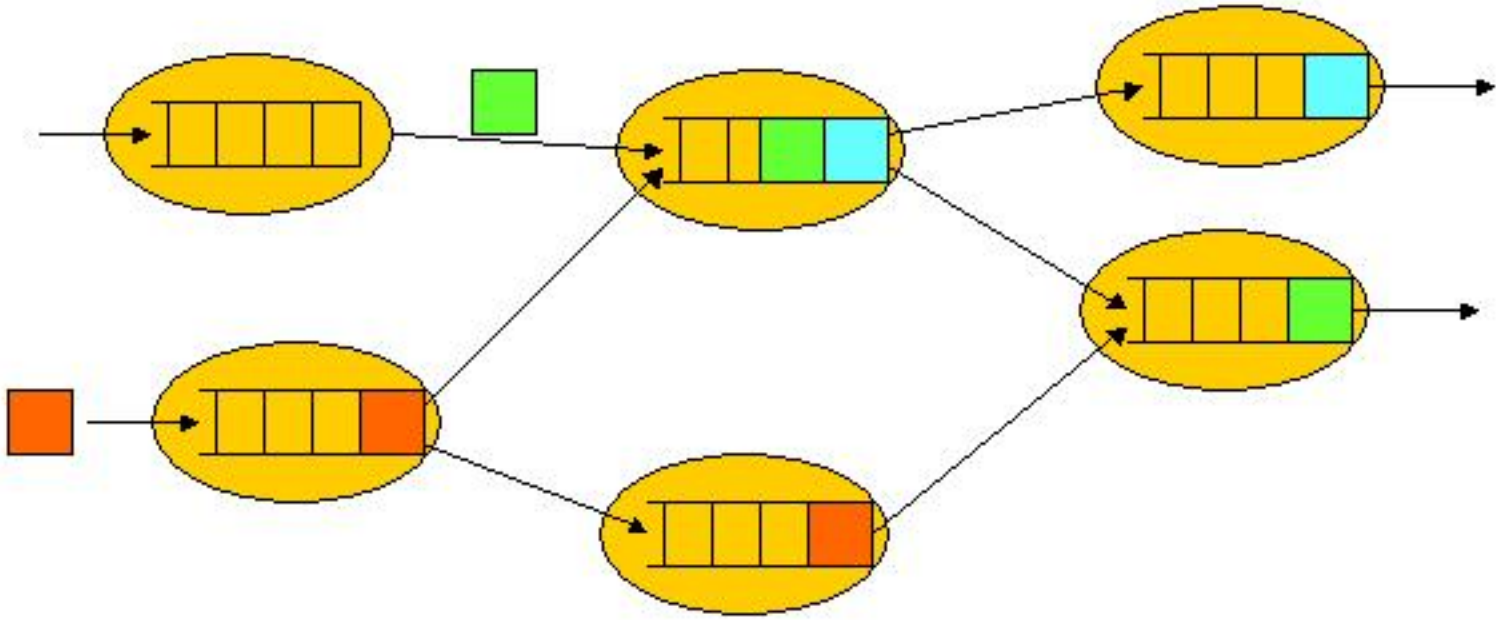
The Internet : a brief history.

- **Packet-switched** network:
  - packets **share** resources (buffers, links)
  - reservation not fixed, but on demand



The Internet : a brief history.

- multiple links (connectivity, reliability)
- buffers (store, process, forward)
- control information in packets (s,d,seq#)



The Internet: properties.

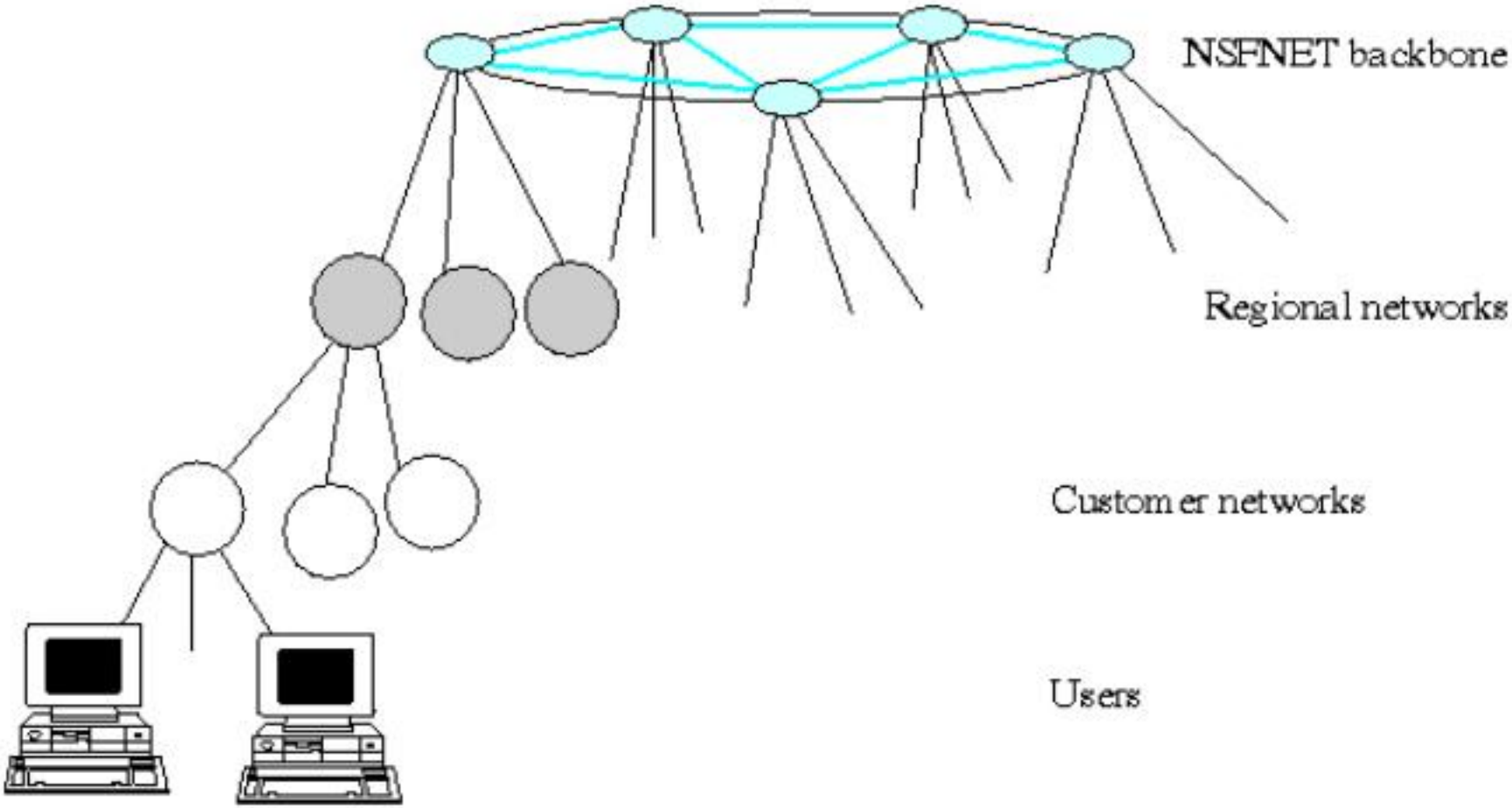
- Interoperability : good.
- Scalability : good (IP addresses ?).
- Diversity / Extensibility : very high, but no guarantees for applications.
- Cost-effectiveness : very good.

Broadband Integrated Services Data Network (BISDN)

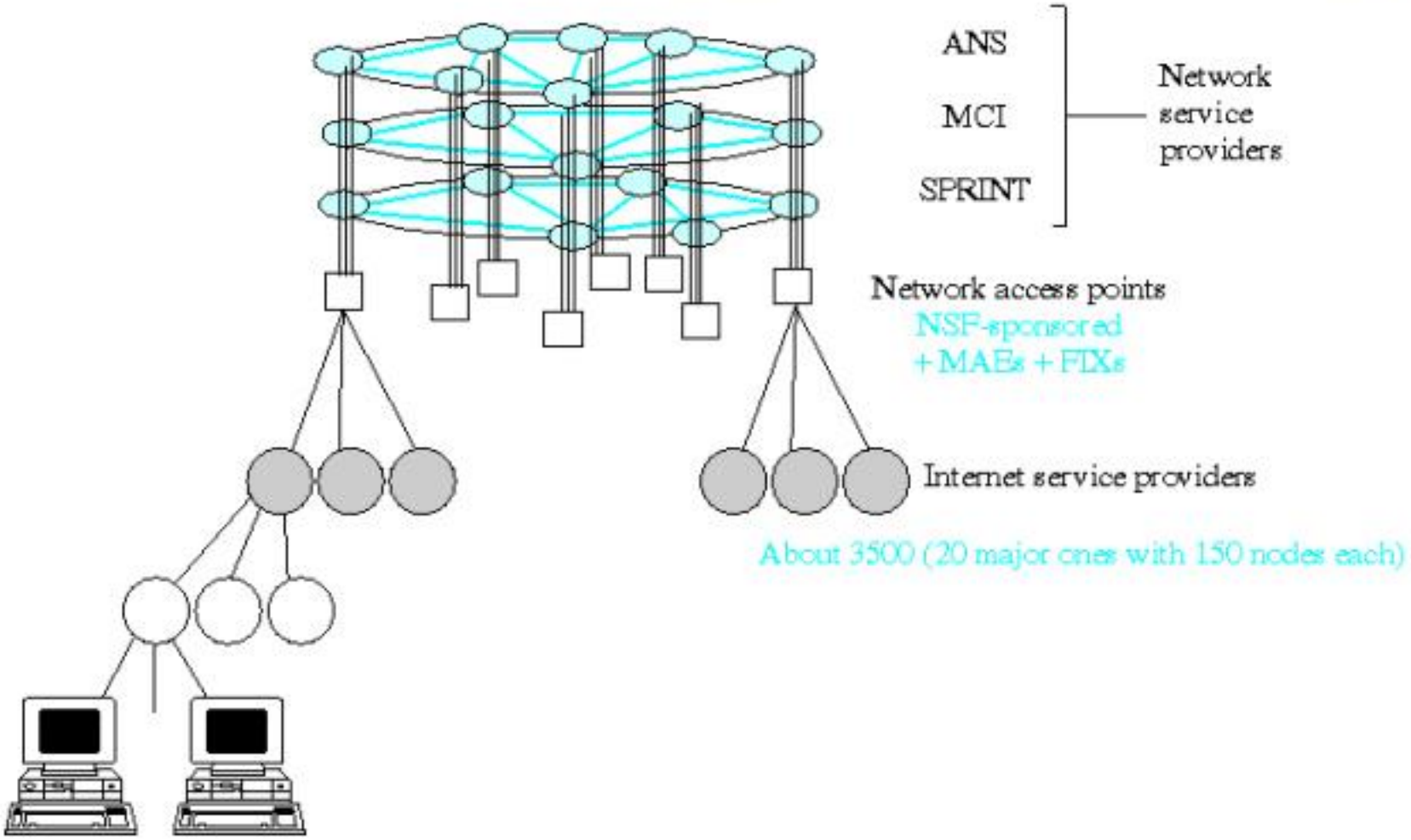
Can the Internet be extended so that it supports all kinds of applications ?

- Many proposed technologies:
  - ATM: extending synchronous (TDM) telephone networks to *asynchronous* (*statistical multiplexing*): still connection-oriented (*virtual-circuits*).
  - More recent approaches: DiffServ, etc.

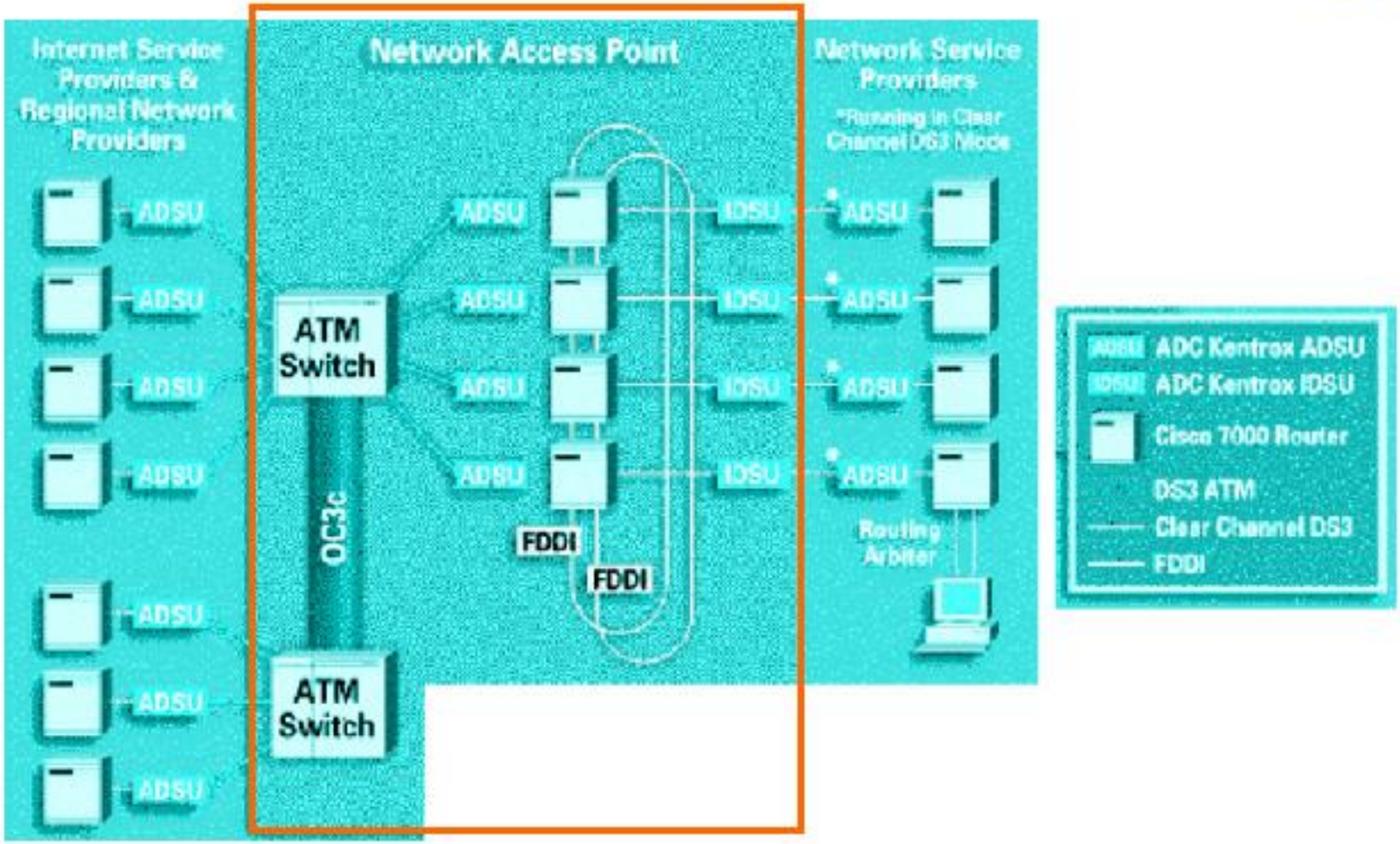
The Internet around 1990



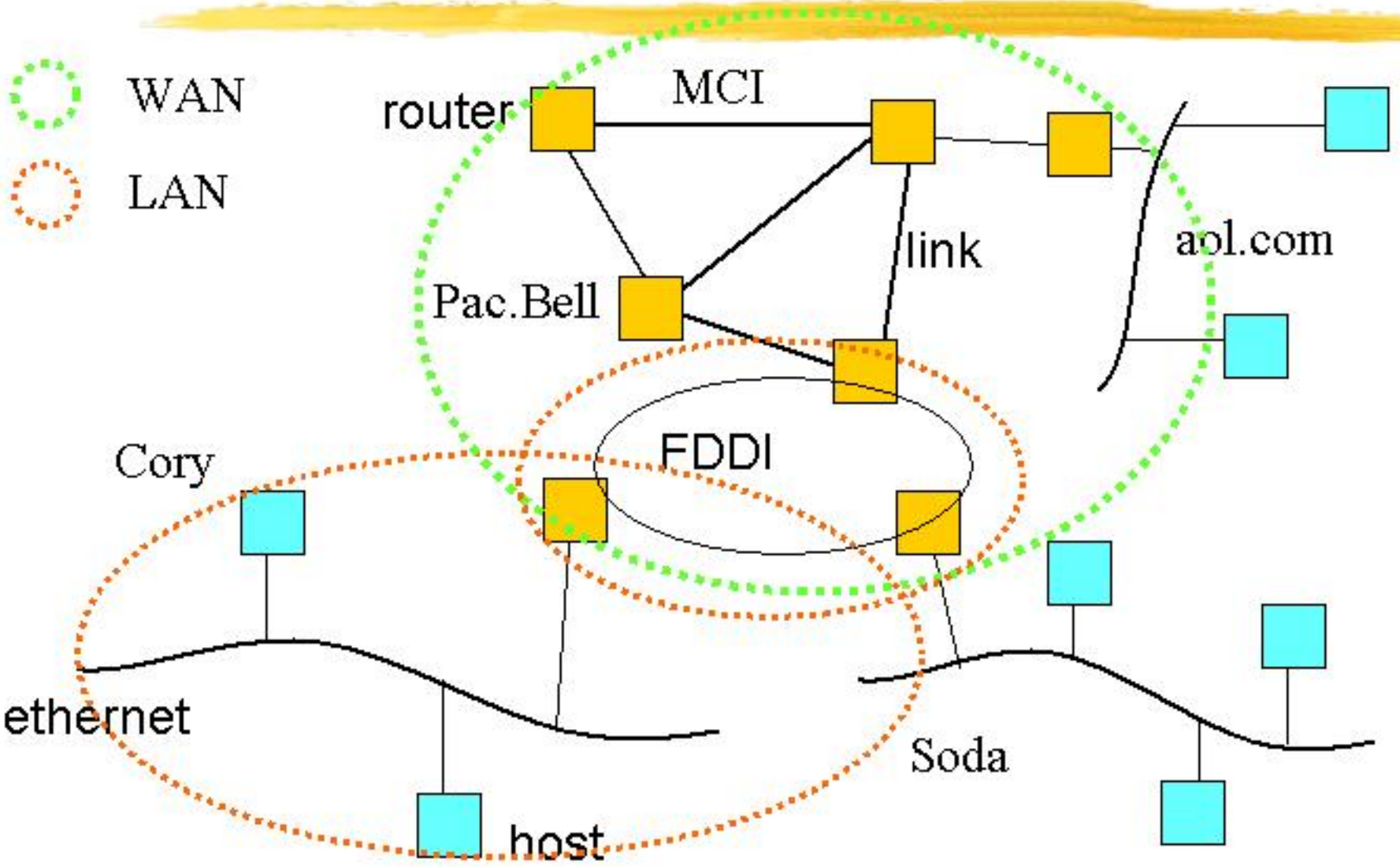
The Internet in 1997



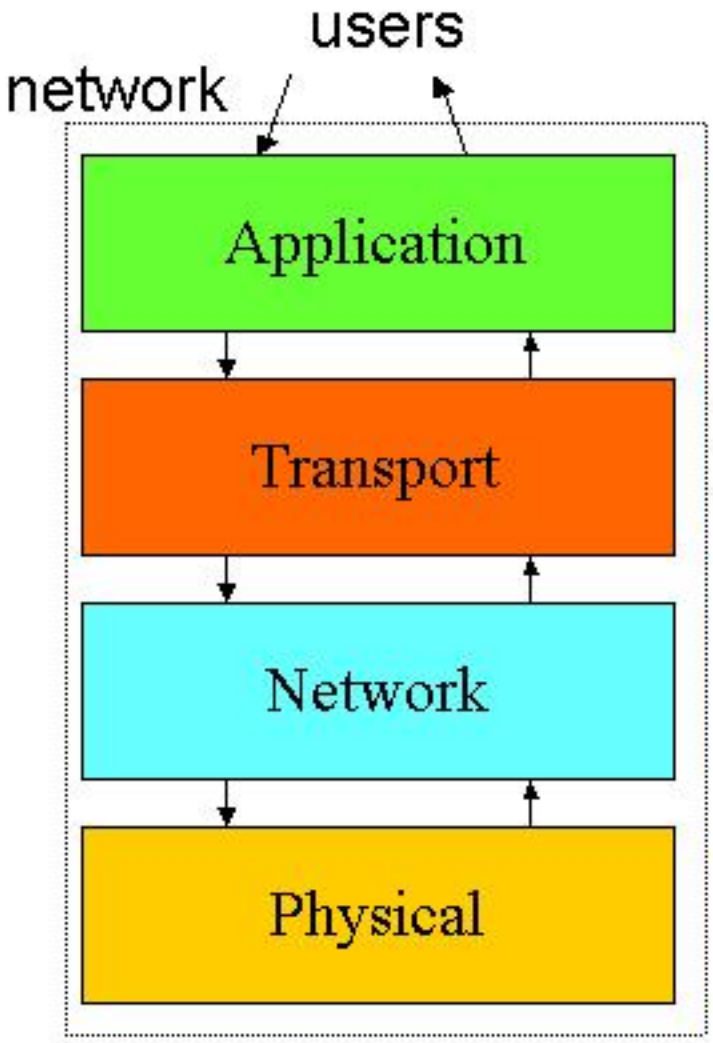
A typical Network Access Point (NAP)



A small Internet



Internet protocol stack: a layered architecture



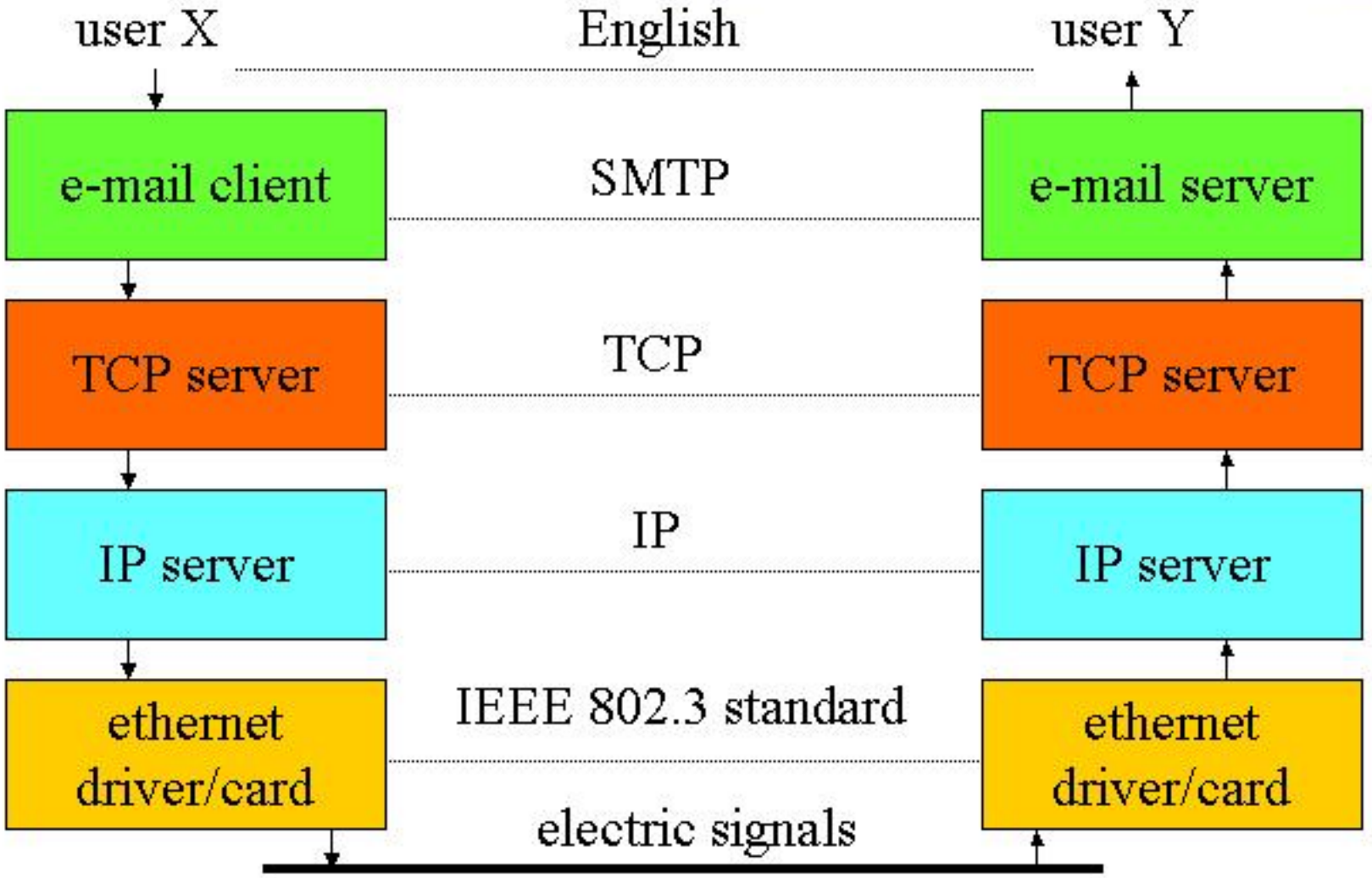
Web (HTTP), E-mail (SMTP),
File transfer (FTP), Name resolution (DNS),
Remote terminal (TELNET), …

Reliable multi-connection bit-stream (TCP),
unreliable multi-connection (UDP).

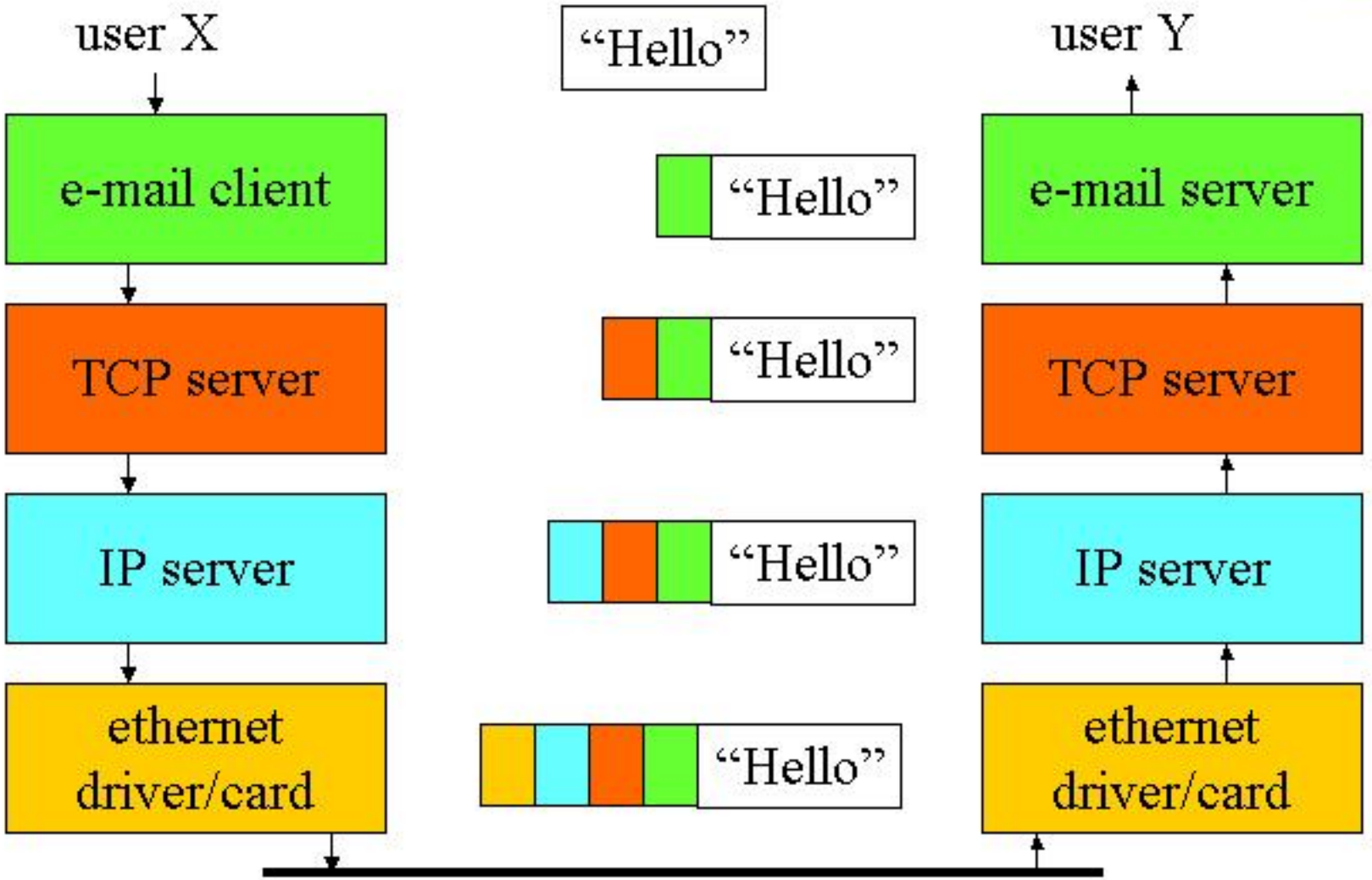
Unreliable end-to-end delivery of
packets up to 64 Kbits (IP).

Point-to-point links (PPP, SONET, …),
LANs (Ethernet, FDDI, wireless, …)

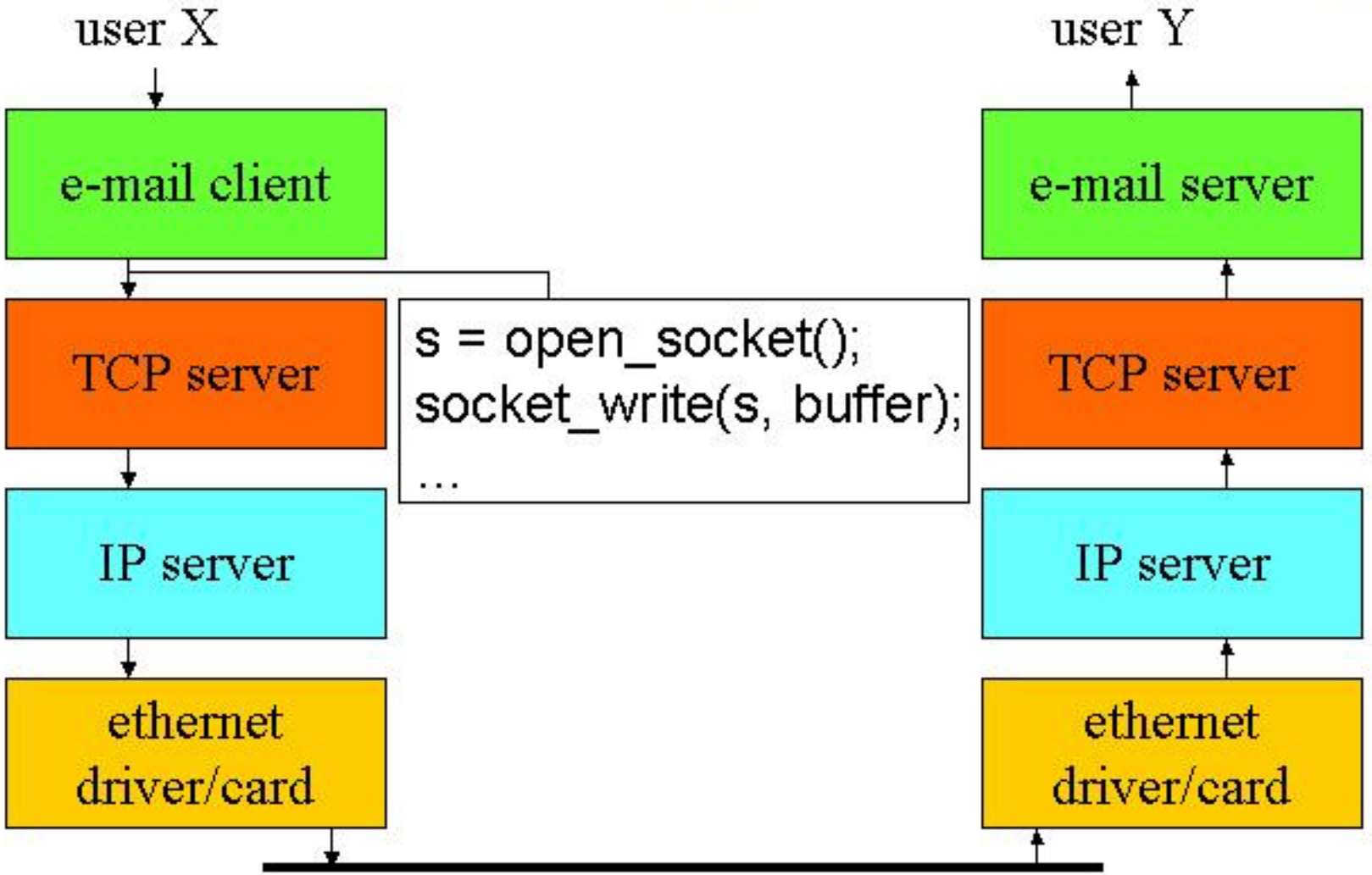
Internet protocol stack



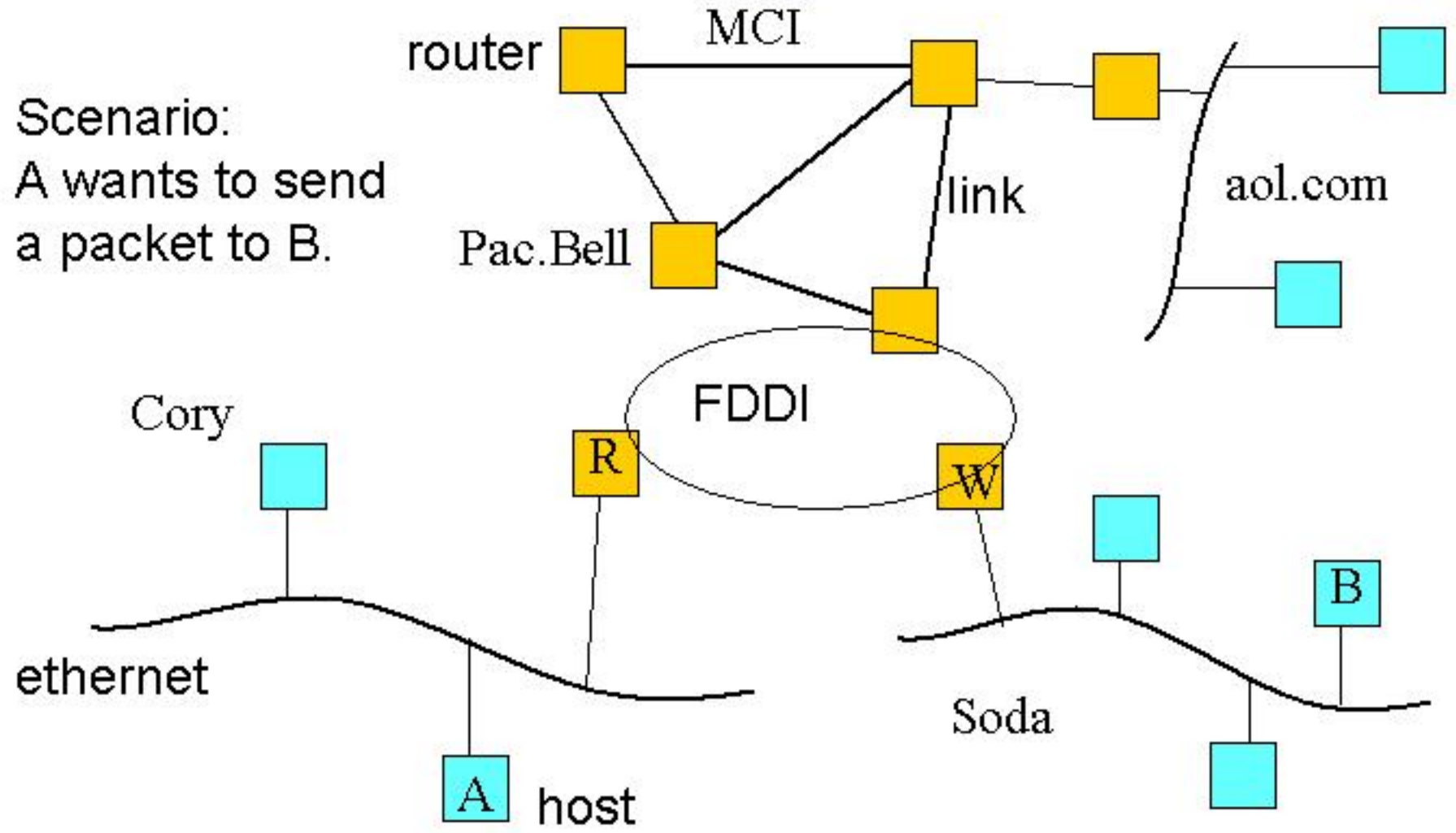
Protocol encapsulation



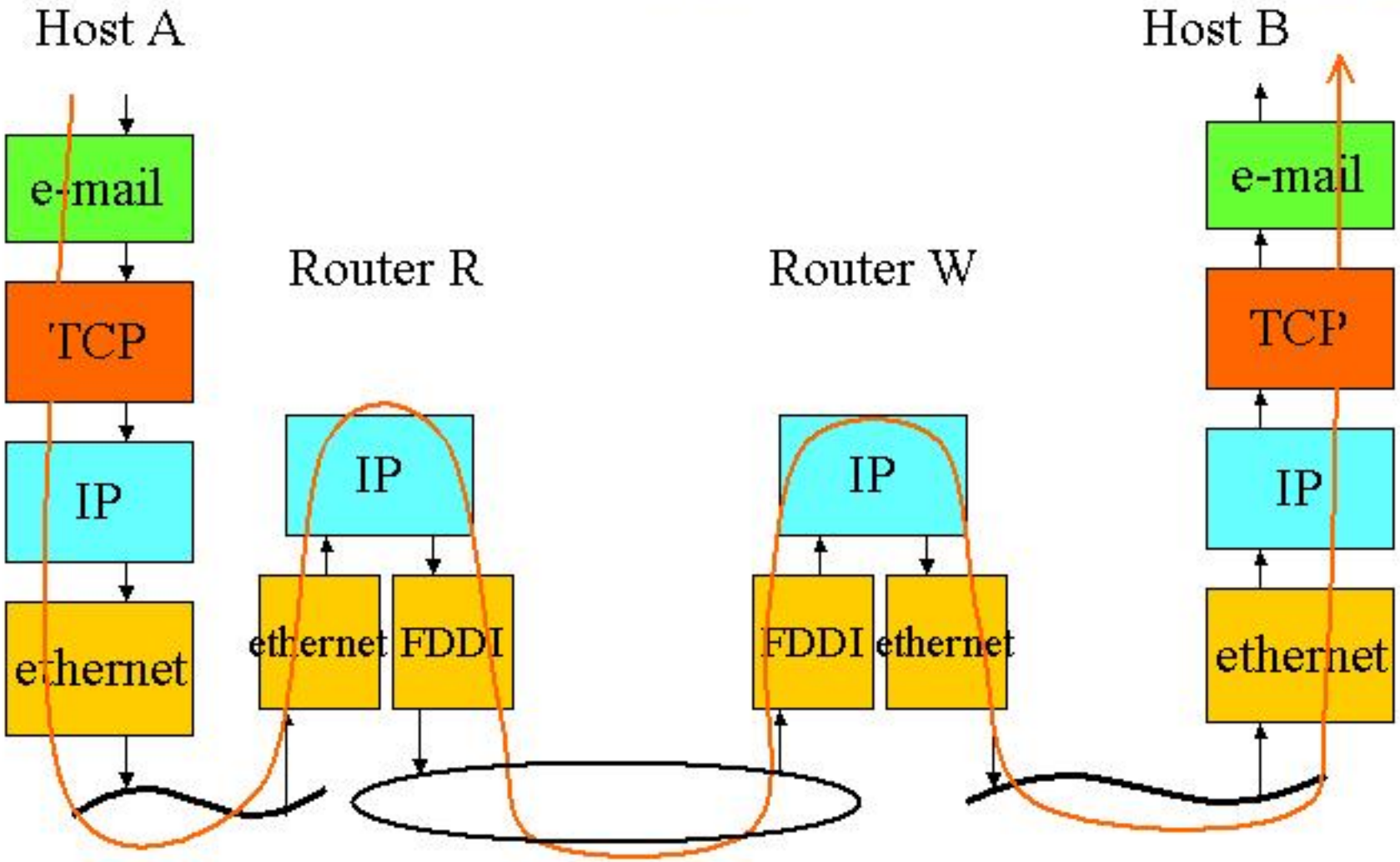
Protocol interfaces



A small Internet



Protocol stack: packet forwarding in IP



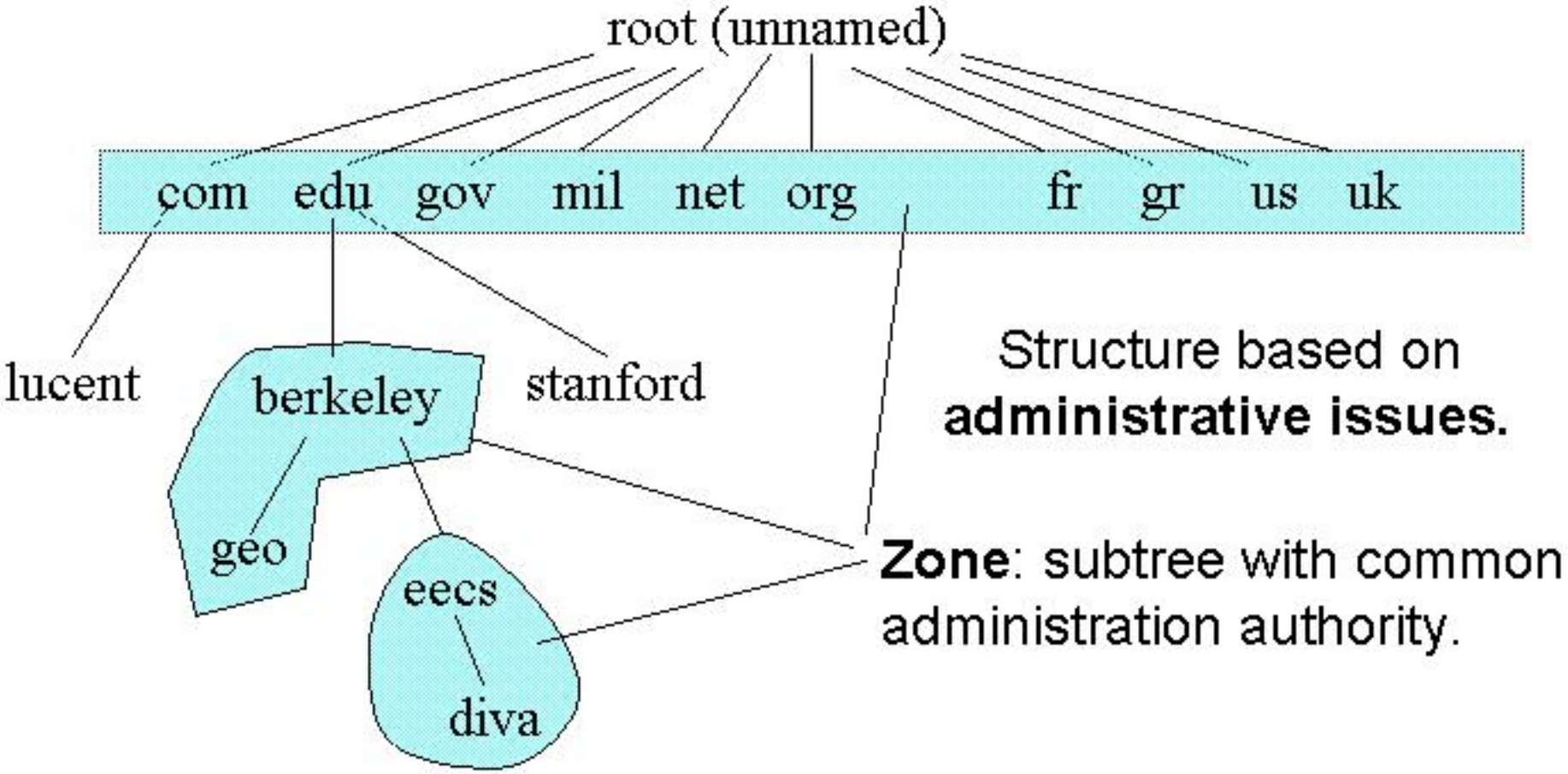
Layered Architectures

- Break-up design problem intro smaller, more manageable problems: build sophisticated services from more basic services.
- Modular design and implementation
  $\Rightarrow$ easy to extend/modify.
- Difficult to maintain efficiency: careful with interaction of layers for efficiency.

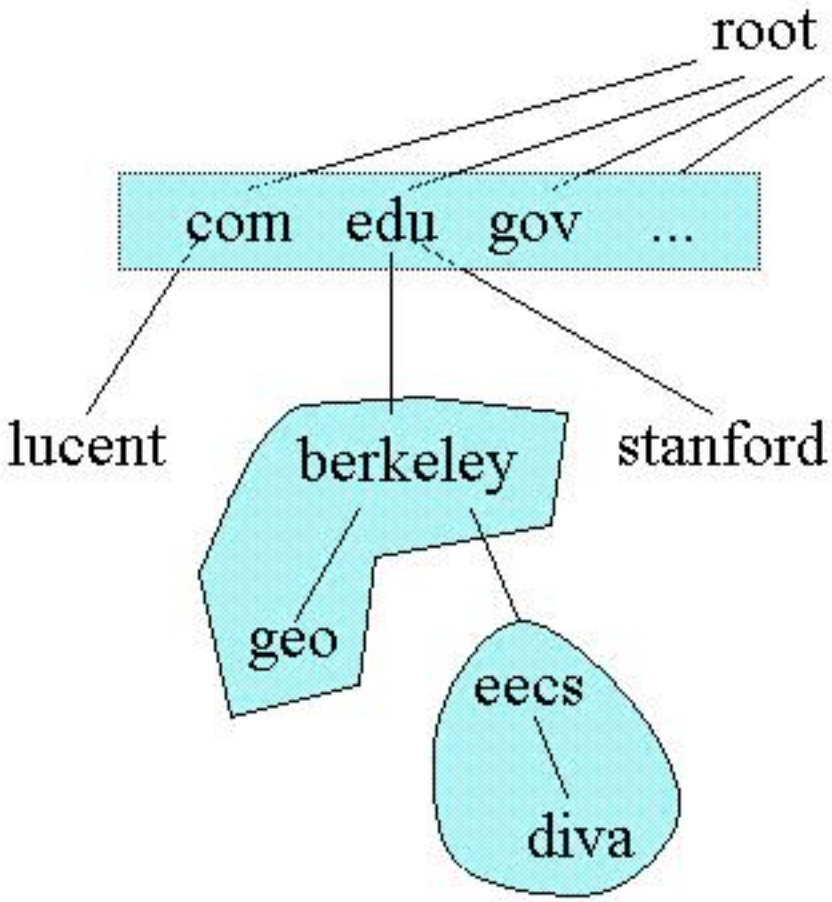
Names and addresses

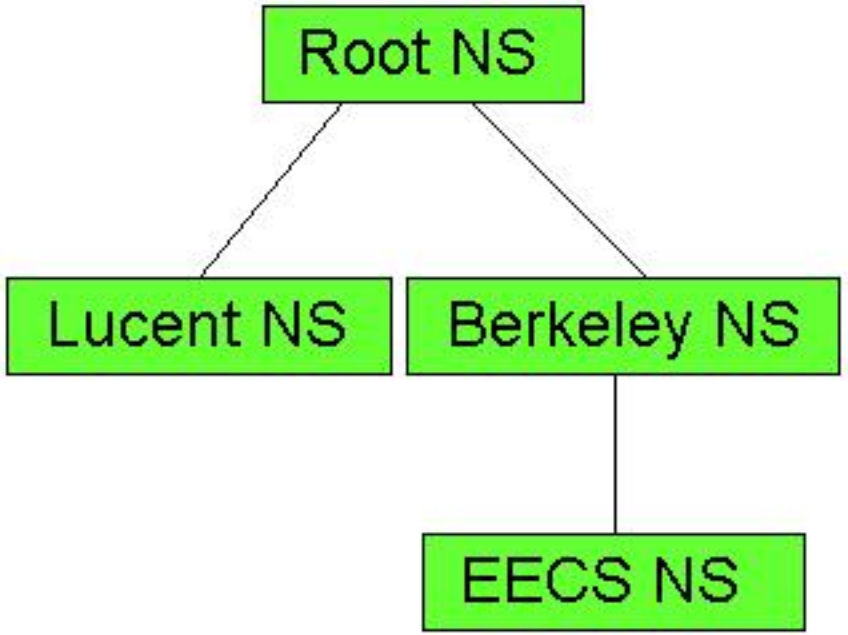
- Each layer has its own means of identification.
- Hosts have names, e.g., stout.eecs.berkeley.edu (easy to remember).
- Hosts have IP addresses, e.g., 128.32.239.44.
- Applications identified by (IP addr., port #).
- LANs have their own addressing schemes, (e.g., 6-byte Ethernet addresses).

Domain name structure

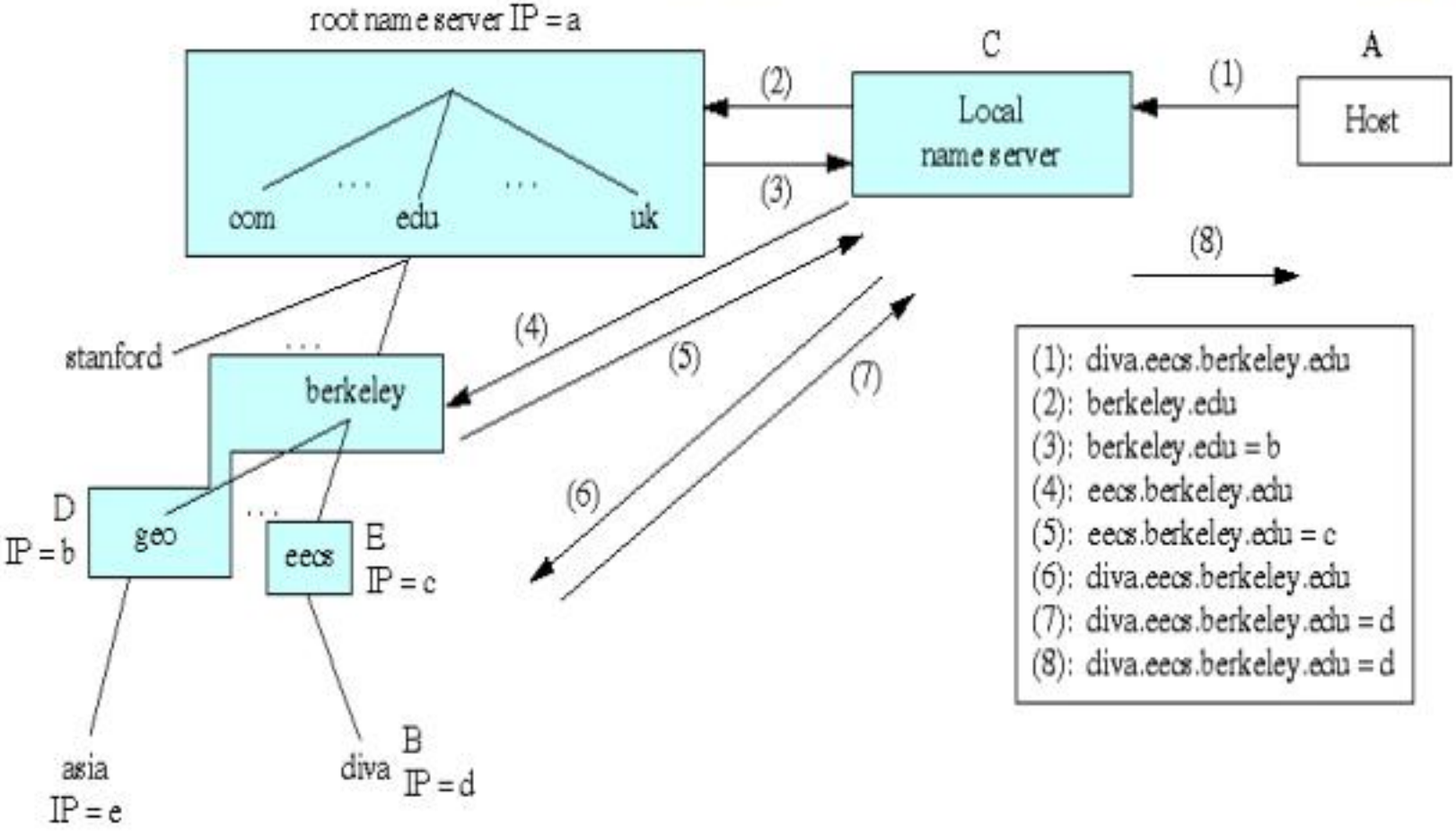


Name Servers (NS)





Name resolution: example

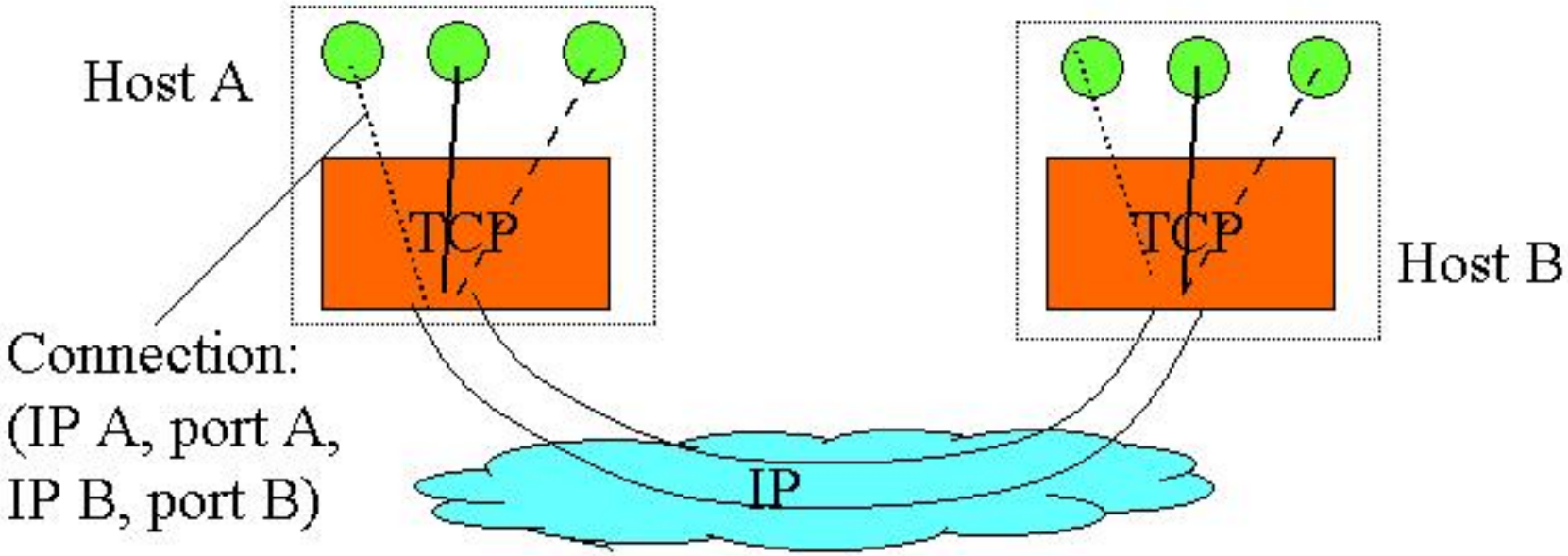


Applications: client-server model

- **e-mail** from X@myPC.bla.com to Y@foo.edu.
  - E-mail client running in myPC.bla.com asks DNS for the IP address of the mail server of foo.edu (say this is M).
  - E-mail client asks TCP server of myPC to open a TCP connection to M at well-known port 25.
  - E-mail client "talks" SMTP to deliver mail to mail server, then closes the connection.
  - IP finds out whether M is a local or remote host. If remote, IP routes the packets to M. IP uses Ethernets, point-to-point links, routers, etc. to transmit bits.
- **Web**: web-browser (client) web-server.
  - They talk HTTP, but also FTP, SMTP, and others.
  - Potentially more than one TCP connections per web-page.

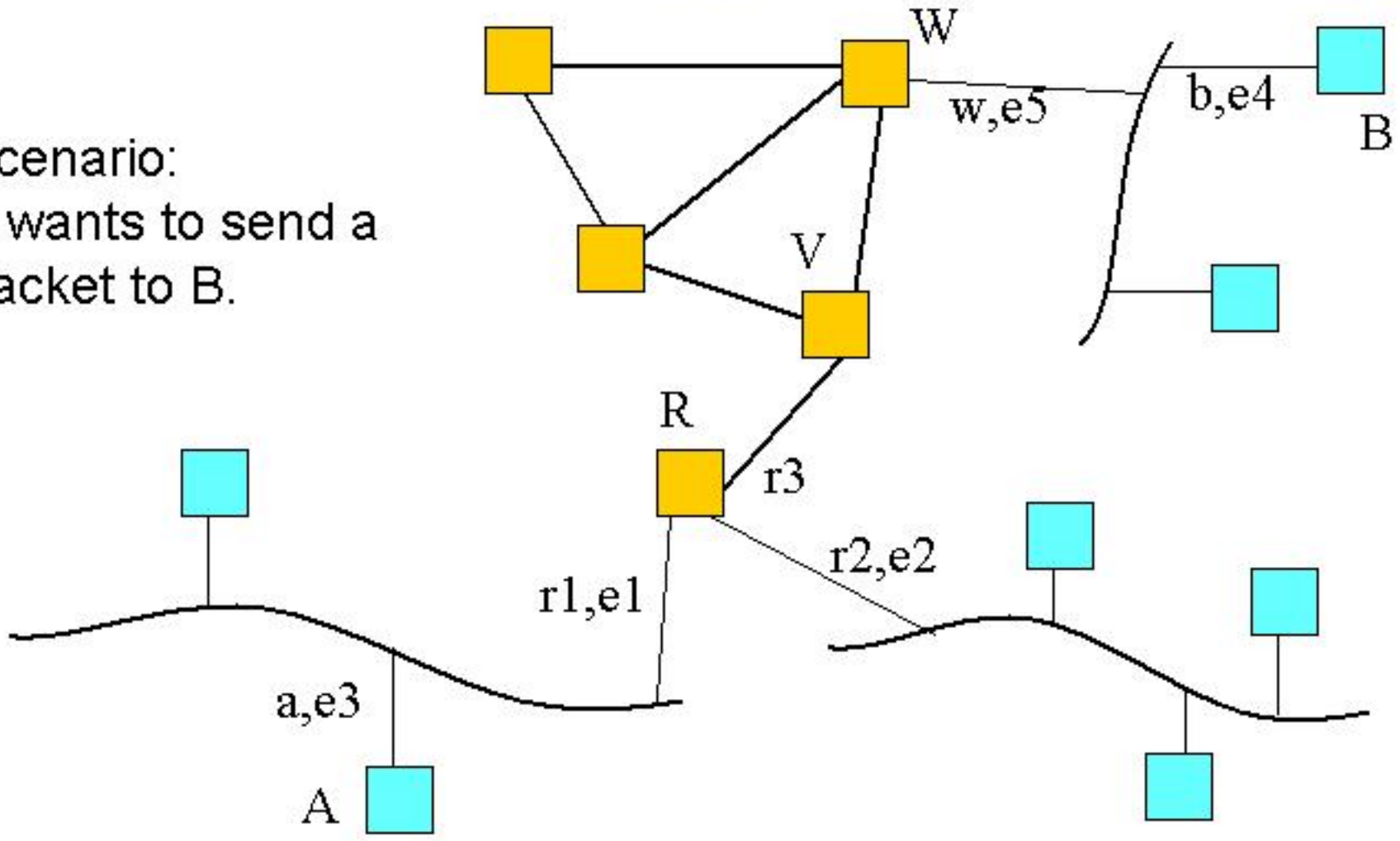
TCP (Transmission Control Protocol)

- Multiplexing (multiple connections between two hosts).
- Reliable, ordered transmission of data.
- Flow control, congestion control.



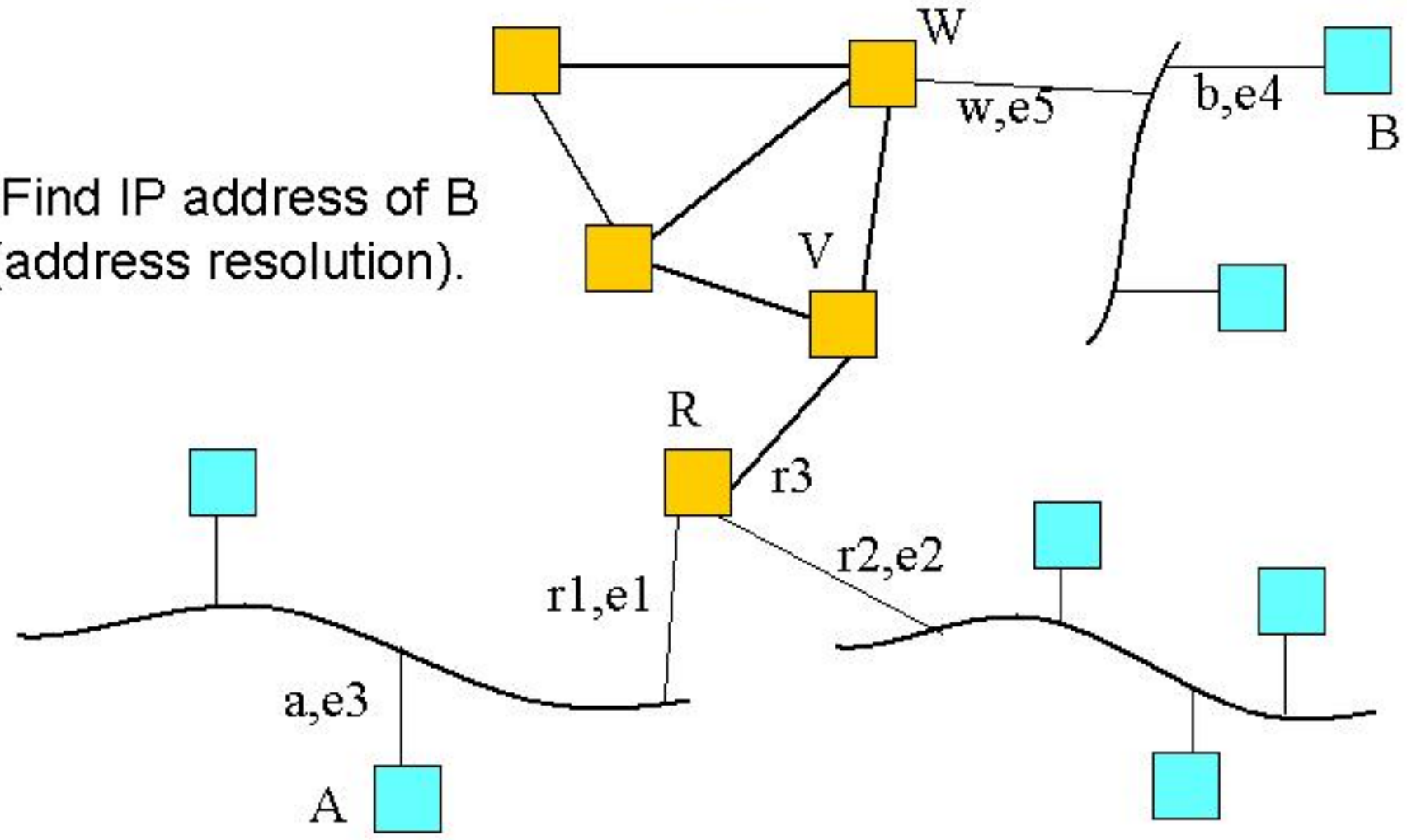
A small Internet

Scenario:
A wants to send a packet to B.



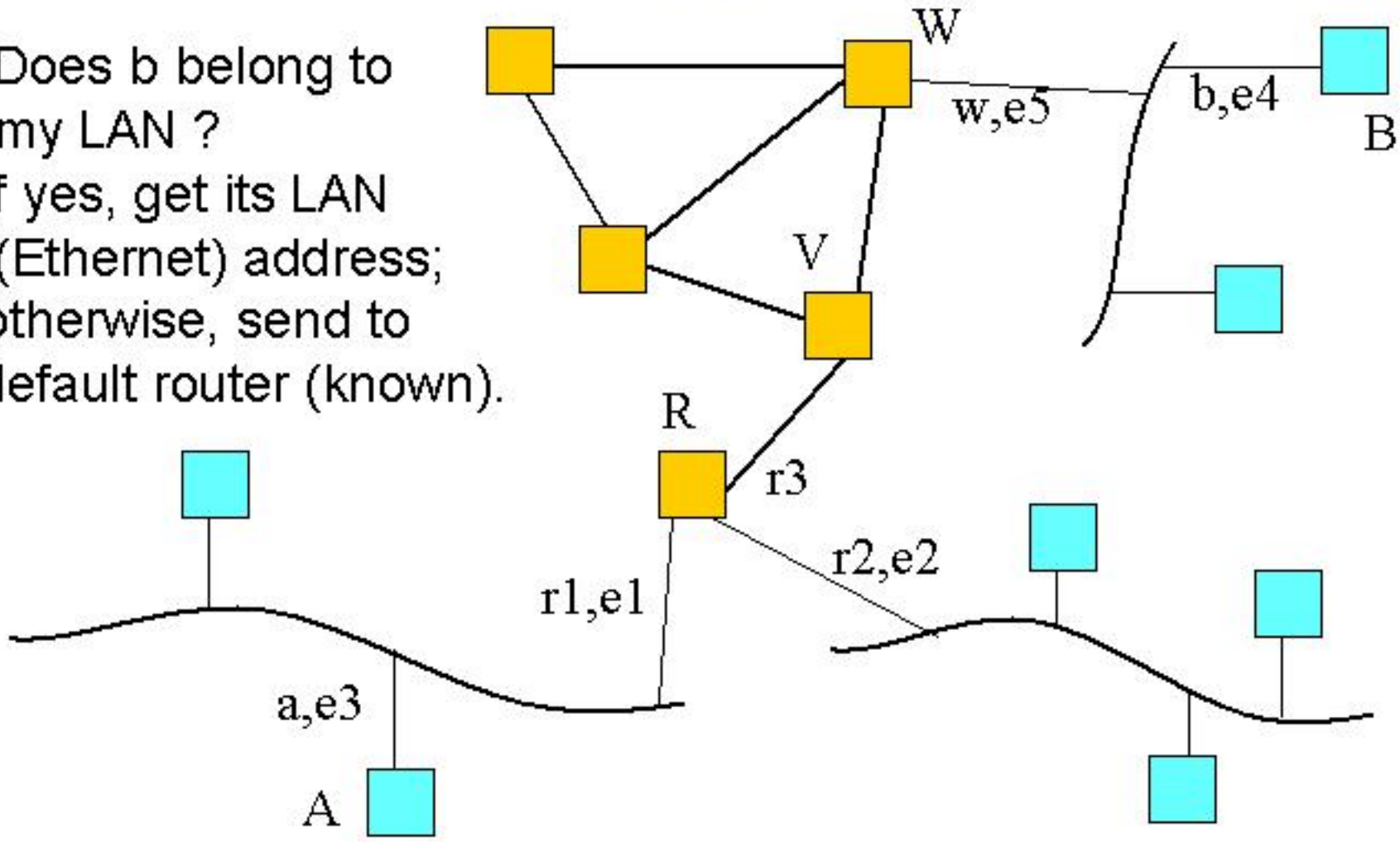
A small Internet

1. Find IP address of B (address resolution).



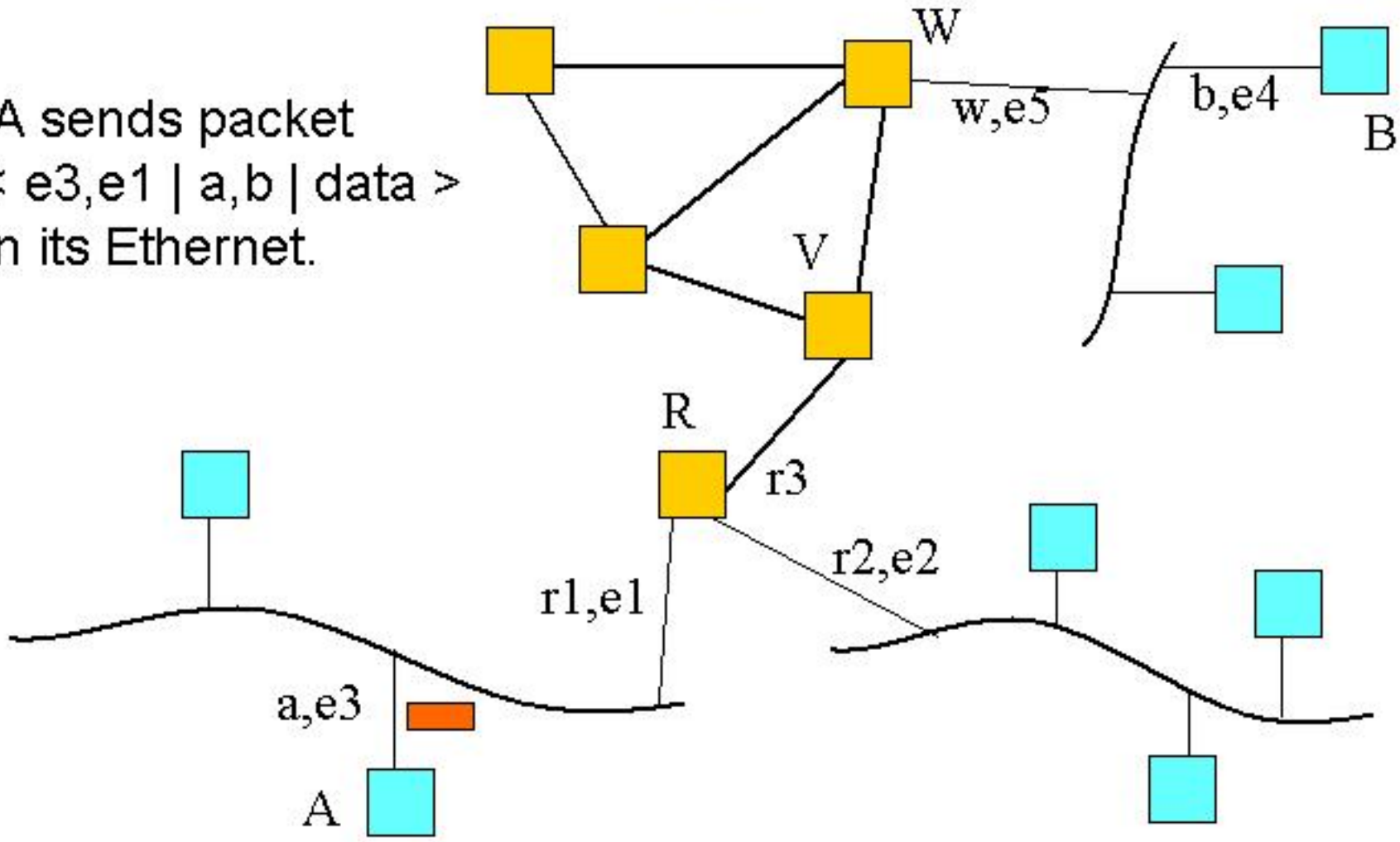
A small Internet

2. Does b belong to my LAN ?
- if yes, get its LAN (Ethernet) address;
- otherwise, send to default router (known).



A small Internet

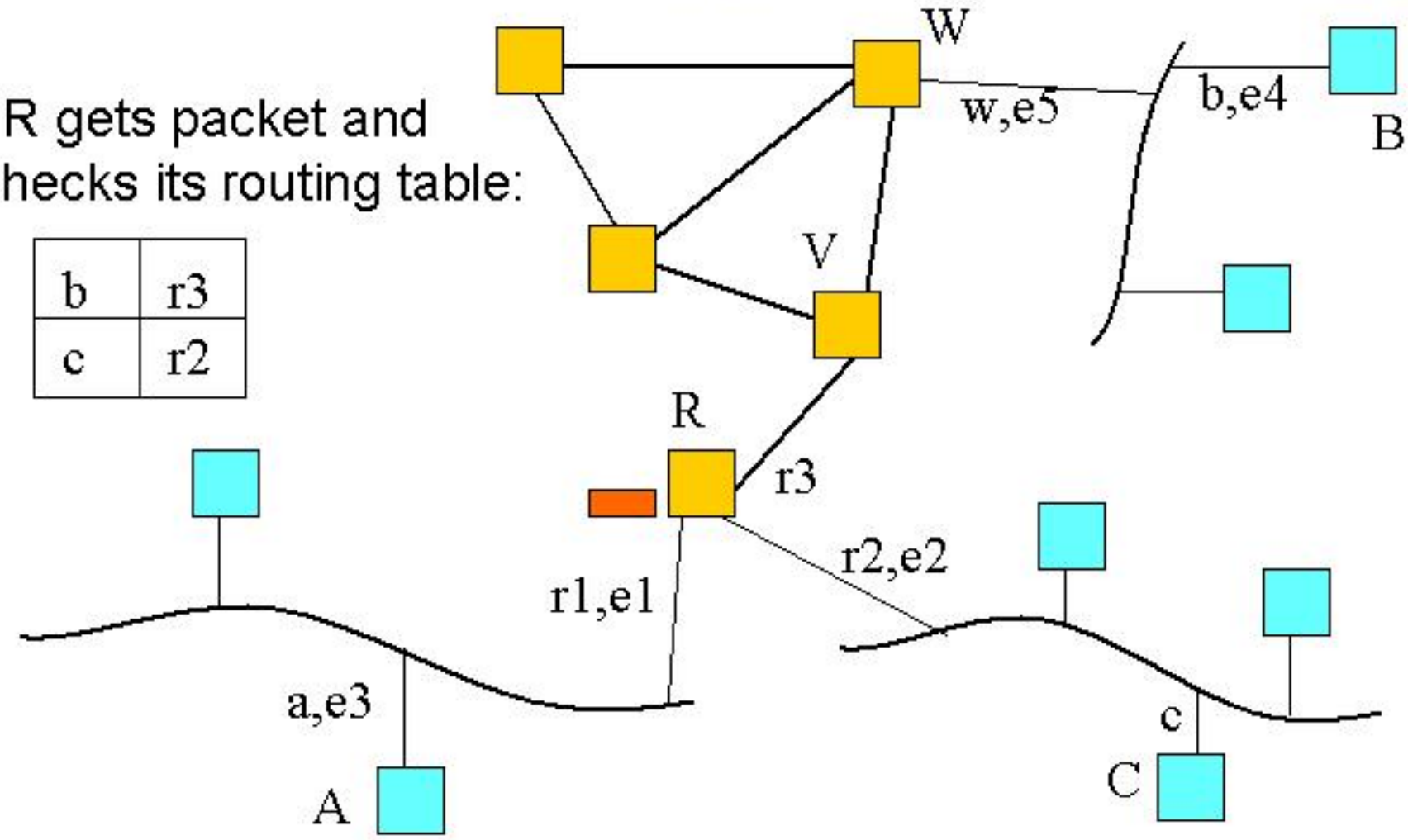
3. A sends packet
   < e3,e1 | a,b | data >
   on its Ethernet.



A small Internet

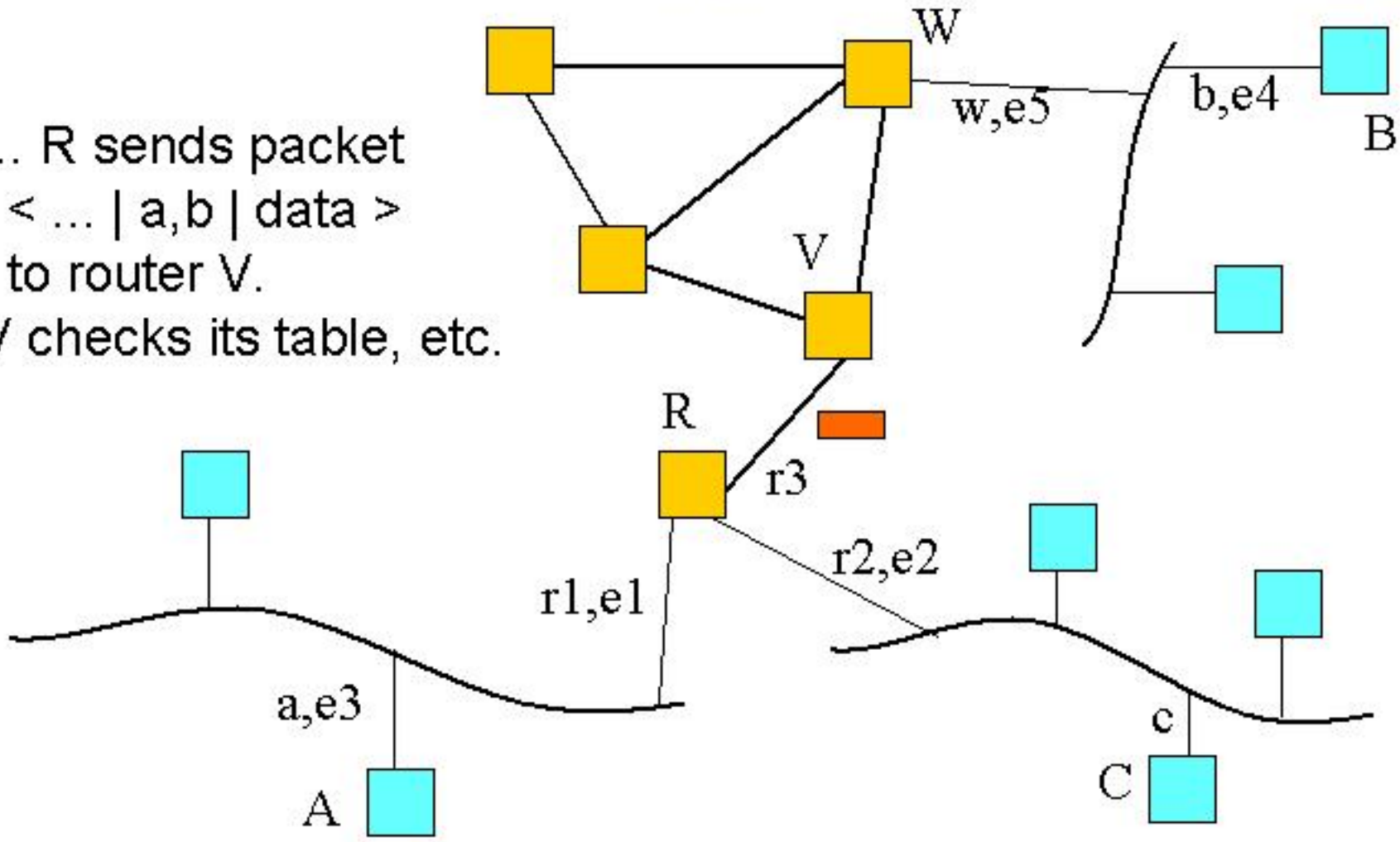
4. R gets packet and checks its routing table:

b	r3
c	r2



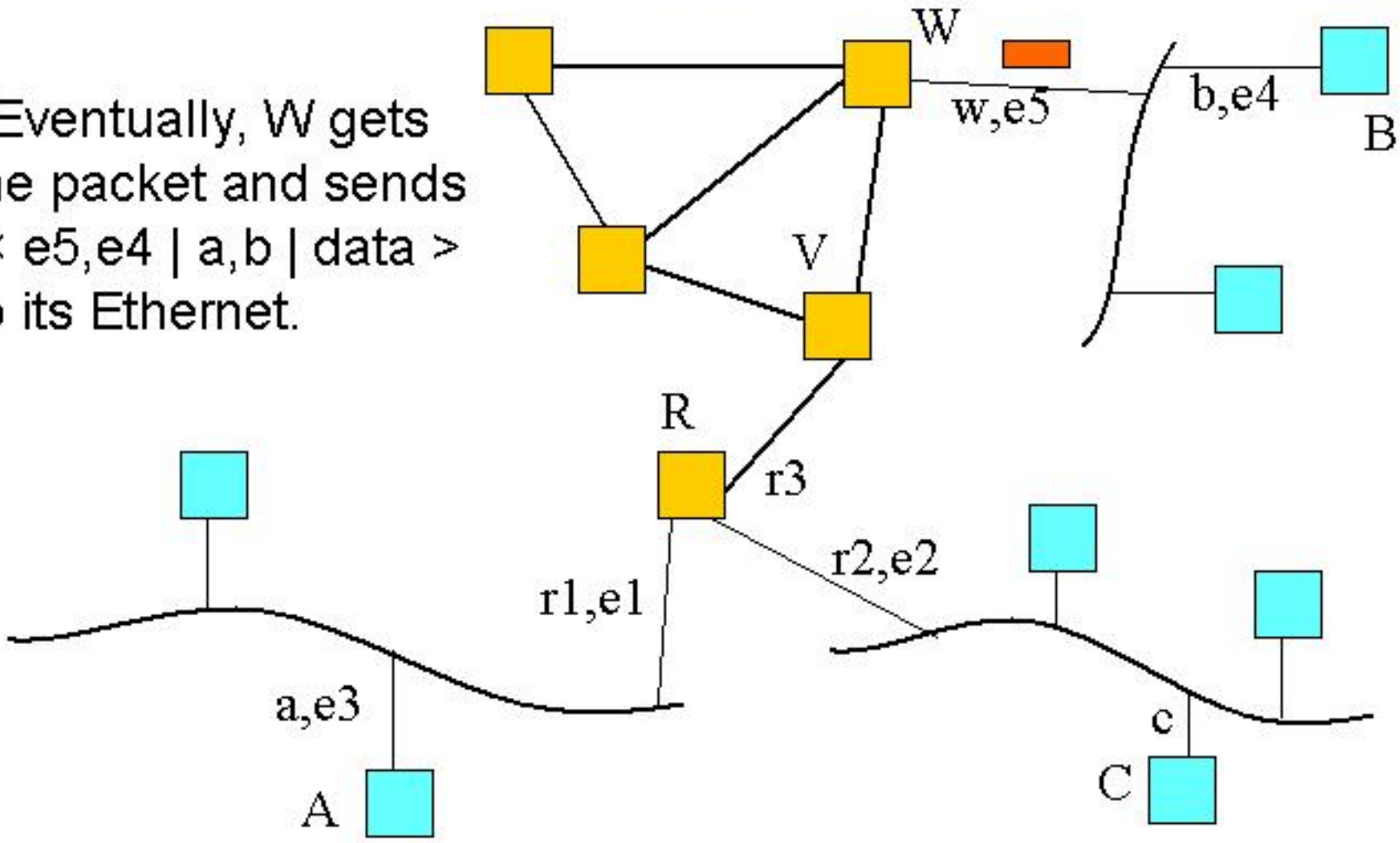
A small Internet

5-... R sends packet
< ... | a,b | data >
to router V.
V checks its table, etc.



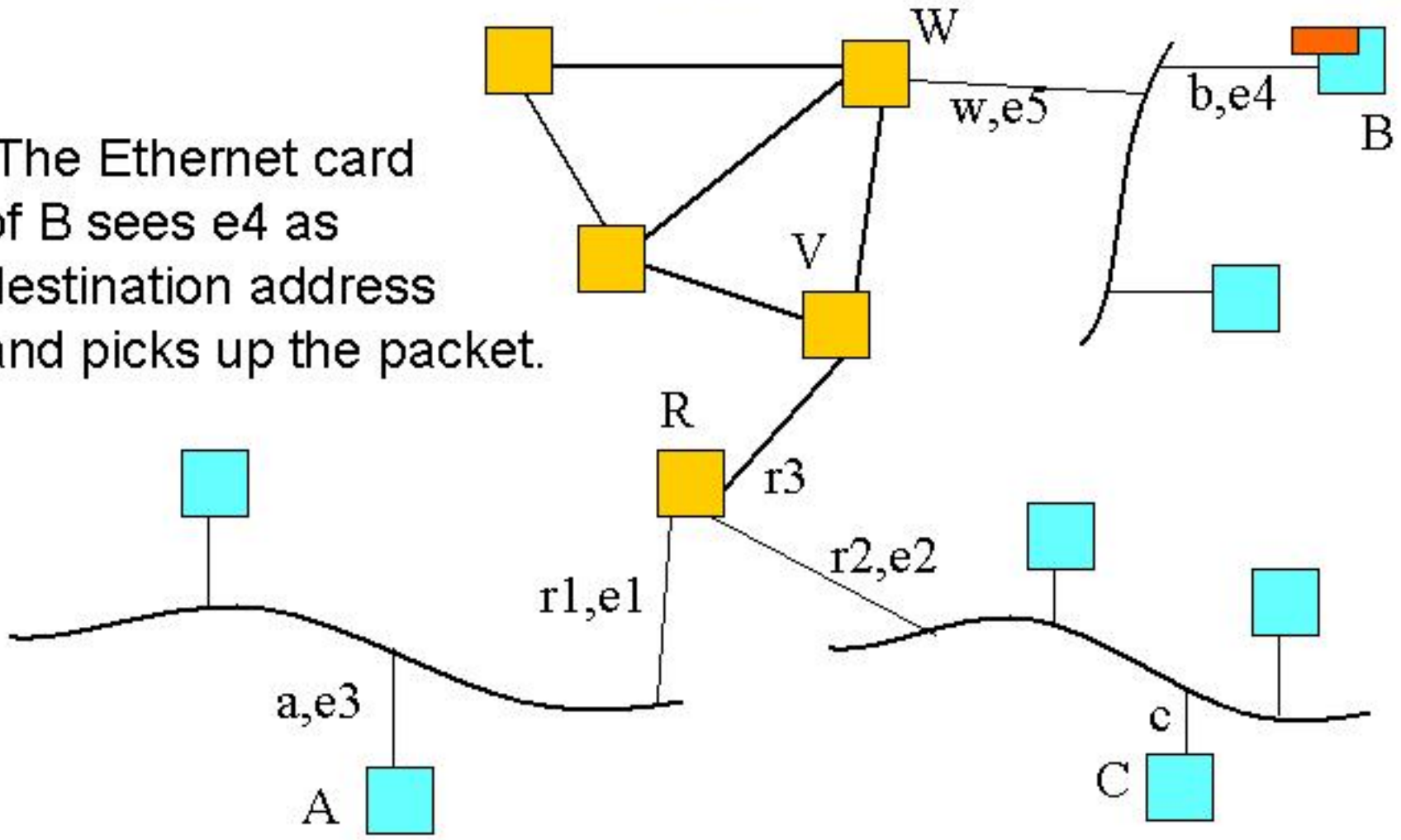
A small Internet

6. Eventually, W gets the packet and sends < e5,e4 | a,b | data > to its Ethernet.



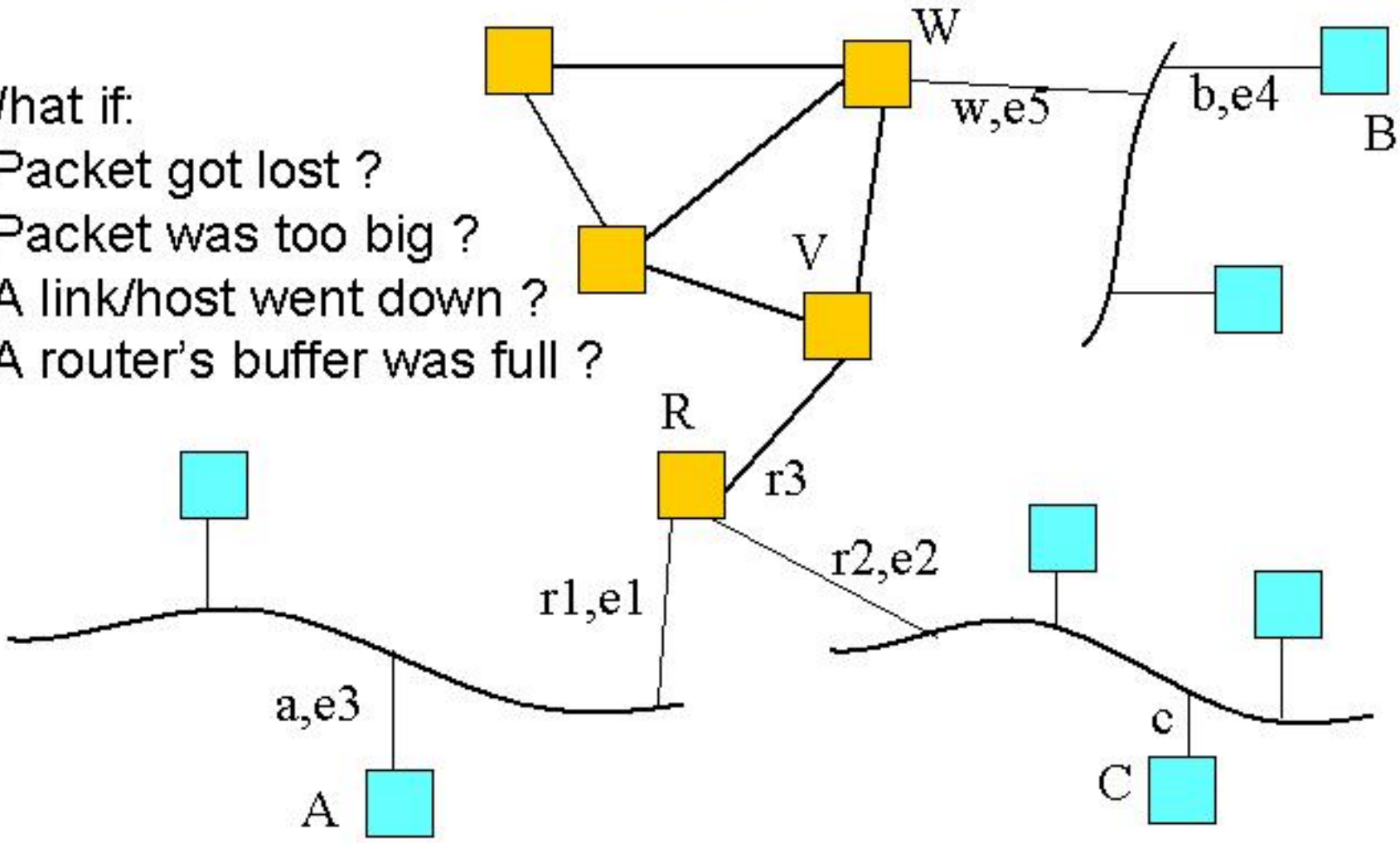
A small Internet

7. The Ethernet card of B sees e4 as destination address and picks up the packet.



A small Internet

What if:
- Packet got lost ?
- Packet was too big ?
- A link/host went down ?
- A router's buffer was full ?



IP (Internet Protocol)

- Network-layer protocol: unreliable **end-to-end** delivery of packets (**datagrams**) of up to 64 kbytes.
- End-to-end: source and destination might not be directly connected, routing involved.
- Unreliable: packets might be lost (corrupted at the physical level or dropped because of full buffers) or not be delivered successfully (destination unreachable, loops). **ICMP** informs the source in the latter case.

IP addresses

- 32-bit addresses.
- Hosts have IP addresses, e.g., 128.32.239.44.
- Networks have addresses too:

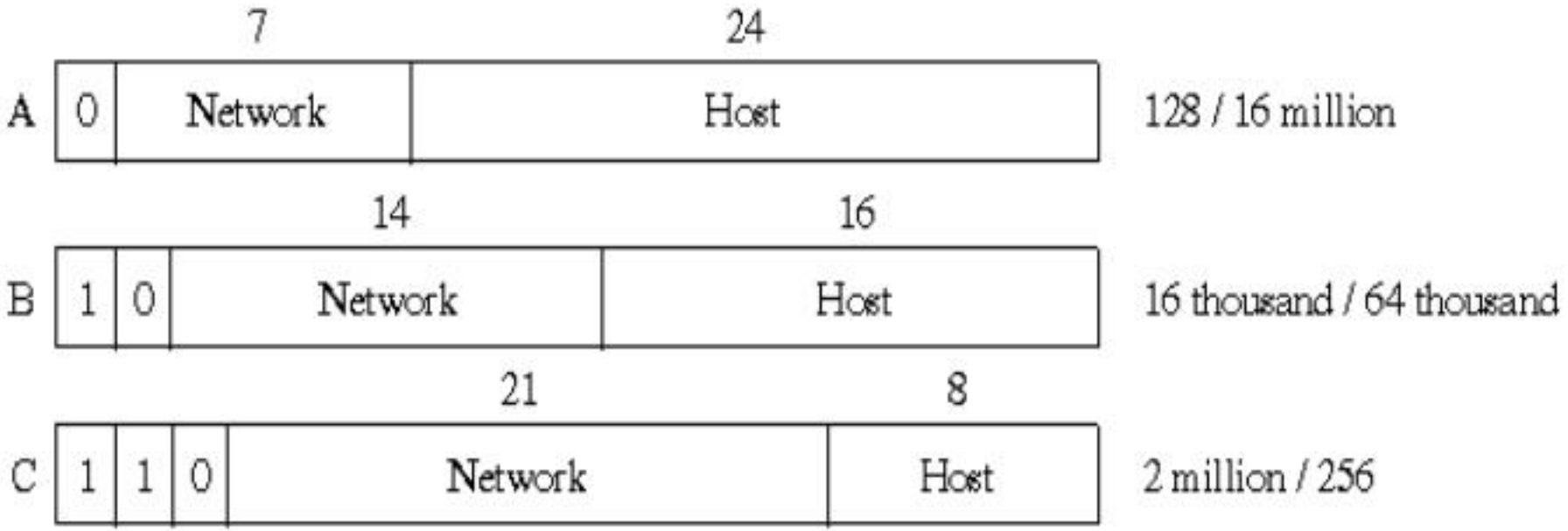
  (IP address, Mask).
- A host X knows its own mask, M.
- To check whether host Y belongs to the same network than X, check if: $X \otimes M = Y \otimes M$.

  ($\otimes$: logical and).

IP addressing

- **Hierarchical,** based on geographical location, like telephone numbers: +1 510 642 5649
  - Scalability in **routing**.
  - Scalability in assigning addresses.
- 3 stages:
  - class-based
  - subnetting
  - classless

Class-based addressing



E.g., UC Berkeley network address:
128.32 = 10000000.00100000 (class B network).

Class-based addressing: problems

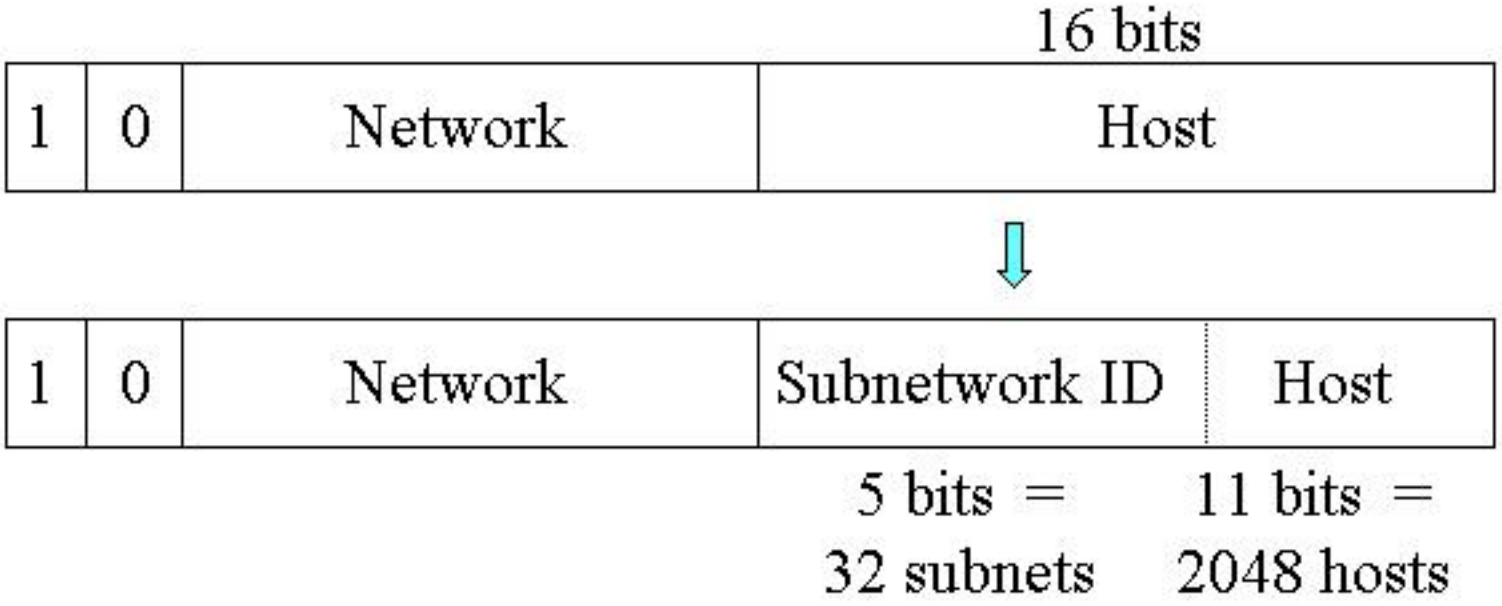
- Class A networks: too big, too few.
- Class C networks: too small, too many.
- Class B networks: not enough (run out of already).
- However:
  - 128 x 16 million + 16,000 x 64,000 + 2 million x 256
    ≈ 4 billion !!!

  ⇒ should have enough addresses for everybody.
- Problem: bad utilization.
- Question: how to have an addressing scheme that meets exactly the needs of the users ?

Subnetting

- Idea: 10 depts. x 2000 hosts per dept.

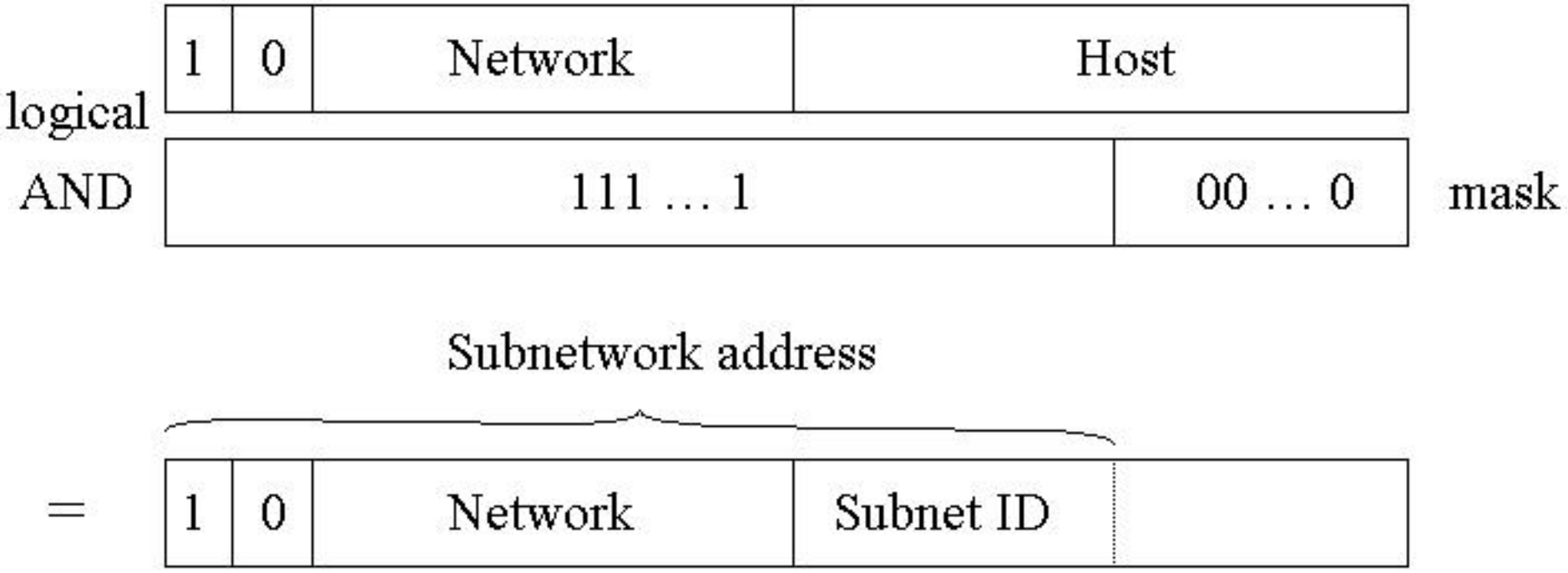
  = 1 class B network address (instead of 10)



Subnetting

- Implementation:

Subnetwork address = IP address AND mask



Subnetting: problem

- Relatively small networks (between 256 and 1000 hosts) still need at least a class B network address. (In fact, so does any network that could prospectively have more than 255 hosts.)

Classless addressing (supernetting)

- Idea: aggregate many class C networks under a common network address, e.g., get 16 consecutive class C addresses: 192.4.16-31 = 16 x 256 = 4096 hosts.
- All addresses share common **prefix**: first 20 bits:

  11000000.00000100.0001xxxx
- This prefix is the "supernet" address (it defies class boundaries, something between class B and class C network).

Classless addressing (supernetting)

- Generalization of subnetting:
  - Subnetting: split class-based address to multiple subnet addresses.
  - Supernetting: also allow aggregation of multiple class-based addresses into a supernet address.
- Current Internet routing protocols use subnetting and supernetting (CIDR):
  - View a collection of subnets as a single IP address.
  - View a collection of IP addresses as a single supernet address.

Packet-forwarding in LANs

- Assume hosts A and B are on the same network (as found by comparing the IP addresses logically-and-ed with the mask).
  - Is B in the same LAN as A (e.g., Ethernet, FDDI) or is there a point-to-point link between A and B ?
  - If yes, what is the LAN/link interface address ?
  - If not, send to (default) router.
- Address Resolution Protocol (ARP).

Packet-forwarding in LANs

- ARP table (or cache):

IP address	LAN address
C	x, Ethernet
D	y, Ethernet
E	z, FDDI
...	

- Basic operations:
  - If A doesn't have an entry for B, it **broadcasts** message "B, are you on my LAN? If yes, give me your interface address".
  - If B is in the LAN, it replies, and A adds an entry in its ARP cache.

Packet-forwarding in LANs

- ARP table (or cache):

IP address	LAN address
C	x, Ethernet
D	y, Ethernet
E	z, FDDI
...	

- Notes:
  - LAN has to support broadcasting (special broadcast address used).
  - Point-to-point link addresses statically configured.

Routing

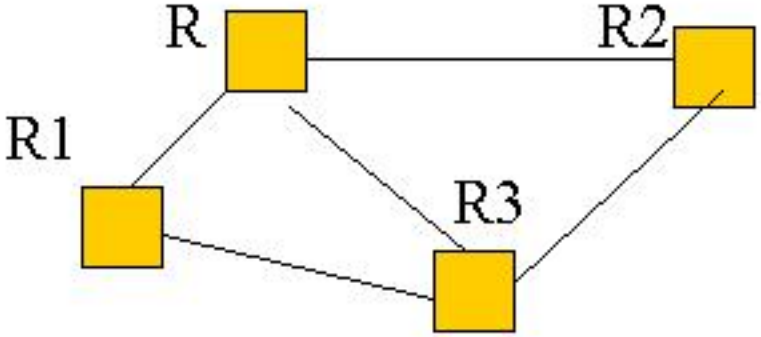
- If A and B are not on the same LAN, a packet from A to B has to be routed.
- The Internet is a network of **heterogeneous** networks:
  - using different technologies;
  - belonging to different administrative authorities.
- Goal of routing: interconnect all these networks.
- Routers, switches, bridges.
- Routing protocols.

Routing requirements

- Scalability.
- Robustness to network changes (link failures, full buffers, etc).
- Efficiency: "good" routes.
- Security, administration issues.

Internet routing: hierarchical

- Routers only know about networks, not hosts.
- Routing table at R:



Destination Network	Next hop
Net1	R1
Net2	R2
Net3	R3

- Network addresses = pairs (IP address, mask)
  - E.g., (128.96.34.0, 255.255.255.128)
  - Can encode class-based, subnetting, supernetting.

Internet routing: hierarchical

- Class-based:
  - class A, e.g., network 4 has pair (4.0.0.0, 255.0.0.0)
  - class B, e.g., network 128.29: (128.29.0.0, 255.255,0,0)
  - class C, e.g., net 192.44.3: (192.44.3.0, 255.255.255.0)
- Subnetting:
  - E.g., Berkeley class B address: 128.29 and EECS subnet address: 128.29.4: (128.29.3, 255.255.255.0)
- Supernetting:
  - Supernet of 16 class C networks: 192.44.16-31 has pair: (192.44.16.0, 255.255.240.0)

Internet routing: hierarchical

- When a packet with destination IP address X arrives at the router:
  - the router goes over the entries in its routing table
  - for each entry ((Y, M), Z), it checks whether:

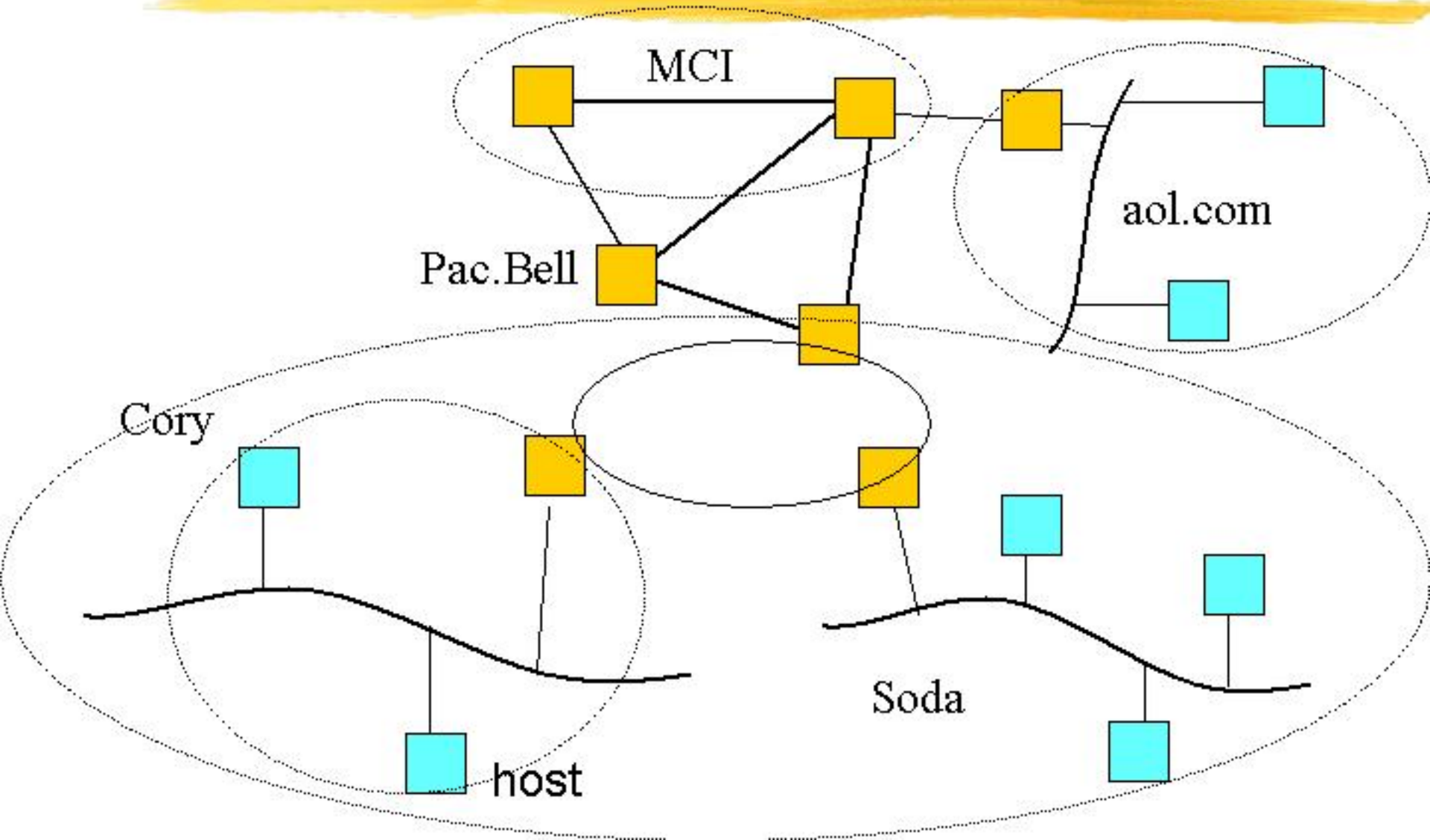
    $$X \otimes M = Y$$

  - if so, then Y is the (sub-/super-) network where X belongs, and the packet is forwarded to Z.
- What if multiple entries match ?
  - Pick the one with the longest match.

Internet routing: hierarchical

- **Autonomous system** (AS): a set of networks subject to a common authority. E.g.,
  - Berkeley campus network.
  - Company network.
  - Regional network (1990), ISP network (today).
  - NSFNET backbone (1990), one of the corporation backbones (today).

A small Internet: ASs



traceroute to ormelune.imag.fr (129.88.43.35): 1-30 hops, 38 byte packets

```
 1  cnr239net.EECS.Berkeley.EDU (128.32.239.1)  0.737 ms  0.503 ms  0.405 ms
 2  fast4-0-0.inr-110-cory.Berkeley.EDU (169.229.1.41)  0.871 ms  0.796 ms  0.752 ms
 3  f4-0.inr-107-eva.Berkeley.EDU (128.32.120.107)  1.49 ms  1.56 ms  1.8 ms
 4  f8-0.inr-666-eva.Berkeley.EDU (128.32.2.2)  2.63 ms  2.70 ms  2.3 ms
 5  fast0-0-0.inr-002-eva.Berkeley.EDU (128.32.0.82)  1.70 ms  2.8 ms  1.57 ms
 6  pos0-2.inr-000-eva.Berkeley.EDU (128.32.0.73)  2.24 ms  1.45 ms  1.59 ms
 7  c2-berk-gsr.calren2.net (128.32.0.90)  1.93 ms  1.58 ms  1.57 ms
 8  Abilene-BERK.POS.calren2.net (198.32.249.42)  5.19 ms  4.71 ms  5.37 ms
 9  denv-scrm.abilene.ucaid.edu (198.32.8.2)  26.9 ms  27.2 ms  26.3 ms
10  kscy-denv.abilene.ucaid.edu (198.32.8.14)  36.6 ms  36.8 ms  37.0 ms
11  ipls-kscy.abilene.ucaid.edu (198.32.8.6)  45.6 ms  47.2 ms  46.4 ms
12  chicago-atm40.1.opentransit.net (193.251.128.169)  151 ms  51.0 ms  50.5 ms
13  bagnolet1-atm30.5.opentransit.net (193.55.152.77)  153 ms  153 ms  152 ms
14  bagnolet2-fddi000.opentransit.net (193.55.152.178)  153 ms  153 ms  153 ms
15  rbs2.renater.ft.net (195.220.180.30)  156 ms  154 ms  153 ms
16  stamand1.renater.ft.net (195.220.180.17)  154 ms  154 ms  154 ms
17  grenoble.renater.ft.net (195.220.180.6)  166 ms  166 ms  166 ms
18  194.199.224.113 (194.199.224.113)  180 ms  207 ms  216 ms
19  194.199.224.122 (194.199.224.122)  210 ms  179 ms  168 ms
20  aramis.grenet.fr (193.54.184.1)  170 ms  167 ms  167 ms
21  r-ujf.grenet.fr (193.54.185.124)  170 ms  168 ms  168 ms
22  bio-gate.ujf-grenoble.fr (193.54.238.9)  169 ms  169 ms  170 ms
23  ormelune.imag.fr (129.88.43.35)  171 ms  *  172 ms
```

Internet routing: hierarchical

- *Intradomain routing* : routing inside an AS.
  - Distance-vector routing (Bellman-Ford's shortest-path algorithm, RIP protocol).
  - Link-state routing (Dijkstra's shortest-path algorithm, OSPF protocol).
- *Interdomain routing* : routing across many ASs.
  - BGP (border gateway protocol).

Intradomain routing

- Network (routing domain) viewed as a **weighted graph**, where:
  - nodes are routers;
  - an edge (R1, R2) means routers R1 and R2 are connected physically (e.g., by point-to-point link, or on the same LAN);
  - the weight of an edge corresponds to a **metric** (latency, capacity, loss probability).

Distance-vector routing (Bellman-Ford's shortest-path)

- Used in (old) RIP (routing information protocol), BSD public distribution of TCP/IP.
- Bellman-Ford algorithm: given a weighted graph and a destination node D, find the shortest path from each node in the graph to D.
- Routers exchange **distance-vectors** to **neighbor** routers, e.g., (R1:5, R2:3, R3:7).
- Update routing table based on received distance vectors.

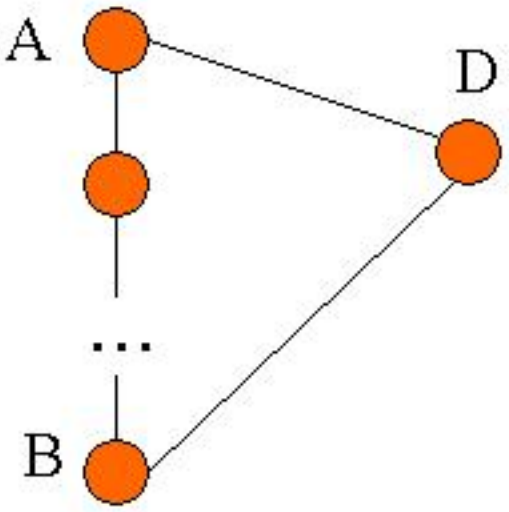
Distance-vector routing: problems

- Convergence is slow.
- Loops can be formed, due to routing table inconsistency: packets being forwarded from router to router and never reach the destination.
- Loops might last for a long time:
  - until convergence, or
  - count to infinity problem.

Link-state routing (Dijkstra's shortest-path)

- Used in (current) OSPF (open shortest path first) protocol, by IETF.
- Dijkstra's algorithm: given a weighted graph and a source node A, find the shortest path from A to each other node in the graph.
- Routers send **link-state** packets (R1,R2,7), to **all other** routers in the same routing domain (**flooding**).
- Each router learns the current state of the whole network and runs Dijkstra's algorithm to build its own routing tables.

Link-state routing: loops



- Links A-D and B-D fail simultaneously.
- A updates its route to D through B and sends LSP to B, saying A-D is down.
- B updates its route to D through A and sends LSP to A, saying B-D is down.
- Until the LSPs are received, there is a loop between A and B for packets to D.
- Loop is **transient**: disappears when one of the LSPs is received.

Link-state vs. distance-vector

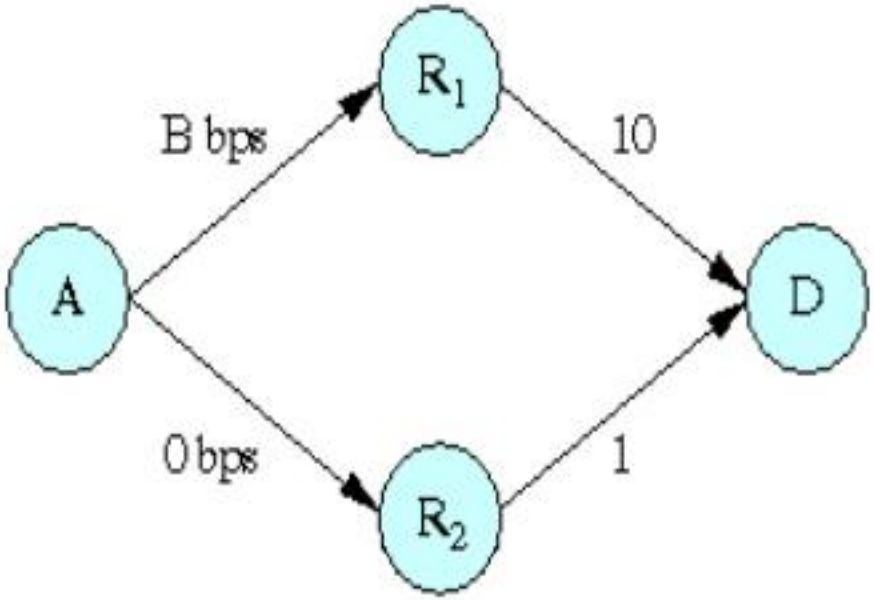
- Experience has shown OSPF to be better than RIP in **stability** (robustness to network changes):
  - distance-vector converges very slowly, loops can last for long periods of time.
  - link-state converges very quickly, loops are transient.
- RIP is distributed, whereas OSPF is centralized (flooding).
  - OSPF creates more routing traffic (flooding).

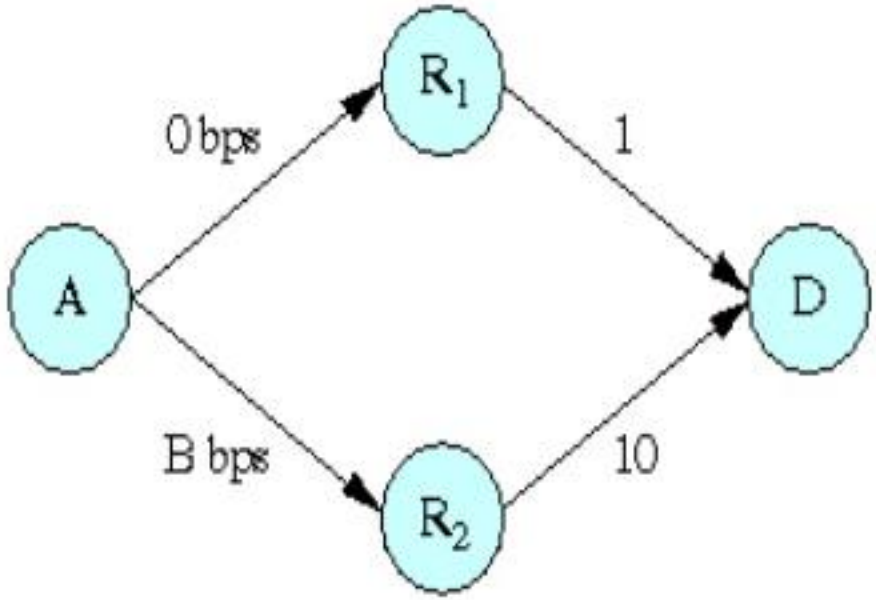
Routing metrics

- How are link weights determined ?
  - Simple: weight 1 to all links (cost of path = hop count).
  - Link latency (queuing + propagation delays).
  - Link capacity (bit-rate).
  - Link reliability (packet-loss probability).
- Routing protocols might use multiple metrics and compute multiple routing tables for different traffic requirements (OSPF does this).
- Problem: link state changes dynamically.

Routing metrics: example of routing oscillations

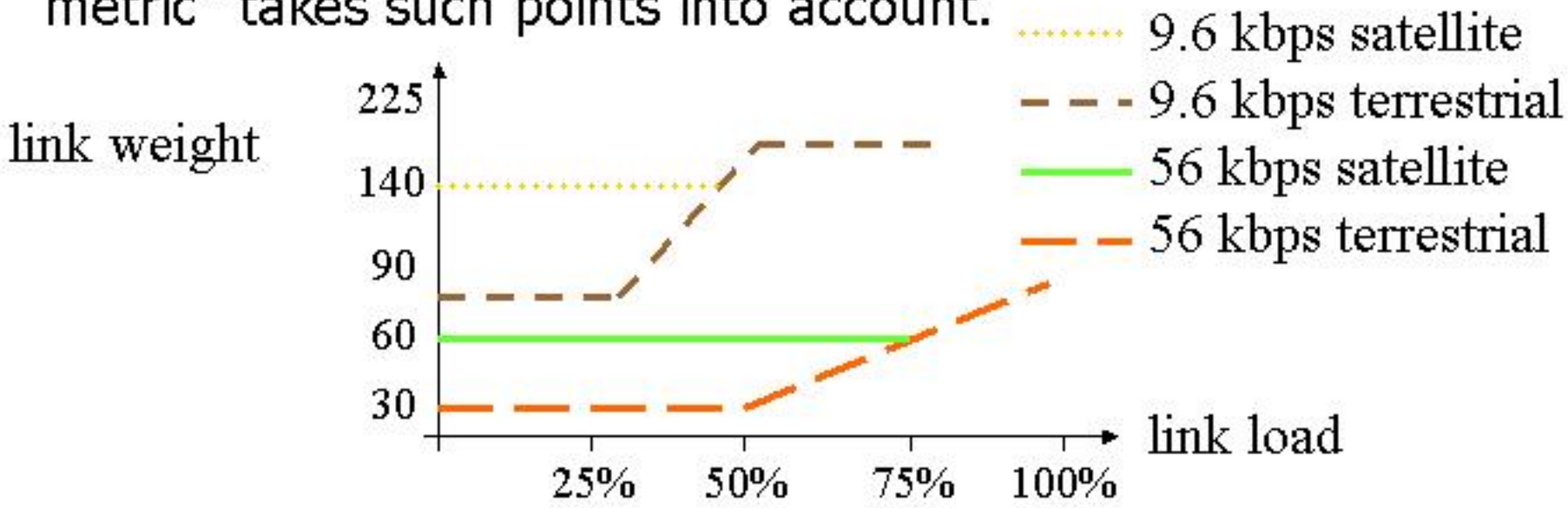
- Routing affects link load.
- Link load affects link weight (latency, available capacity, etc).





Routing metrics

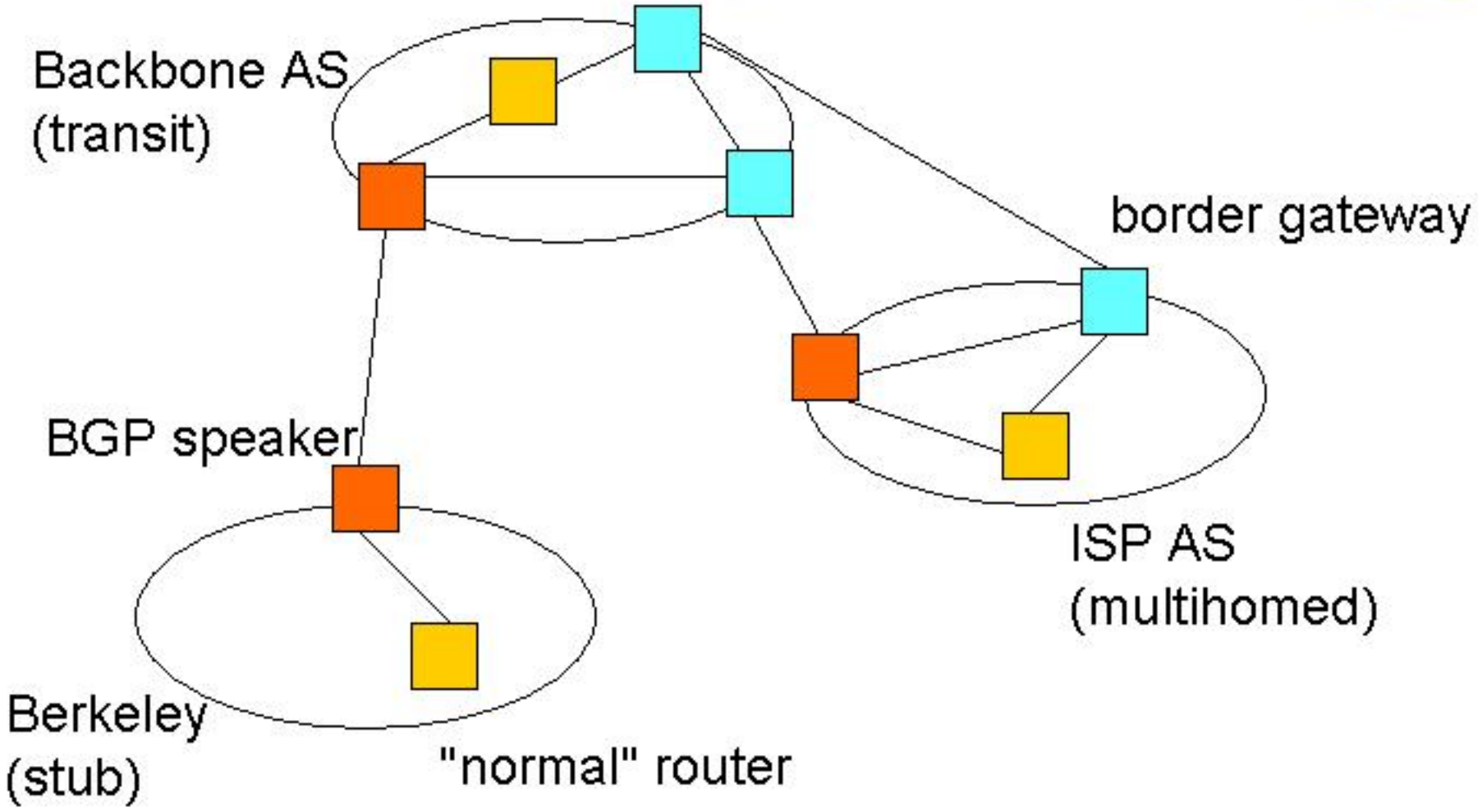
- To avoid such oscillations:
  - "Normalize" different routing metrics.
  - Take into account link type (e.g., satellite, terrestrial).
  - Smoothen the variation of metric in time.
- (After many experiments) "Revised ARPANET routing metric" takes such points into account.



Interdomain routing

- At the IP (network) level.
- Network: set of AS connected by **border gateways** (routers connecting one AS to another).
- AS classification:
  - *stub* AS: single connection to another AS, local traffic;
  - *multihomed* AS: multiple connections, local traffic;
  - *transit* AS: multiple connections, transit traffic.
- One border gateway from each AS is the BGP **speaker**.

Interdomain routing



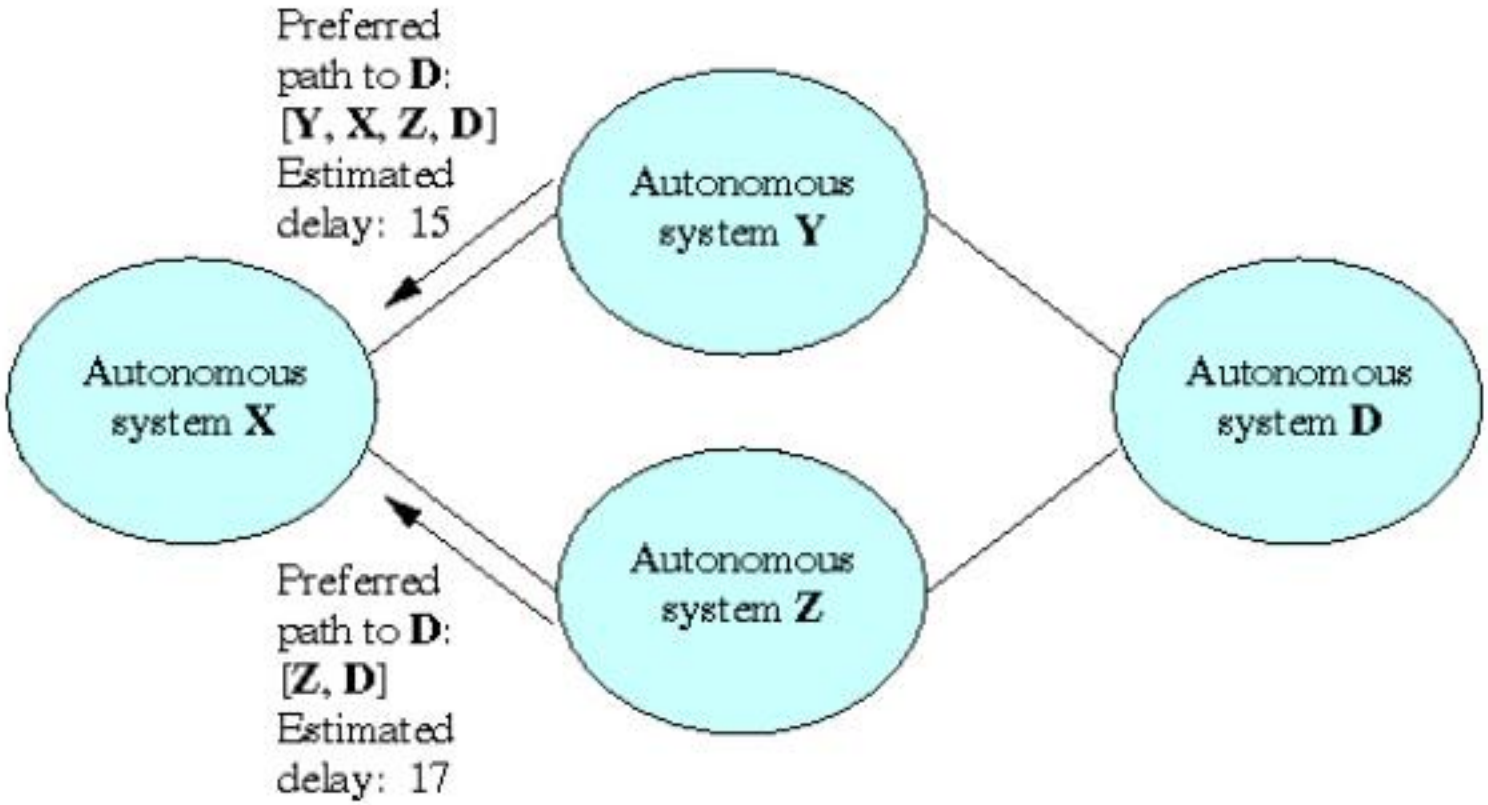
Interdomain routing: BGP

- BGP (border gateway protocol) is executed among BGP speakers.
- BGP speakers then distribute routing information to other border gateways inside their AS.
- Why not execute OSPF instead of BGP ?
  - Too many ASs (50,000): flooding becomes too expensive.
  - Administrative reasons: security, economics (cf. stub vs. transit ASs).
  - Metrics not always consistent among ASs.

Interdomain routing: BGP

- BGP speakers advertise **preferred** paths (routes).
- BGP is distributed: each speaker has its own view of the network (might be inconsistent w.r.t. the view of other speakers).
- To avoid loops, advertise **complete** paths, e.g., AS X says: "My preferred path to AS Z is (X, Y, W, Z)".
- When Y hears this, it knows it shouldn't go through X to get to Z.

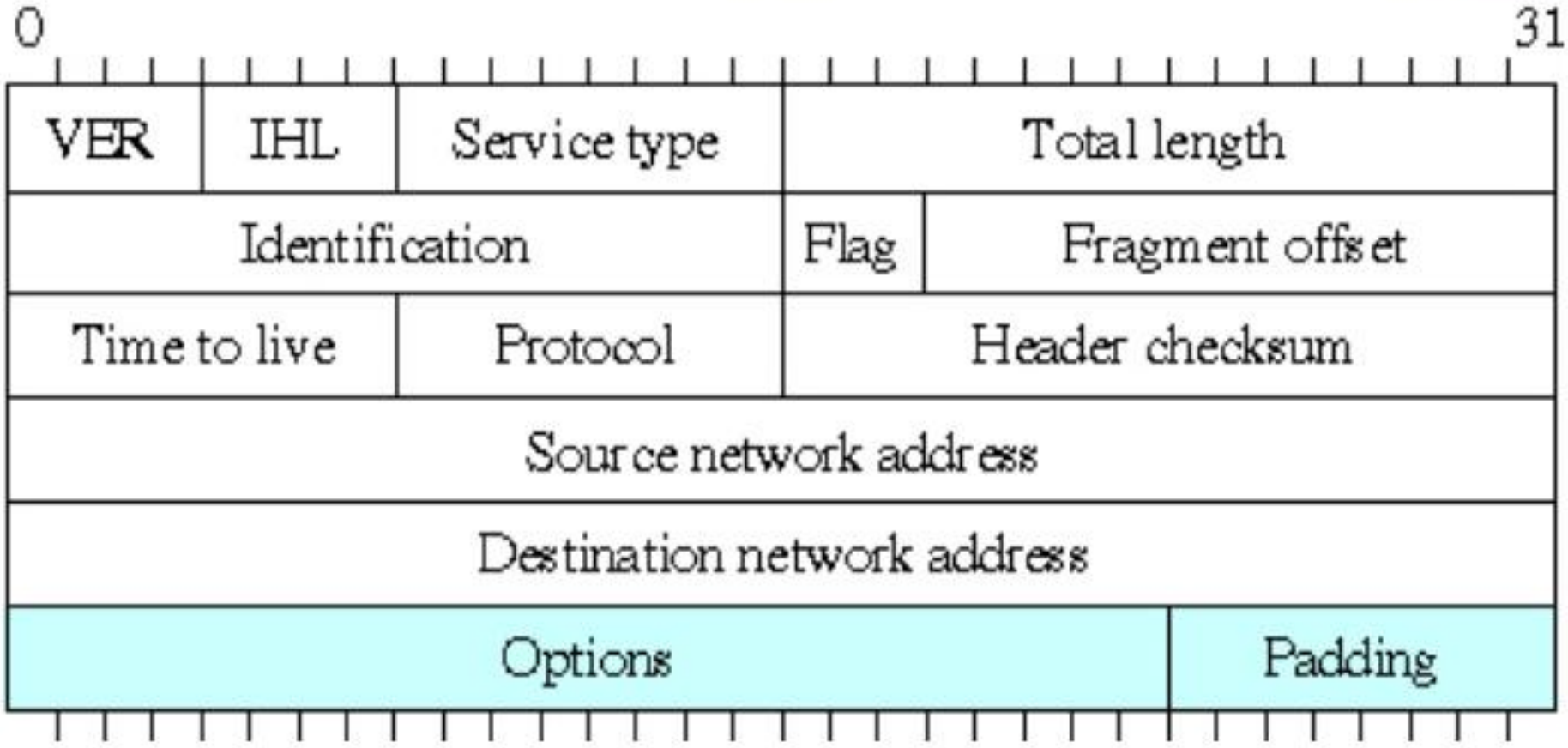
BGP: example


Preferred path to D: [Y, X, Z, D] Estimated delay: 15
Autonomous system Y
Autonomous system X
Autonomous system D
Autonomous system Z
Preferred path to D: [Z, D] Estimated delay: 17

Modern internetworking

- Mobility:
  - Plug-and-play: allowing hosts to dynamically/temporarily connect to the Internet, in a particular domain. Dynamic Host Configuration Protocol (DHCP).
  - Allowing hosts to move between multiple domains. Mobile IP.
- Multicasting: sending packets to multiple hosts.
  - Simple (inefficient) solutions: flooding, multiple transmissions.
  - Spanning tree, Multicast OSPF, PIM, etc.

IP header format



IP fragmentation/reassembly

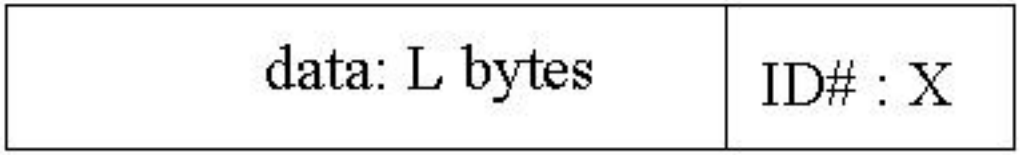
- Different link layers have different **Maximum Transfer Units** (MTUs): maximum size of packet they can transmit.
- When *Size(datagram) > MTU* the datagram needs to be split in many pieces: **fragmentation**.
- The pieces are joined together to form the original datagram at the destination: **reassembly**.
- A datagram may be fragmented multiple times along its route to the destination.

IP fragmentation/reassembly

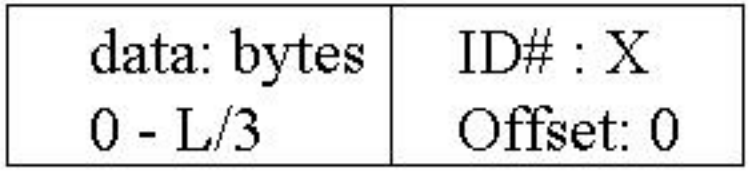
- ID# used to identify the datagram: all fragments of the same datagram have same ID#.
- Offset used to mark the "starting position" of the fragment from the beginning of the datagram (in number of bytes).

IP fragmentation/reassembly

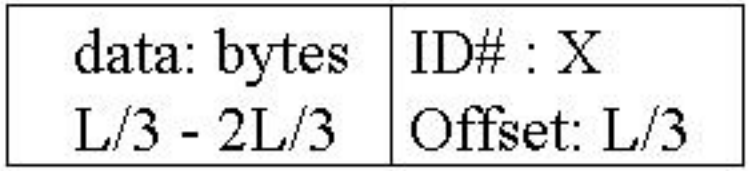
- Example:

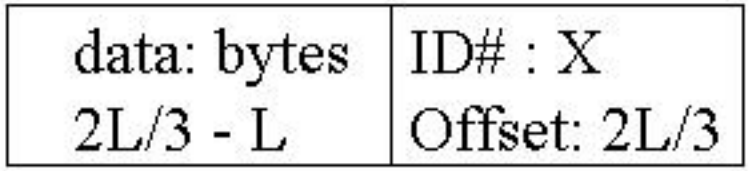


original datagram



MTU ≈ L/3





LANs (local area networks)

- Building-blocks: physical layer.
- Different architectures:
  - Ethernet: multiple access (CSMA/CD).
  - Wireless (IEEE 802.11): multiple access (CSMA/CA).
  - FDDI, IEEE 802.4, 802.5: logical token-rings.
  - Point-to-point links.
- Different properties (efficiency, medium access time guarantees, etc.)

FDDI analysis

- Can prove that for each node i :

  TRTi < 2 TTRT.

  This implies that the medium access timeis at most 2 TTRT.

  Note book error: assumption 2TTRT < ρ to be replaced by TTRT > ρ.

- Efficiency: close to 100%.

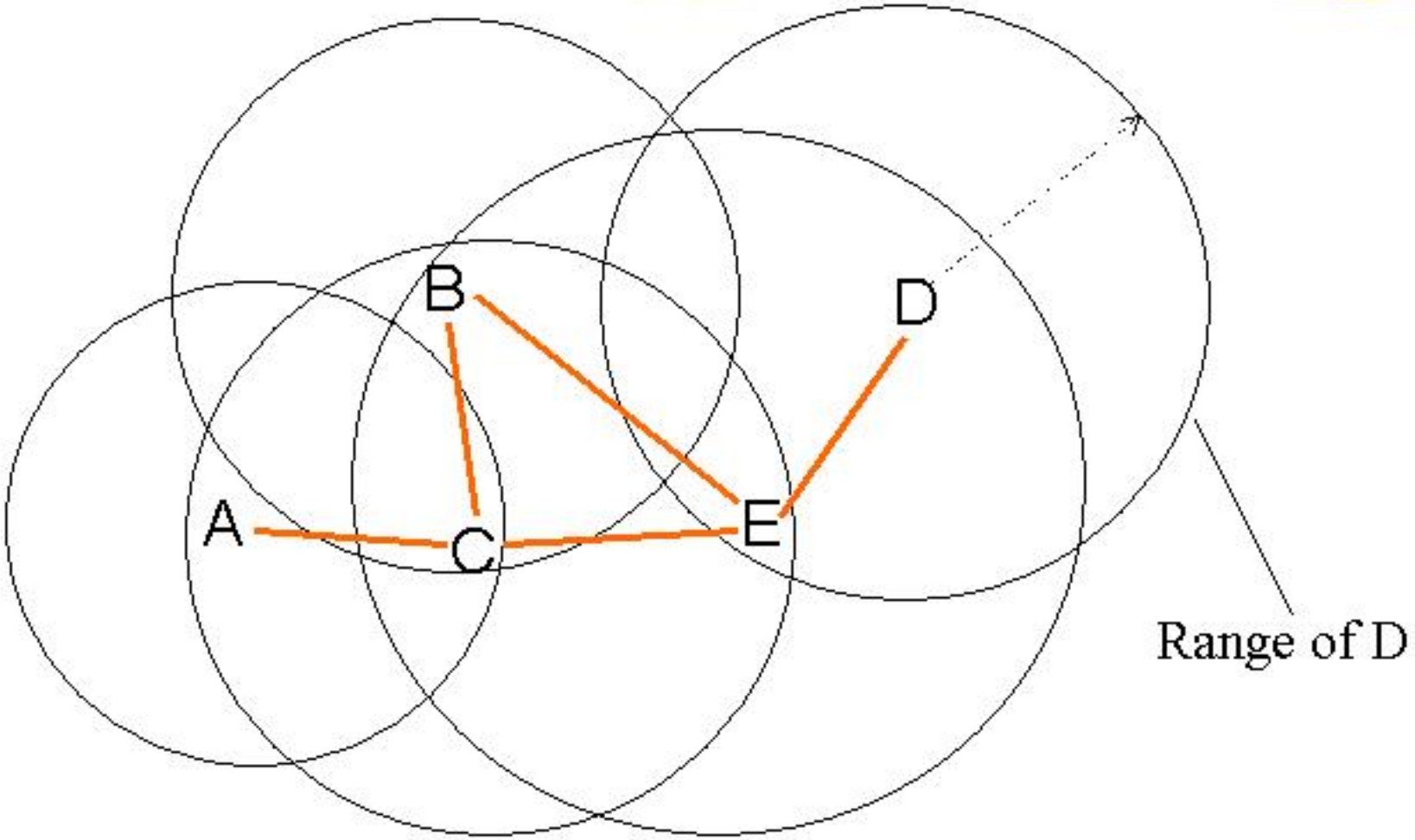
Wireless LANs

- Unique features of wireless networks.
- Emerging standards:
  - Europe: ETSI Hiperlan
  - US: IEEE 802.11
- Emerging products:
  - WaveLan (Lucent)
  - Metricom
  - Ricochet
  - Nokia
  - etc.

Characteristics of wireless networks

- Medium: 3D space.
- Signals: radio waves on specific frequencies.
  - Frequency is a precious resource.
- "Difficult" medium:
  - Interference, noise, shadowing, multipath effect.
  - Higher bit-error rates, lower capacity (1-2Mbps).
- Power limitations $\Rightarrow$ non-broadcast medium.
  - Carrier-sense not very helpful.
  - Collision-detection would require full-duplex radio channel $\Rightarrow$ too expensive.
- Mobility of hosts.

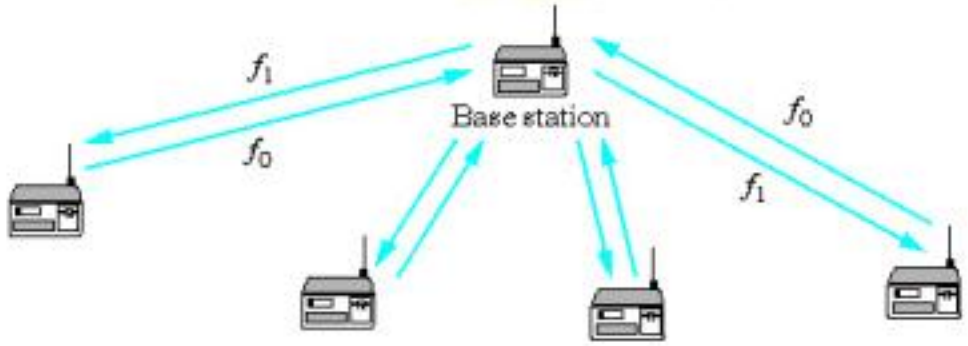
Simplified view of wireless network



Difficulties for carrier-sense and collision-detection.

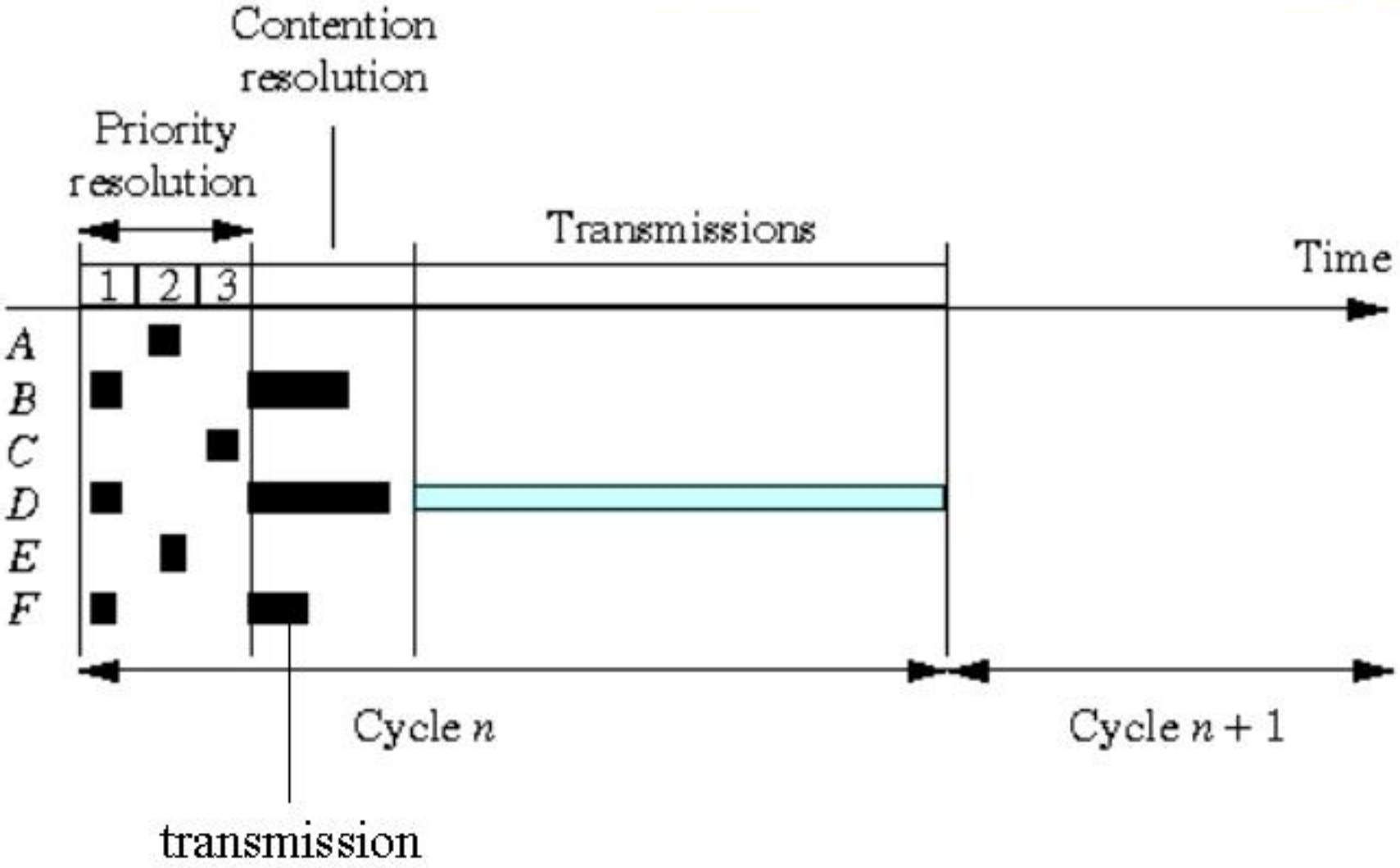
- *Hidden-terminal problem* : A cannot "hear" C transmitting. If A wants to transmit to B and hears nothing, it cannot assume that collision won't happen, so carrier-sense does not help. Collision-detection not possible at sender's side.
- *Exposed-terminal problem* : C can hear B transmitting but does not know who the receiver is. C can still transmit to D while B transmits (to A or some other node) without the two signals interfering at the receivers' sides.

ALOHA protocol



- Multiple access without carrier-sense or collision-detection (two terminals cannot hear each other).
- Two versions:
  - Pure ALOHA: a station can transmit at any time.
  - Slotted ALOHA: time divided into slots (each slot is enough for 1 packet), stations can transmit only at the beginning of a slot. Better performance, harder to implement (need to synchronize clocks).

ETSI Hiperlan standard

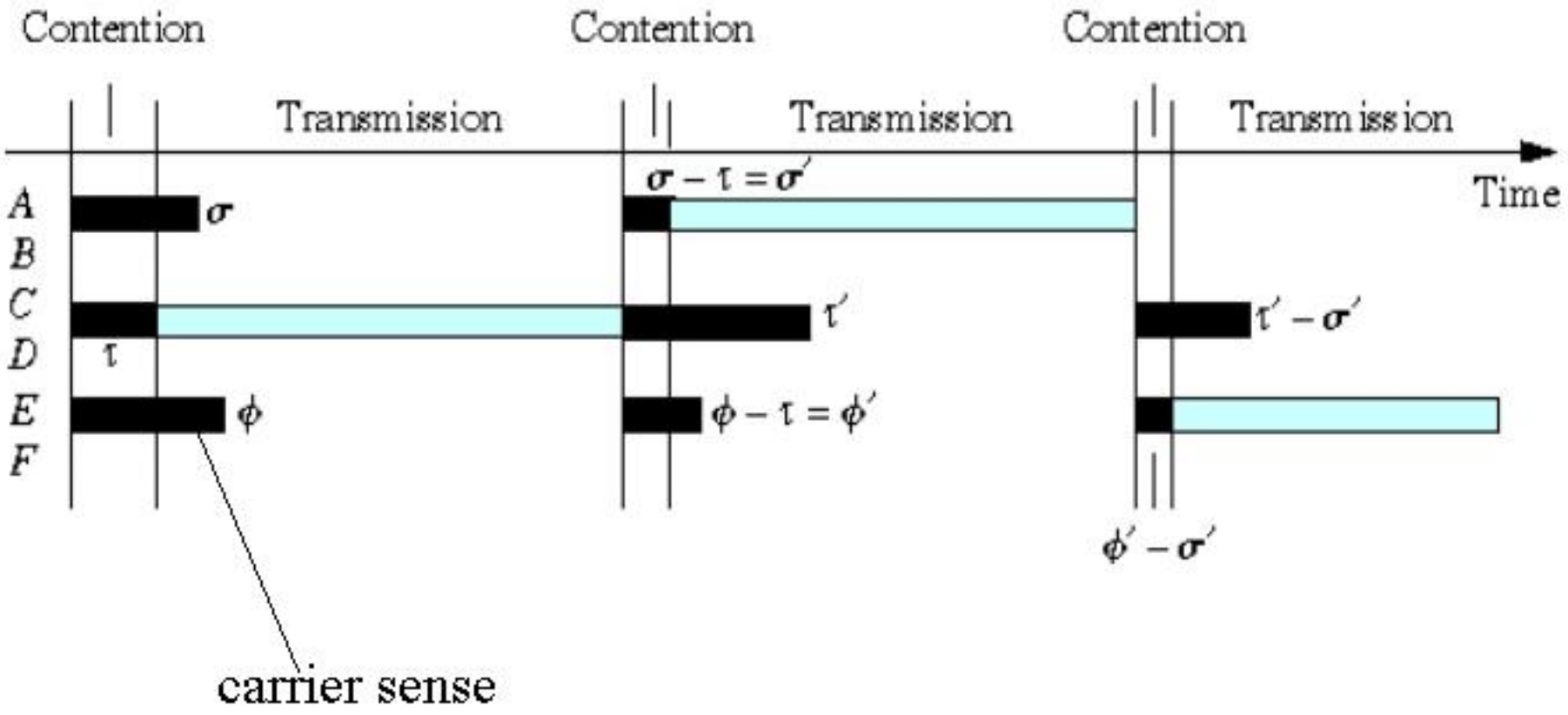


IEEE 802.11 standard

- Physical and MAC specifications.
- "Open" standard, leaves many possibilities for implementation ⇒ not clear whether different 802.11-compliant devices can inter-operate.
- Physical:
  - Unlicensed bands, e.g., in US: 900MHz, 2.4GHz, 5.7GHz.
  - Various restrictions on use.
  - Direct-sequence or Frequency-hopping spread-spectrum.
- MAC: different modes of operation:
  - Centralized: a base station gives access to the nodes one-by-one.
  - Distributed: CSMA/CA

IEEE 802.11 MAC (cont'd)

- Carrier-sense:

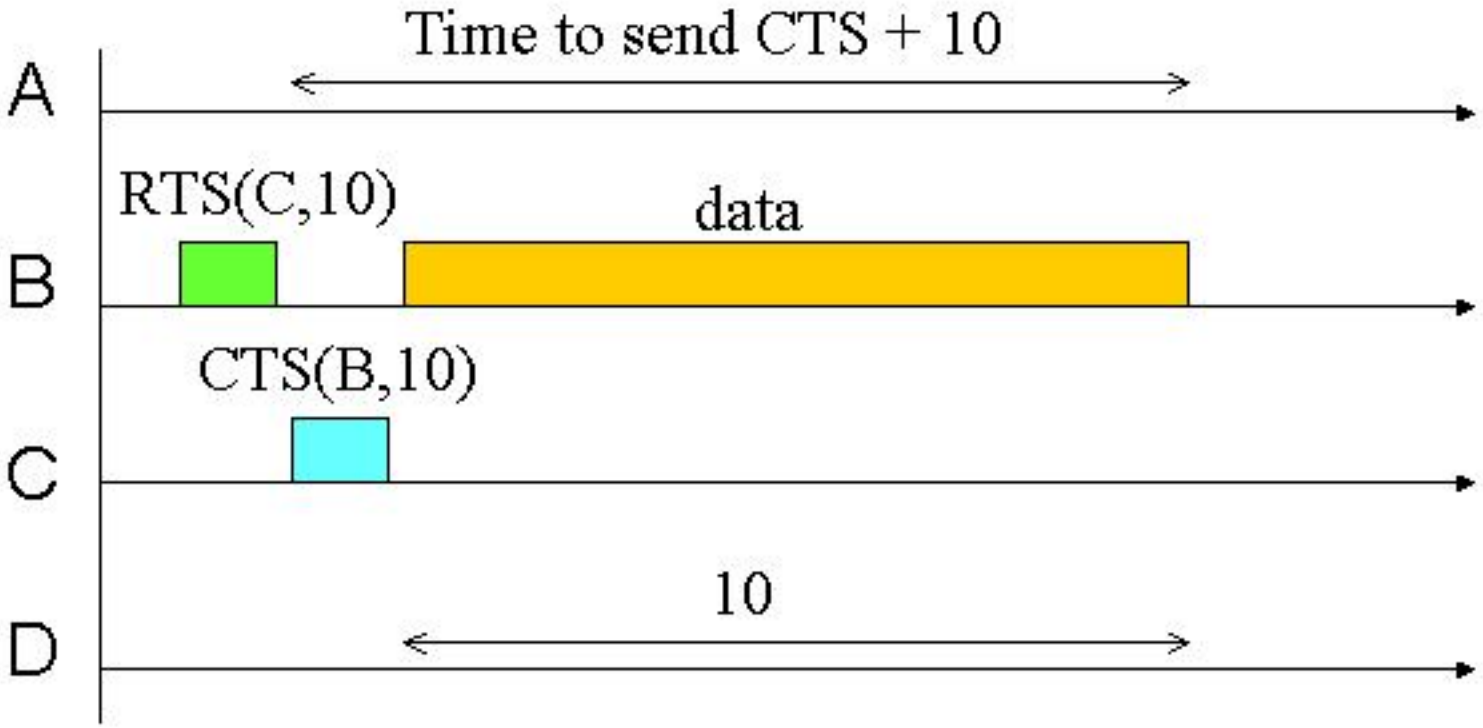


IEEE 802.11 MAC (cont'd)

- Collision-avoidance (stemmed from MACA protocol):
  - Sender transmits special Request-to-send (RTS) packet: the packet contains the length of data to be sent, L.
  - Receiver replies with Clear-to-send (CTS) packet: this packet also contains the length of data (same as before).
  - Every node hearing the RTS remains quiet for CTS+L.
  - Every node hearing the CTS remains quiet for L.
  - If sender does not receive CTS, it knows the receiver is busy and does not transmit data.
  - CTS/RTS packets may still collide, but they are small, so the probability of collisions is reduced.

IEEE 802.11 MAC: example

- Example:



References

- Web sites:
  - www.cs.berkeley.edu/~amc/eecs122
  - www.ietf.org, www.internic.org, etc.
- Text-books:
  - Walrand, *Introduction to Communication Networks*, 2nd Edition, 1997.
  - Peterson&Davie, *Computer Networks: A Systems Approach*, 1996, (2nd Edition soon).
  - Many others .
  - RFCs, etc. (check EECS122 web-site).
- Journals: JSAC, etc.