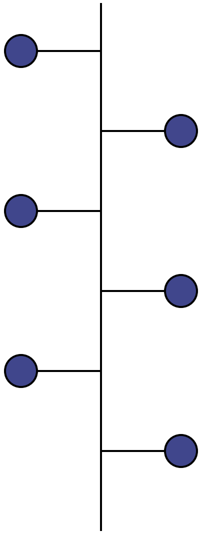
Introduction to Communication Networks and Distributed Systems



Unit 11: Network Layer

Prof. Dr.-Ing. Adam Wolisz

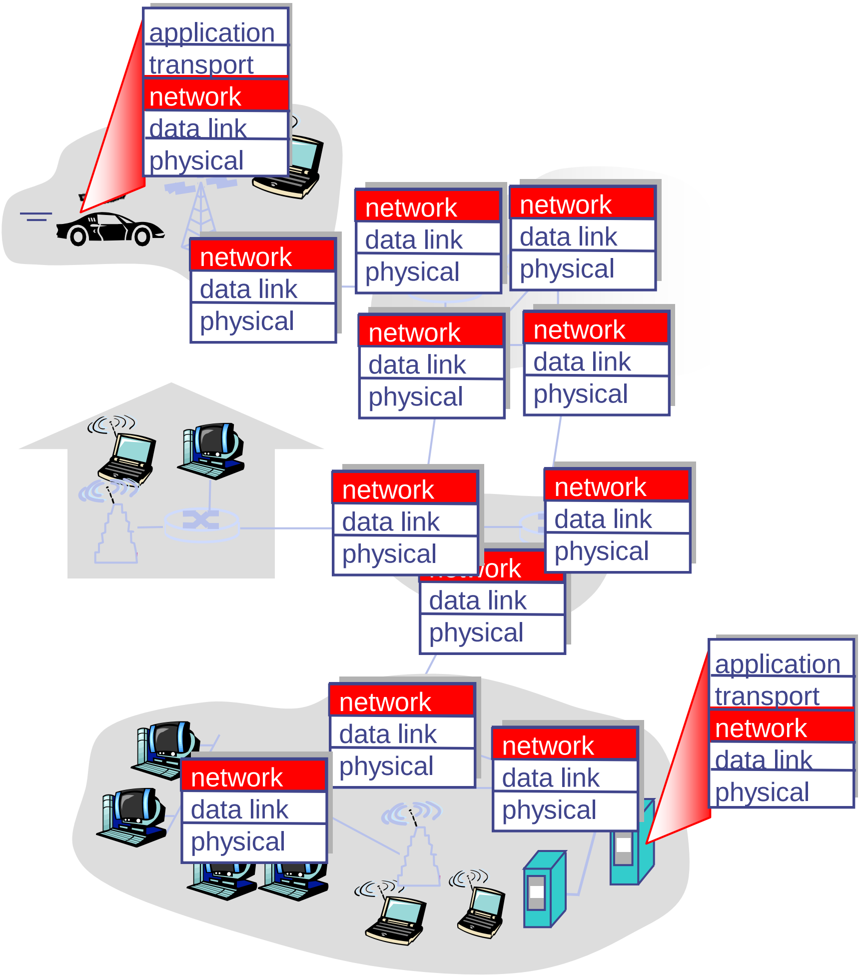
TKN Telecommunication Networks Group

Network layer (Layer 3)

- Overview
  - Network of networks
  - IP and addressing
  - DHCP, ARP
  - Routing

Network layer: connecting networks

- Transport segment from sending to receiving host
- On sending side encapsulates segments into datagrams
- On receiving side, delivers segments to transport layer
- Network layer protocols in every host, router
- Router examines header fields in all IP datagrams passing through it

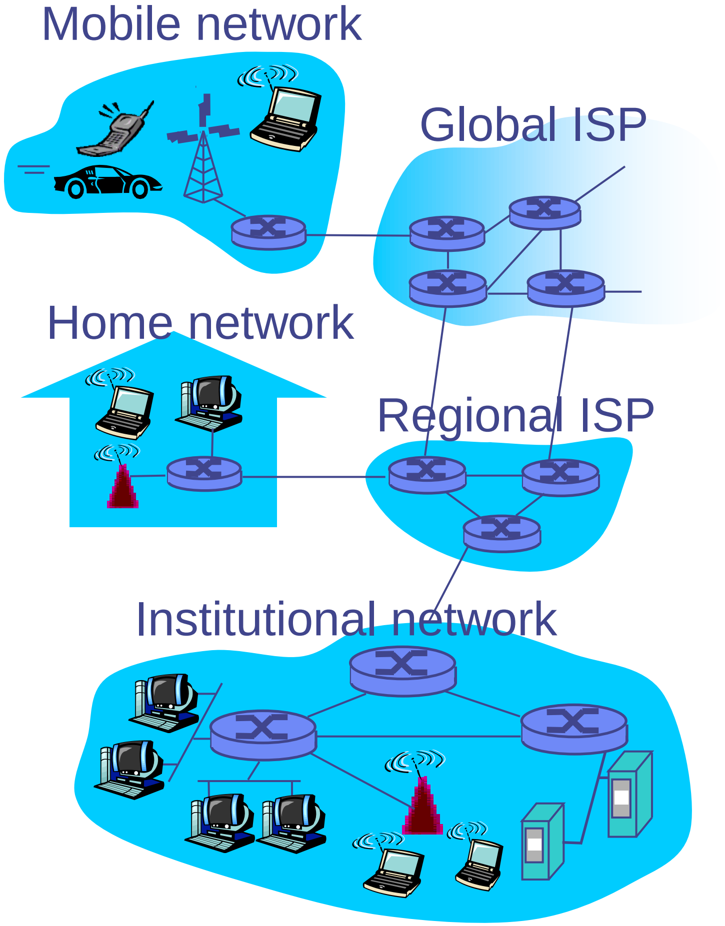


The Key Network-Layer Functions

- Data transmission independent from the applied technology in the subnets
  - ⇒Abstraction from the network topology and overreaching addressing for target systems
- Connection-coupled and connection-less services
- Routing
  - – Path discovery over a number from devices from source station to the target station
  - – Parameters: shortest path, shortest delay, requested quality, cost, …
  - – Wide selection of algorithms from simple (flooding) to sophisticated (Link State Routing)
- Handling network congestions

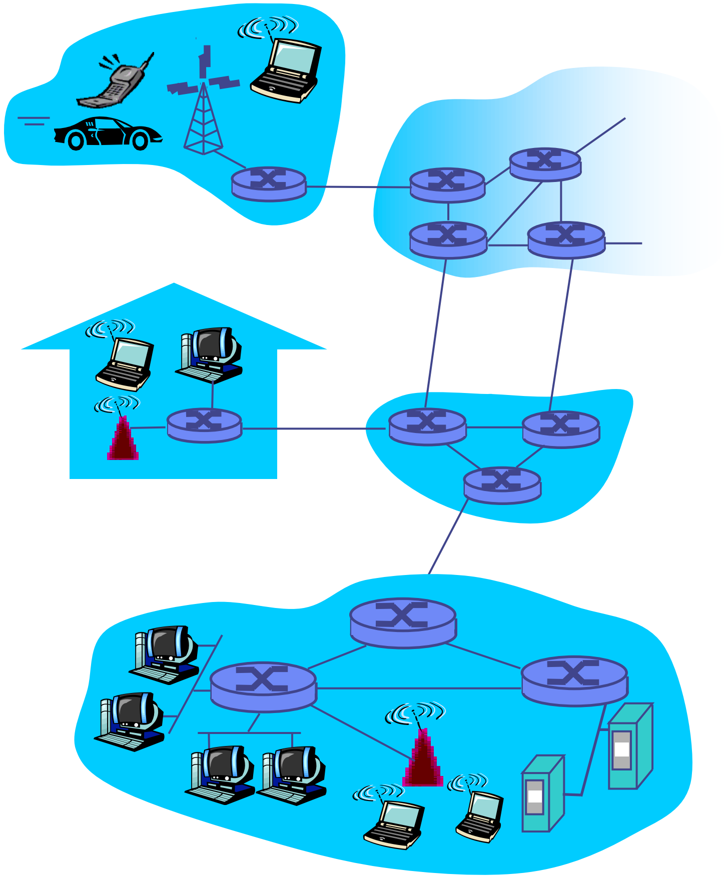
What’s the Internet: “nuts and bolts” view

- Millions of connected computing devices:
  - hosts = end systems
  - running network apps
- Communication links
  - fiber, copper, radio, satellite
  - transmission rate = bandwidth
- Routers



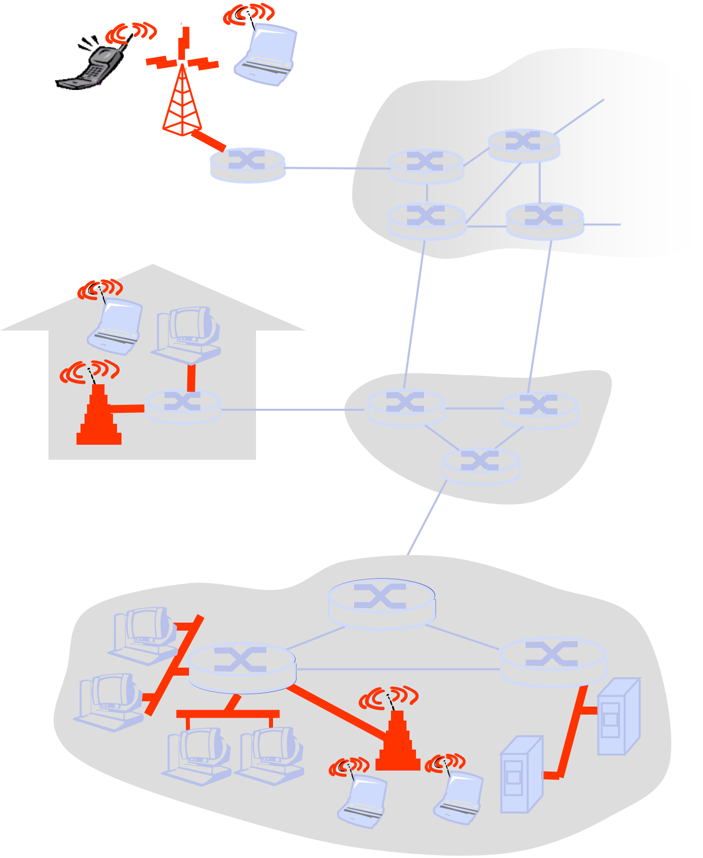
A closer look at network structure

- Access networks, physical media
  - wired, wireless communication links
  - Conencting hosts with their applications
- Network core
  - interconnected routers
  - network of networks



Access networks and physical media

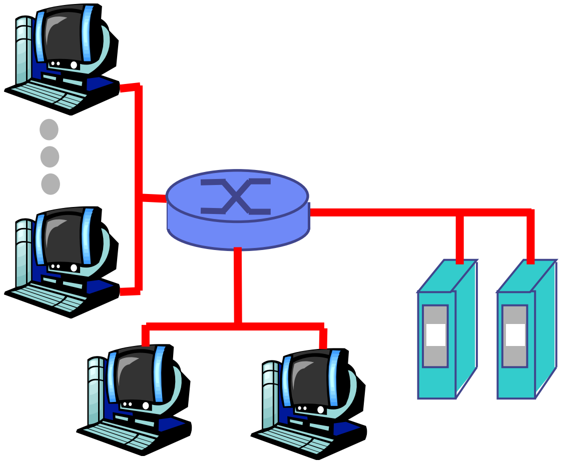
- How to connect end systems to edge router?
  - Residential access nets
  - Institutional access networks (school, company)
  - Mobile access networks



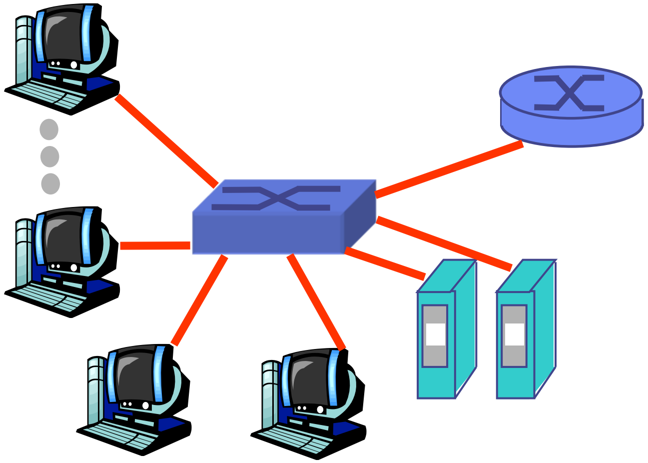
- Keep in mind
  - Bandwidth (bits per second) of access network?
  - Shared or dedicated?

Company access: local area networks

- Company/university local area network (LAN) connects end system to edge router

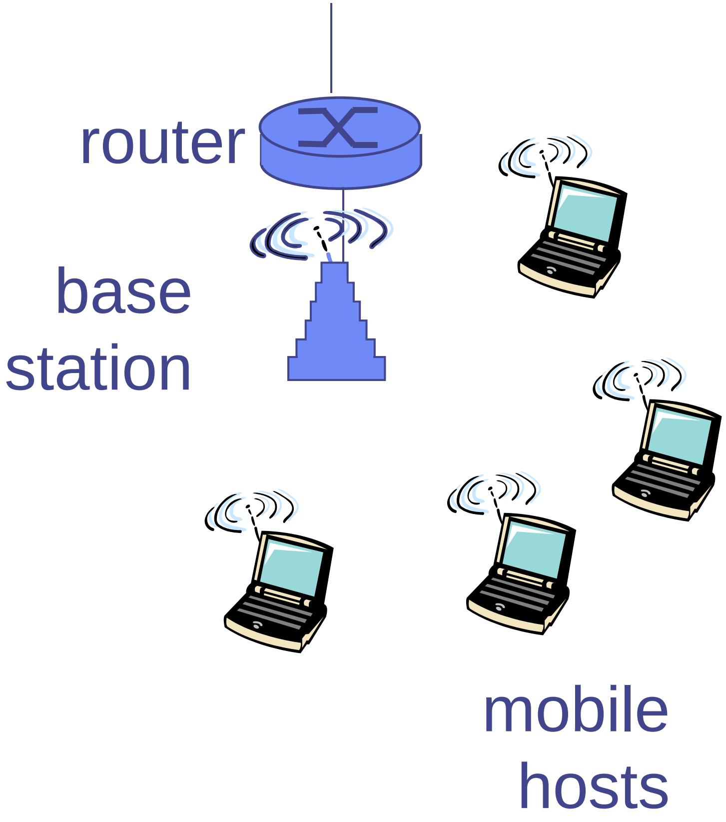


- Ethernet
  - 10 Mbs, 100Mbps, 1Gbps, 10Gbps Ethernet
  - Modern configuration: end systems connect into Ethernet switch



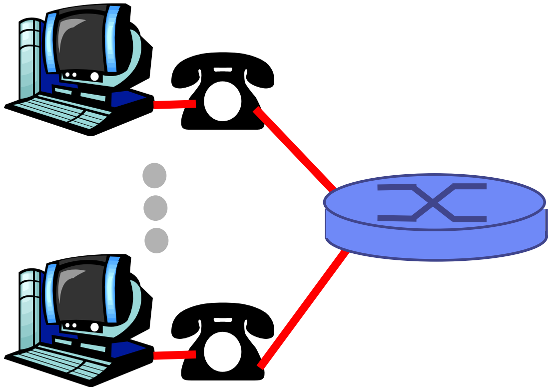
Wireless access networks

- Shared wireless access network connects end system to router
  - via base station aka “access point”
- Wireless LANs
  - 802.11b/g/n (WiFi): 11 / 54 / 600 Mbit/s
- Wider-area wireless access
  - Provided by telco operator
  - ~1Mbps over cellular system (EVDO, HSDPA)
  - next up: LTE over wide area



Residential access: point to point access

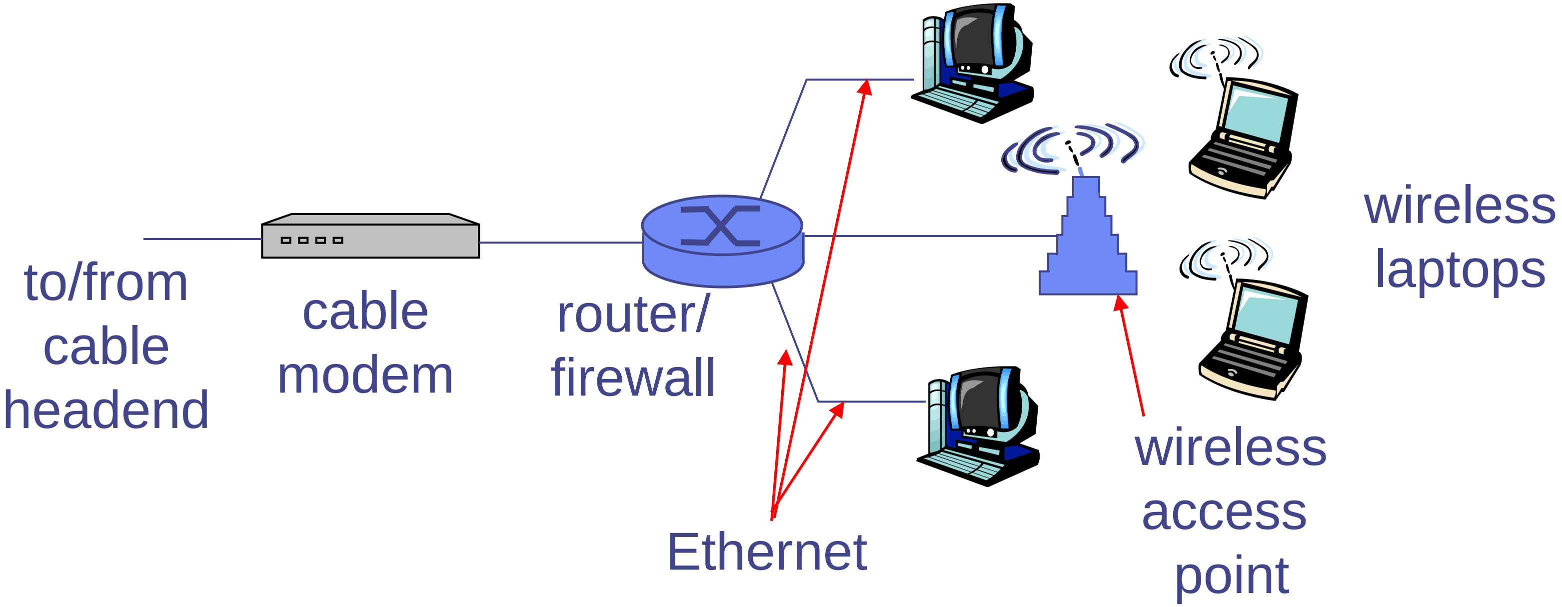
- (history) Dialup via modem
  - up to 56Kbps direct access to router (often less)
  - Can’t surf and phone at same time: can’t be “always on”



- DSL: digital subscriber line
  - deployment: telephone company (typically)
  - dedicated physical line to telephone central office (shared with phone, but parallel usage possible!)
- Cable TV, Fiber….

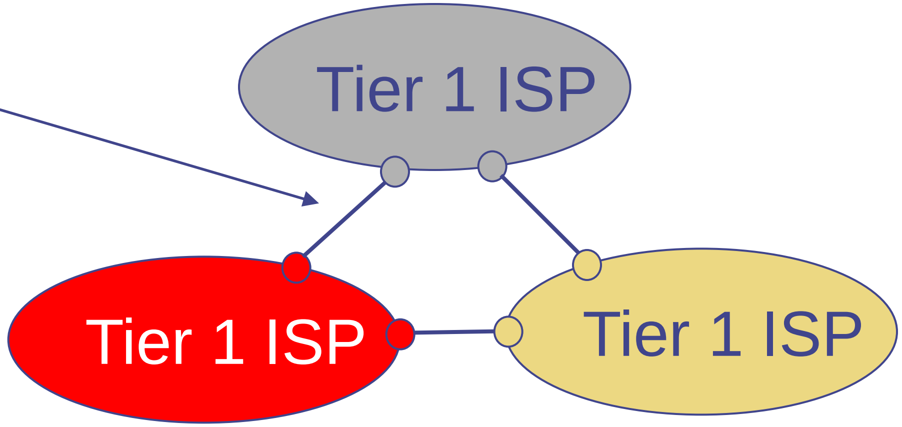
Home networks

- Typical home network components
  - DSL or cable modem
  - Router/firewall/NAT
  - Ethernet
  - Wireless access point



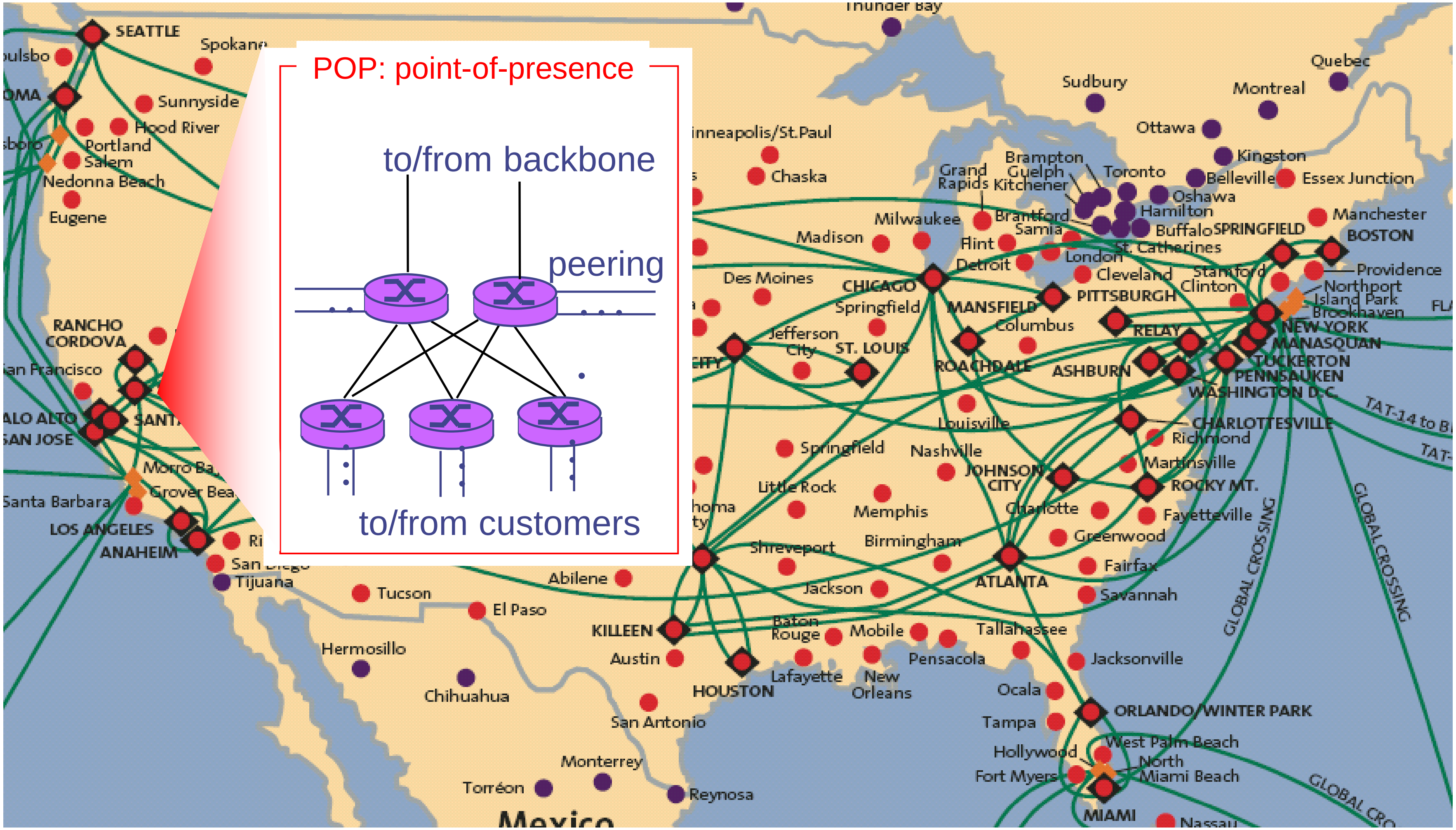
Internet structure: network of networks

- Roughly hierarchical
  - at center: “tier-1” ISPs (e.g. Telekom , 1und1), national / international coverage
  - treat each other as equals



Tier-1 providers interconnect (peer) privately

Tier-1 ISP: Sprint (USA)



POP: point-of-presence

to/from backbone

peering

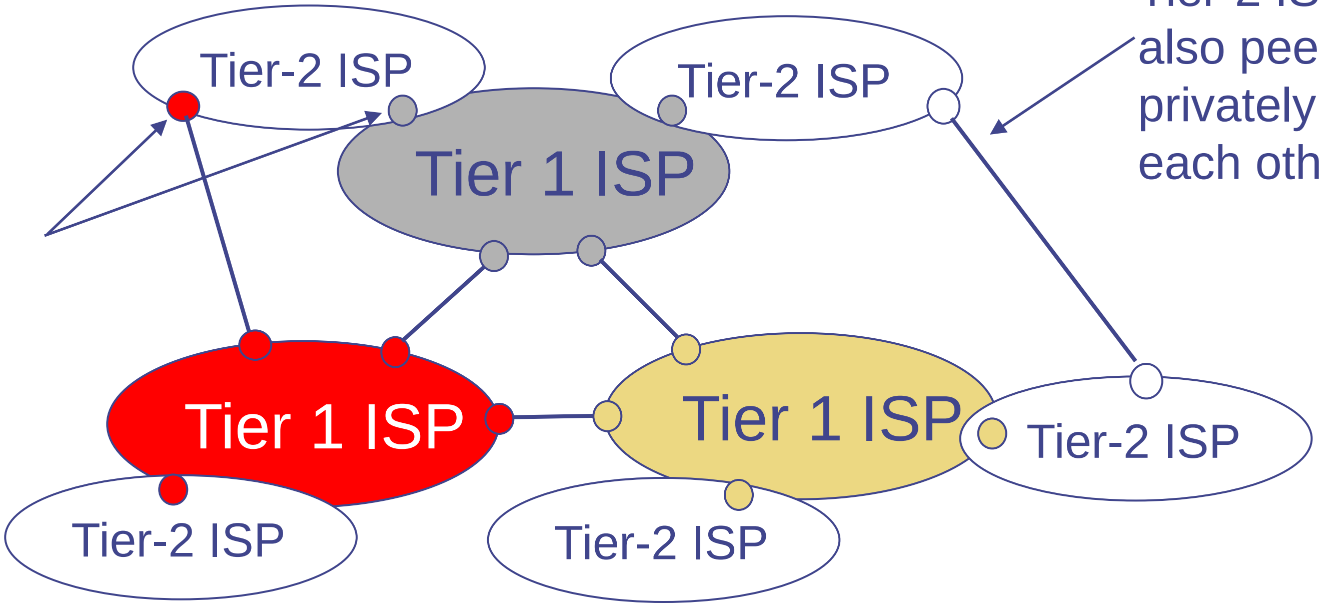
to/from customers

How Are ISPs related?

- Peering
  - The business relationship whereby ISPs reciprocally provide to each other connectivity to each others' local or "inherited" customers
- Transit
  - The business relationship whereby one ISP provides (usually sells) access to all destinations in it's routing table.
  - Sure: the party buying the service is also made visible to the "rest of the world" as seen by the selling ISP…

Internet structure: network of networks

- “Tier-2” ISPs: smaller (often regional) ISPs
  - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs

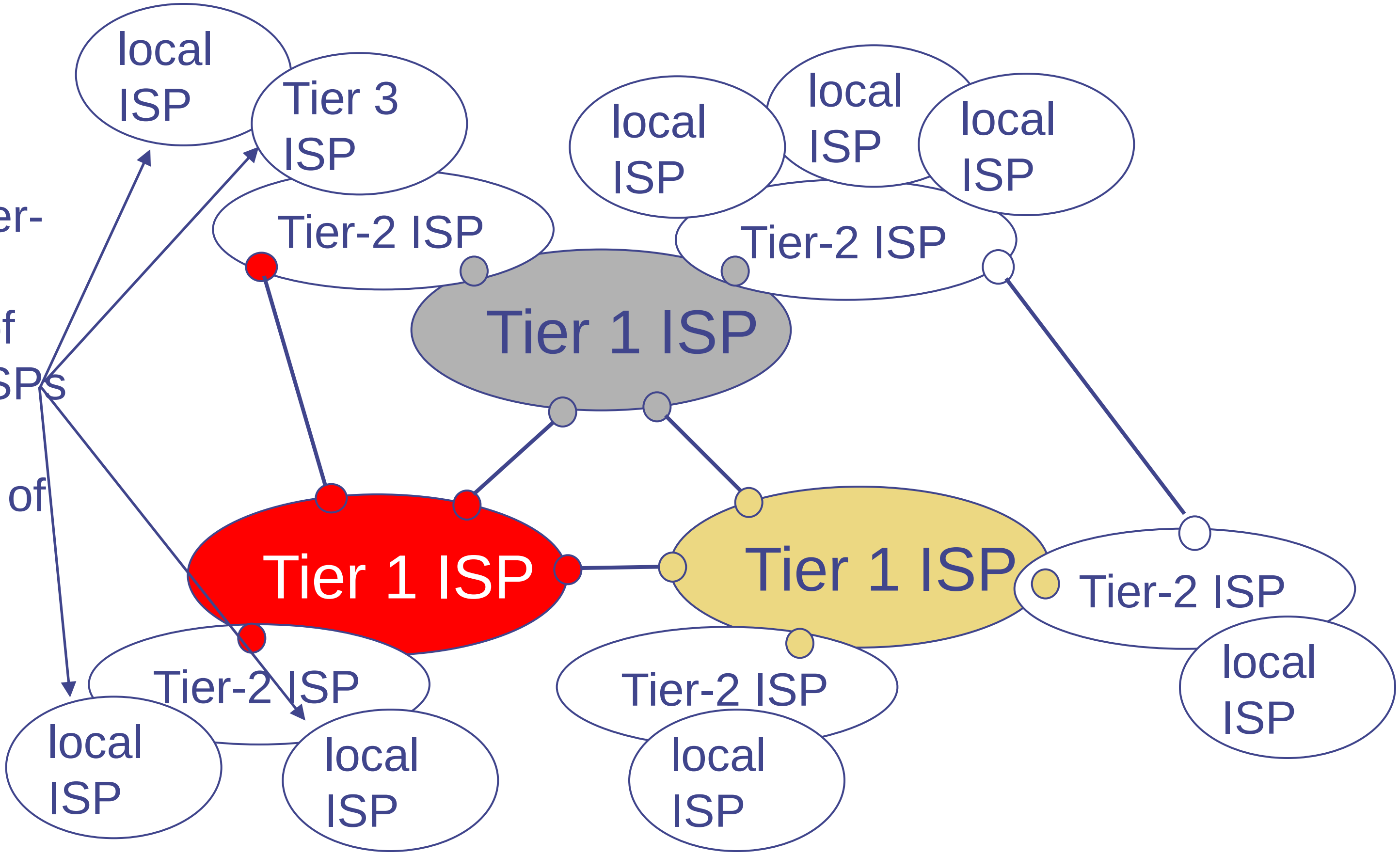


Tier-2 ISP pays tier-1 ISP for connectivity to rest of Internet ⇒ tier-2 ISP is *customer* of tier-1 provider

Tier-2 ISPs also peer privately with each other.

Internet structure: network of networks

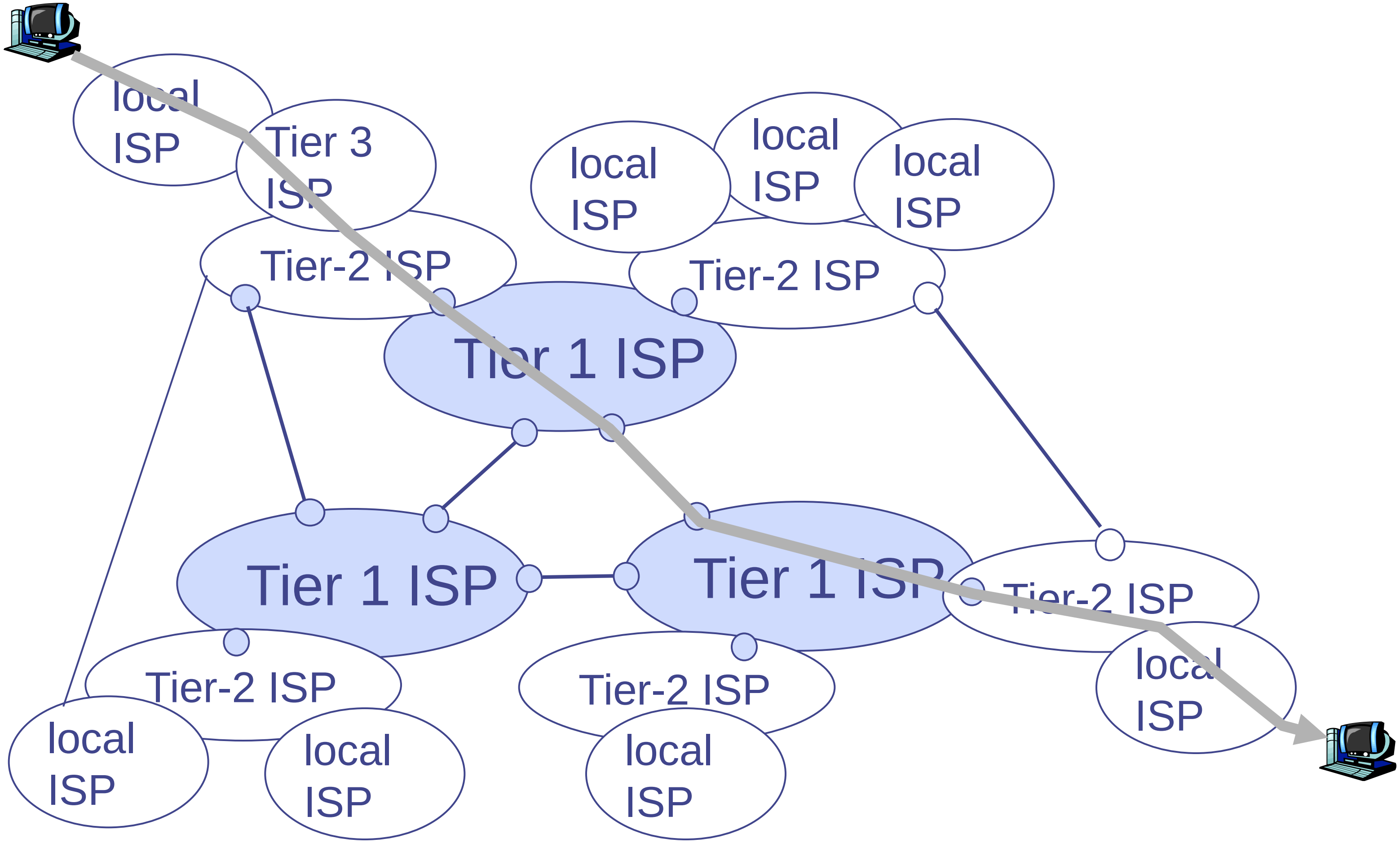
- "Tier-3" ISPs and local ISPs
  - last hop ("access") network (closest to end systems)



Local and tier-3 ISPs are *customers* of higher tier ISPs connecting them to rest of Internet

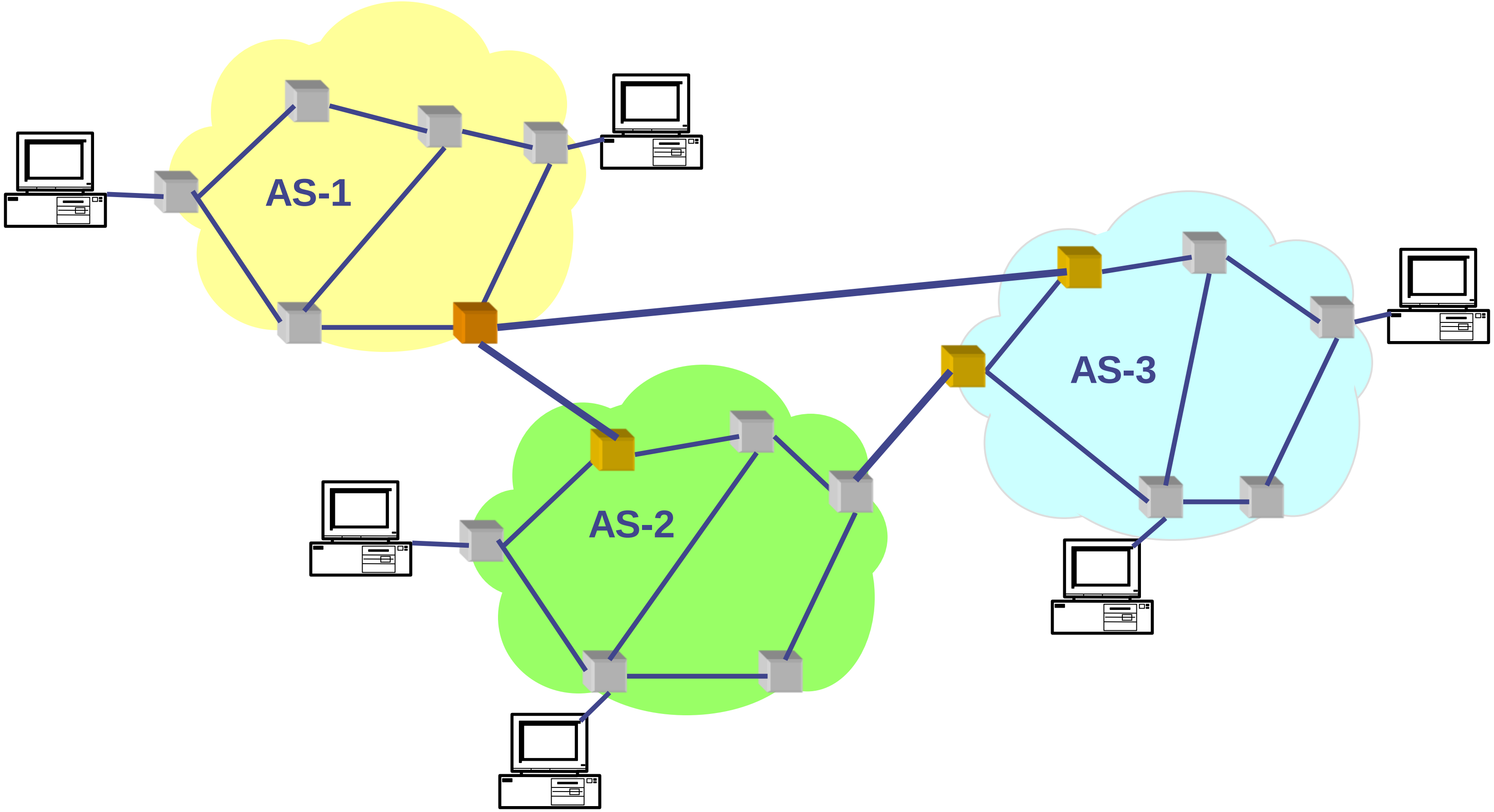
Internet structure: network of networks

- Packet passes through many networks!

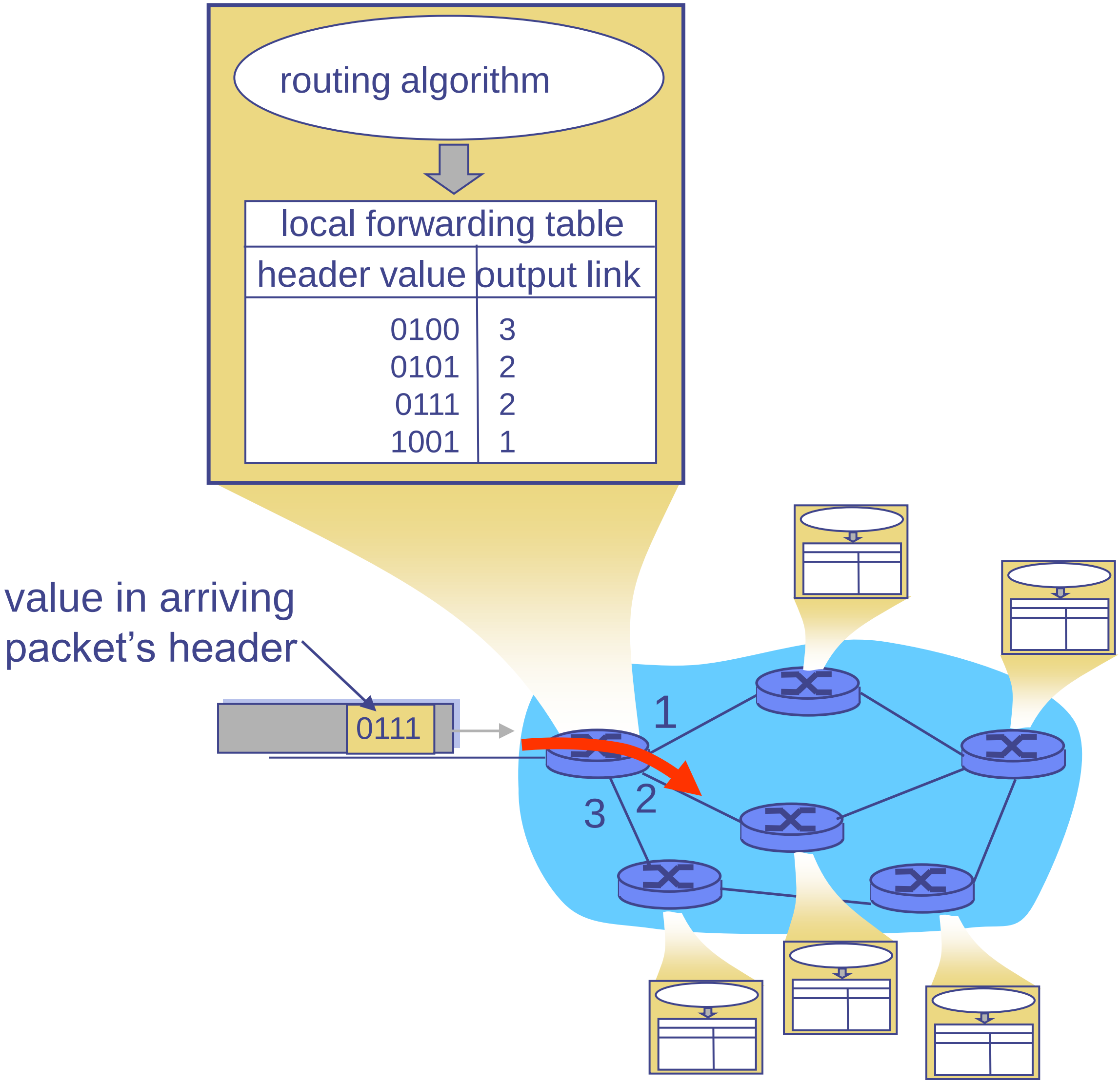


Global view of the internet - reality…

- AS (Autonomous System) set of interconnected networks under common administration



Interplay between routing and forwarding



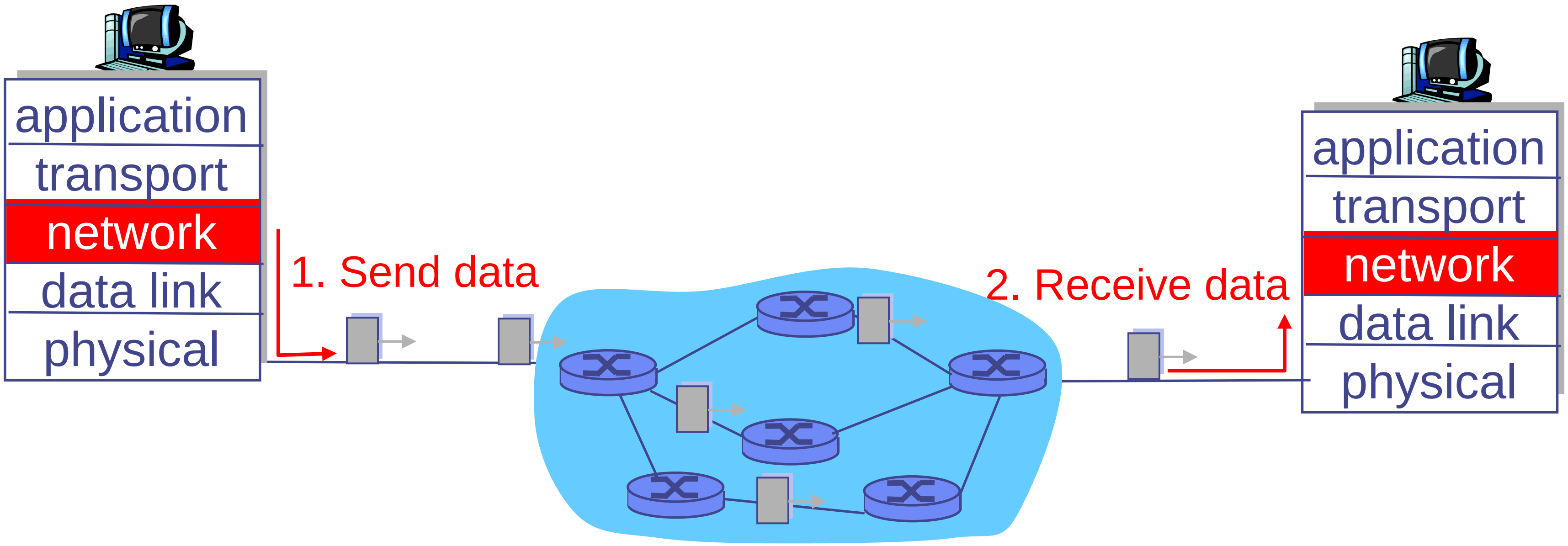
routing algorithm

local forwarding table	
header value	output link
0100	3
0101	2
0111	2
1001	1

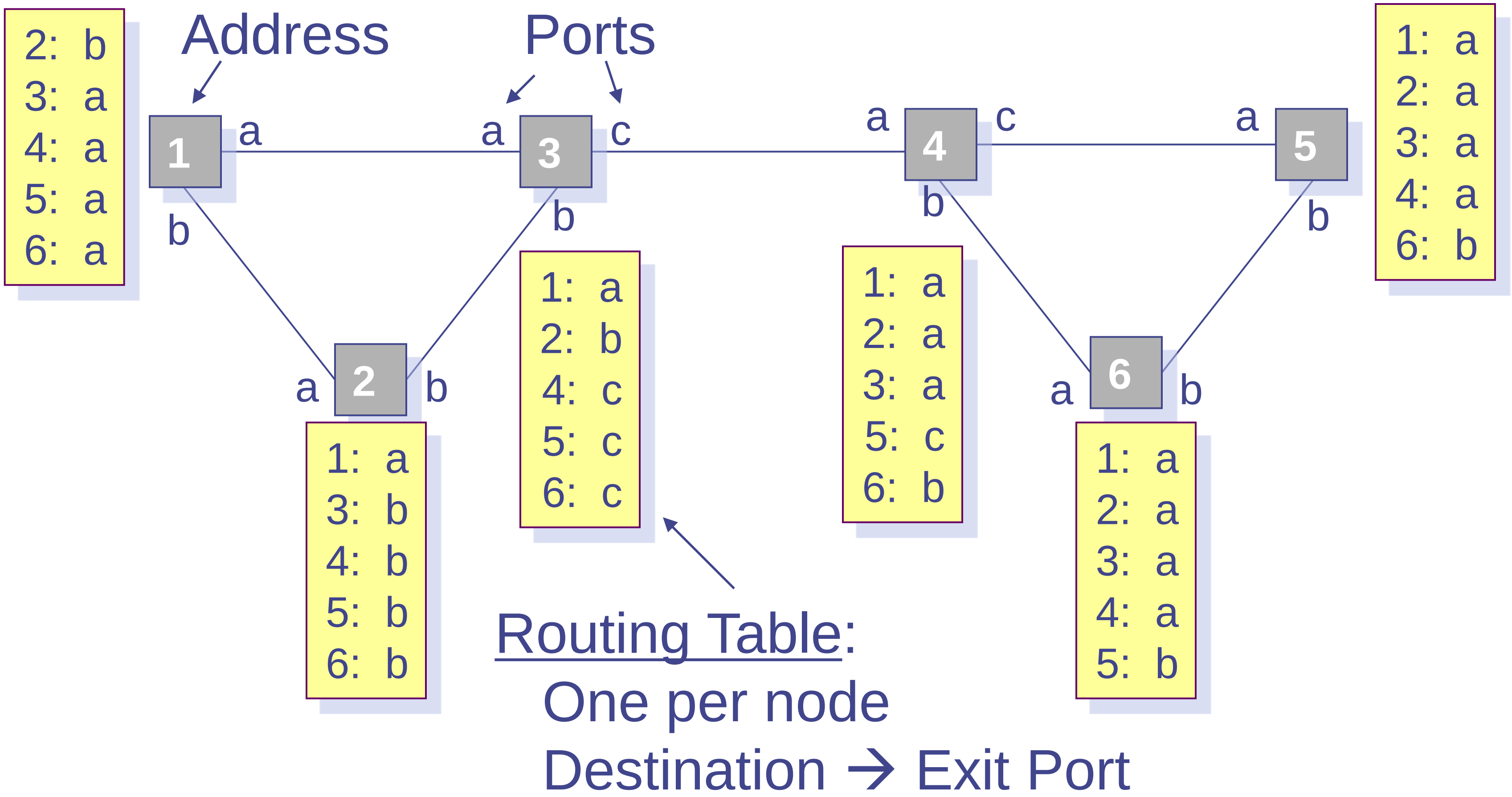
value in arriving packet's header

Datagram networks

- No call setup at network layer
- Routers
  - no state about end-to-end connections
  - ⇒no network-level concept of “connection”
- Packets forwarded using destination host address
  - packets between same source-dest pair may take different paths



Datagrams: Basic forwarding table



Node 1:
2: b
3: a
4: a
5: a
6: a

Node 2:
1: a
3: b
4: b
5: b
6: b

Node 3:
1: a
2: b
4: c
5: c
6: c

Node 4:
1: a
2: a
3: a
5: c
6: b

Node 5:
1: a
2: a
3: a
4: a
6: b

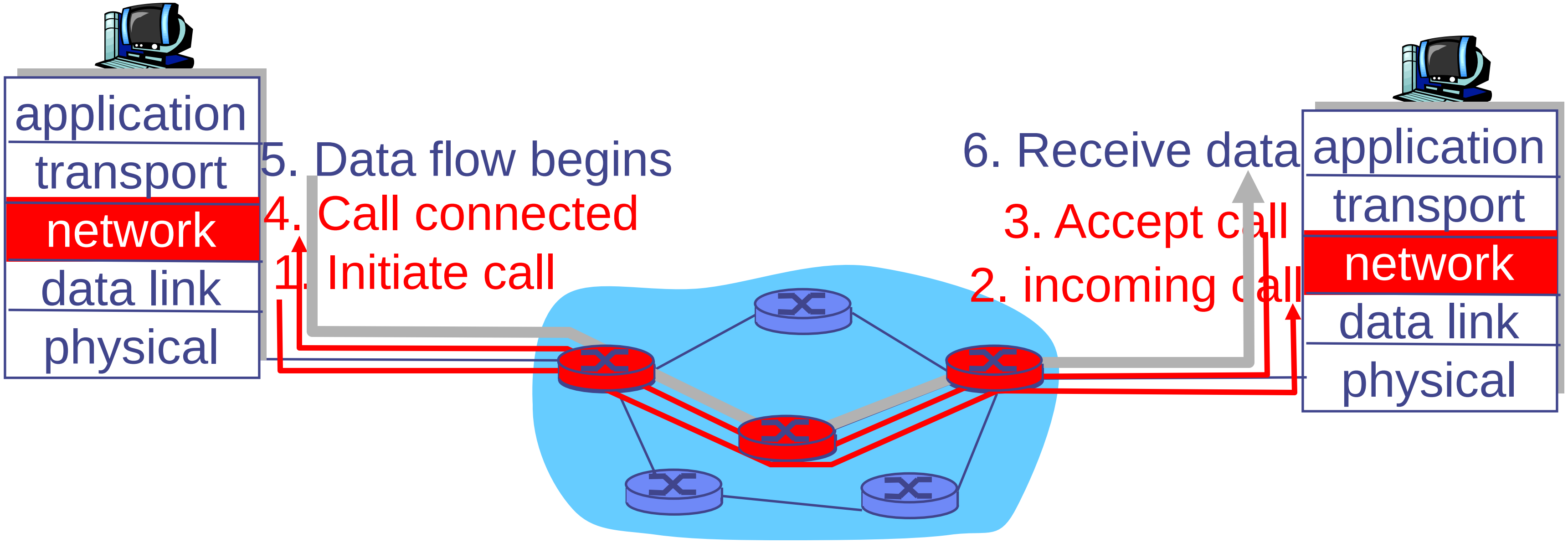
Node 6:
1: a
2: a
3: a
4: a
5: b

Routing Table:
One per node
Destination → Exit Port

Not scalable in case of FLAT addressing structure!!!

Virtual circuits: signaling protocols

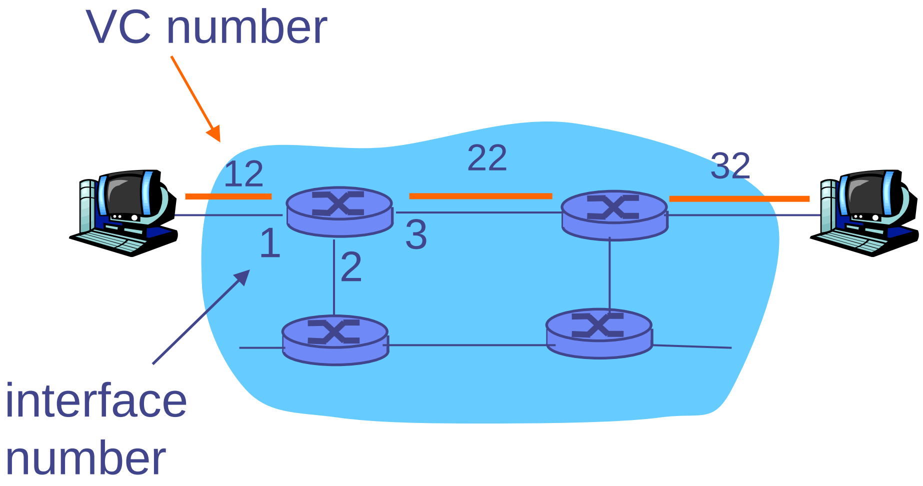
- Used to setup, maintain teardown VC
- Used in ATM, frame-relay, X.25
- Not used in today’s Internet



Virtual circuits

- Virtual circuit consists of
  - Path from source to destination
  - VC numbers, one number for each link along path
  - entries in forwarding tables in routers along path
- Call setup, teardown for each call before data can flow
- Each packet carries VC identifier (not destination host address)
- Every router on source-dest path maintains “state” for each passing connection
- Link, router resources (bandwidth, buffers) may be allocated to VC (dedicated resources = predictable service)
- The “source-to-dest path behaves much like telephone circuit”

Forwarding table



Forwarding table in northwest router

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
…	…	…	…

With VCs: Routers maintain connection state information!
(but only for established VCs!)

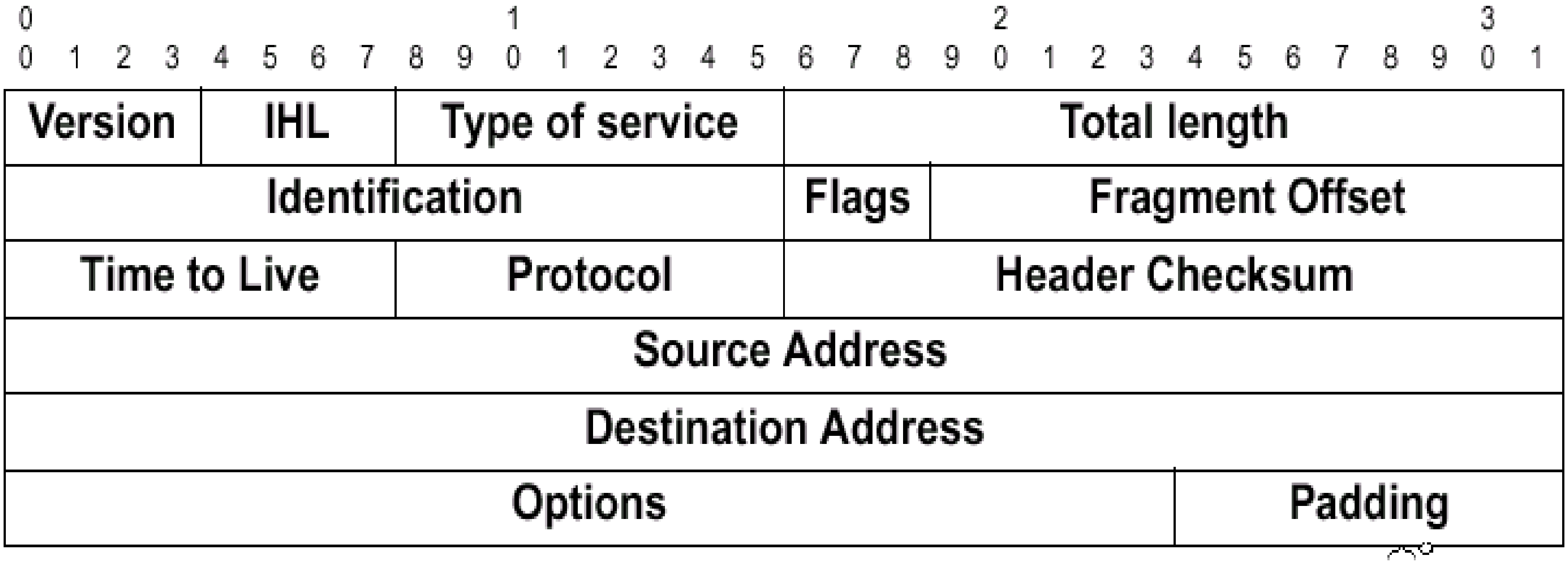
Internet Protocol

- Internet Protocol is specifically limited in scope
- Main issues covered by the protocol are
  - Addressing
  - Forwarding
  - (Fragmentation)
- There are no mechanisms to augment
  - end-to-end data reliability,
  - overload (packets are dropped in this case!)
  - sequencing, or
  - other services common to host-to-host protocols.

The internet protocol **supports the lowest common denominator of service. If - by chance- the underlying quality is good :) if not :(**

IPv4 header (RFC 791)

- Version
  - Version 4 , version 6, further possible…
- IHL
  - Length of the IP header in 32bit words, i.e. specifies the beginning of the payload. Typical length: 20 bytes ☺
- Total Length
  - 16 bits, limits datagram to 65 535 bytes



Version	IHL	Type of service	Total length
Identification		Flags	Fragment Offset
Time to Live	Protocol	Header Checksum	
Source Address			
Destination Address			
Options			Padding

Transferring IP Packet on a network

- IP packets are transported as a PAYLOAD on intermediate networks - e.g. Ethernets
- Underlying network has limits on packet payload length – this is called MTU- maximum transfer unit ....
  - Every internet module must be able to forward a datagram of 68 Bytes
  - Every internet destination must be able to receive a datagram of 576 Bytes
- But: IP sender does not, in general, have to know which networks will transmit the packet... and use longer ones
- Fragmentation – division of a long packet in „Pieces“

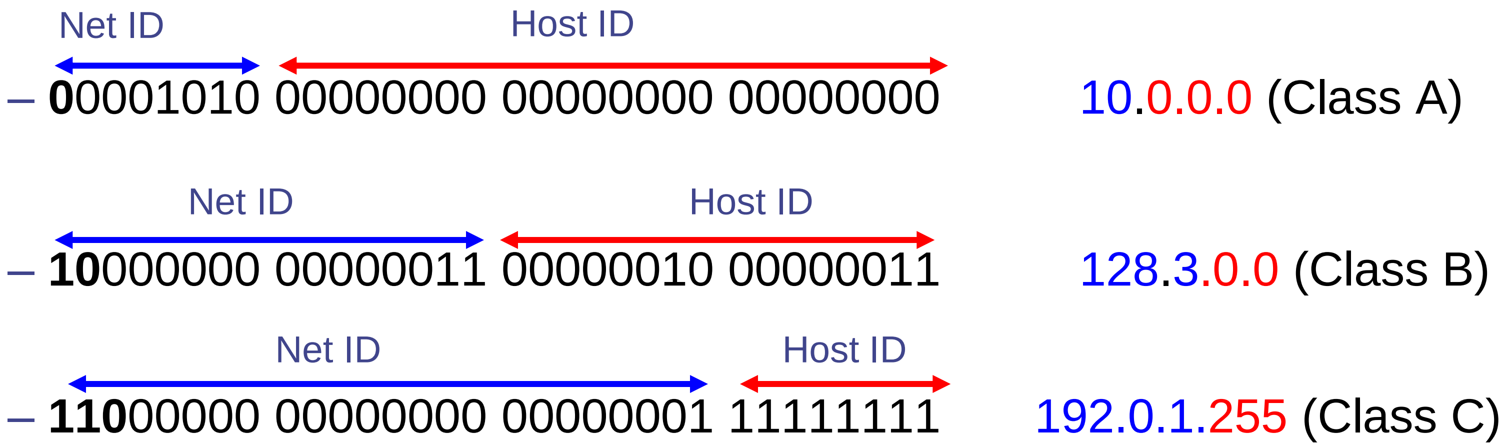
- Alternatively
  - Use packet lengths known as „transportable“ (short enough)
  - Use path features discovery

IP addressing scheme

- IP uses 32 bit address
  - Dotted decimal notation: 4 decimal integers, each specifying one byte of IP address: 130.149.49.60
- Identifies an interface, not a host!
  - ⇒Two or more addresses per host possible – per router a must!
- Special addresses
  - loopback: 127.0.0.1 (packets never appear on network)
  - local broadcast: 255.255.255.255
- Separation of concerns – for scalability support!
  - Network address: used for large scale routing/forwarding only
  - Host address: used for local routing/forwarding within the network

IP Addresses: Structure, Historical View - Classes

- IP addresses consist of 4 integers (8 bit) separated with dots



Net ID Host ID

- **0**0001010 00000000 00000000 00000000 10.0.0.0 (Class A)

Net ID Host ID

- **10**000000 00000011 00000010 00000011 128.3.0.0 (Class B)

Net ID Host ID

- **110**00000 00000000 00000001 11111111 192.0.1.255 (Class C)

Class	First octet	Hosts / network	Nets
Class A	< 128	16 mio.	128
Class B	128..191	65534	16384
Class C	192..223	254	2 mio.
Class D	224..239		268 mio.

Class D: Multicast (not discussed in this course!)

Addresses: Where do they come from, problems

- An ISP gets its address block from its own provider OR from one of the 3 routing registries
  - ARIN: American Registry for Internet Numbers
  - RIPE: Reseaux IP Europeens
  - APNIC: Asia Pacific Network Information Center

- Example: an organization initially needs 100 addresses
  - Allocate it a class C address
  - Organization grows to need 300 addresses
  - Class B address is allocated. (~64K hosts)
    - ⇒That's overkill - a huge waste
    - ⇒Only about 8200 class B addresses!

  ⇒Artificial address crises

Lookup Performance Required

[McKeown]

- Routers have to be quick....
  - Do we really want to have all Class C Networks in each router?

Line Rate	**Pktsize=40B**	**Pktsize=240B**
1.5Mbps	4.68 Kpps	0.78 Kpps
155Mbps	480 Kpps	80 Kpps
622Mbps	1.94 Mpps	323 Kpps
2.5Gbps	7.81 Mpps	1.3 Mpps
10 Gbps	31.25 Mpps	5.21 Mpps

Kpps = kilo packets per second

CIDR: Classless InterDomain Routing

- CIDR allows networks to be assigned on arbitrary bit boundaries
  - Address ranges can be assigned in chunks of 2k k=1…32
- Idea: Aggregation
  - ⇒provide routing for a (large?) number of networks by advertising one common prefix
  - ⇒Reduces the size of routing tables, but maintains connectivity
- Address format: a.b.c.d/x, where x is # bits in subnet portion of address

200.23.16.0/23

CIDR address blocks [liebherr]

- CIDR notation expresses blocks of addresses
  - ⇒Blocks are used to allocate IP addresses for routing tables

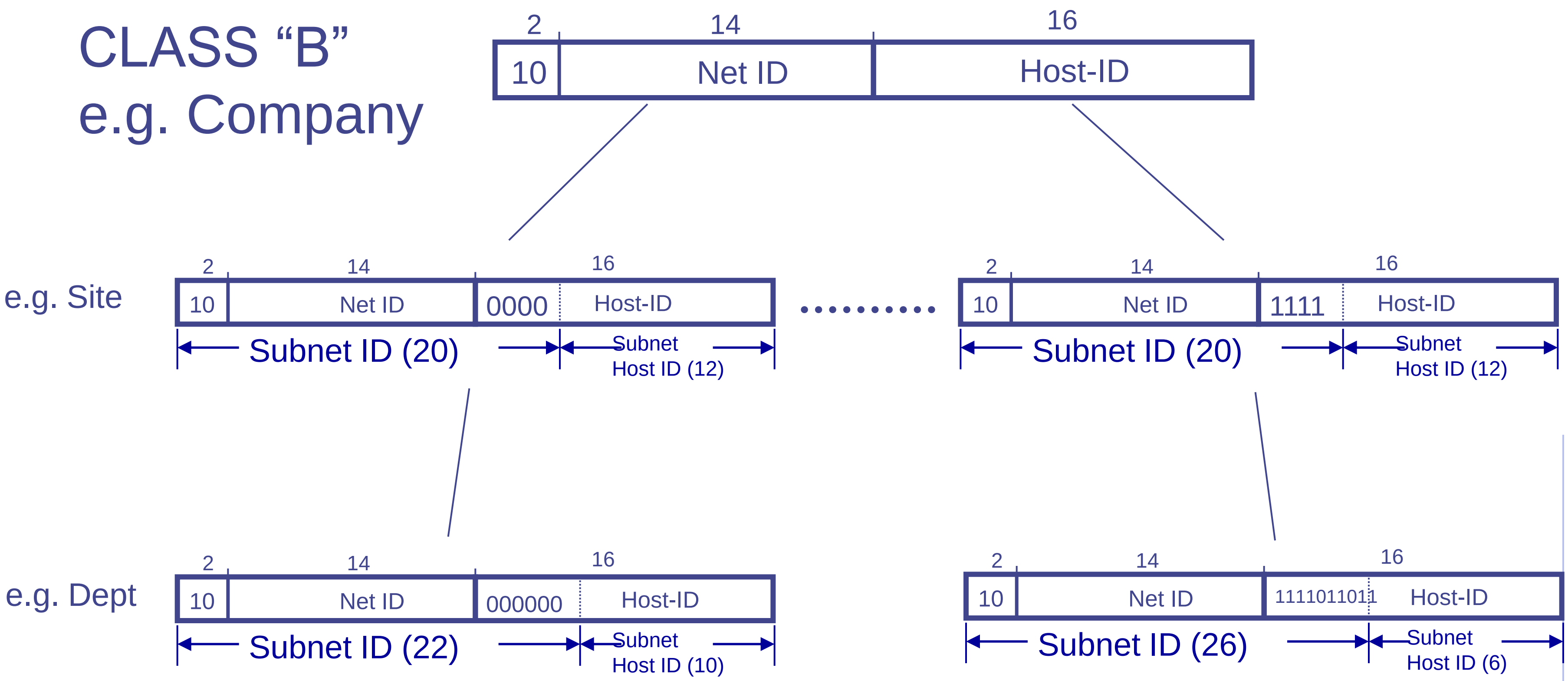
⇒CIDR Block Prefix	# of Host Addresses
/27	32
/26	64
/25	128
/24	256
/23	512
/22	1,024
/21	2,048
/20	4,096
/19	8,192
/18	16,384
/17	32,768
/16	65,536
/15	131,072
/14	262,144
/13	524,288

CIDR – an Example

- Suppose 50 computers in network are assigned IP addresses 128.23.9.0 – 128.23.9.49
  - ⇒They share the prefix 128.23.9
- Is this the longest prefix?
  - Range is
    - 01111111 00001111 00001001 00000000 to
    - 01111111 00001111 00001001 00110001
  - How to write
    - 01111111 00001111 00001001 00x
  - Convention: 128.23.9.0/26
  - There are 32-27 = 6 bits for the 50 computers
    - ⇒2^6 = 64 IP addresses

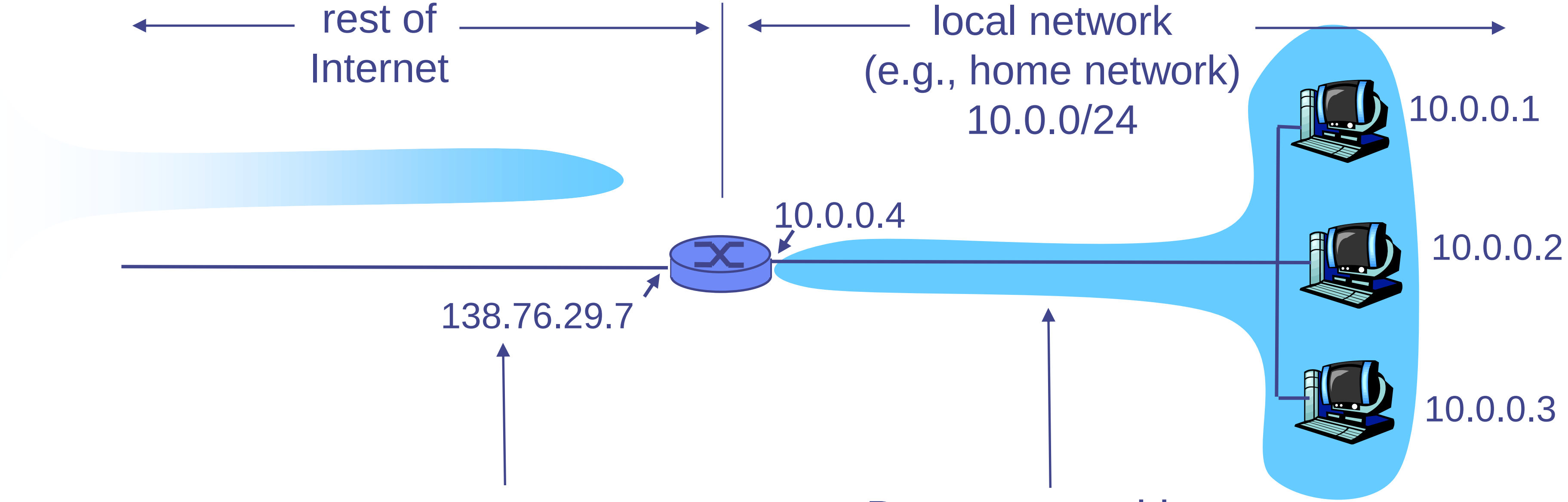
Subnetting [McKeown]

- Large organizations: multiple LANs with single IP network address
  - ⇒Subdivide “host” part of network address ⇒ subnetting



Network Address Translation

- NAT idea: Show ONE IP address, run multiple IP addresses



All datagrams leaving local network have same single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT - Objectives

- Enlarge IPv4 address space
  - Provider view: IP address are a scarce resource
  - Address shortness is one motivation for a new IP version (IPv6)
- Prevent home users from running servers at home
  - Session must be initiated from “inside”
- Connect multiple hosts to the Internet using single IP address
  - User view: Each IP address (contract with ISP) costs money
- Hide internal topology to outside world
  - Security aspect (administrator view)

How do I get an Internet address?

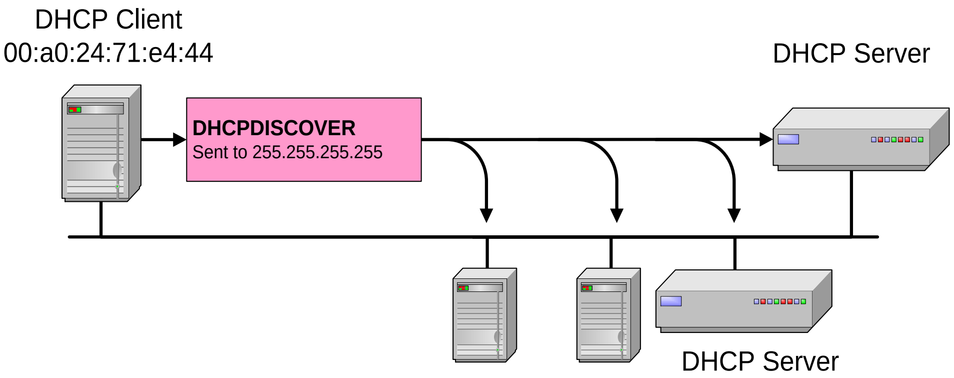
- Your notebook has
  - Ethernet interface with a MAC address
  - WLAN interface with a MAC address
  - Possibly some more

- I arrive at the Campus – How do I get an IP Address?
  - You might start looking for your Sysadmin, request an IP address...
  - YOU go home... You call the system provider...
  - You go to the coffee shop

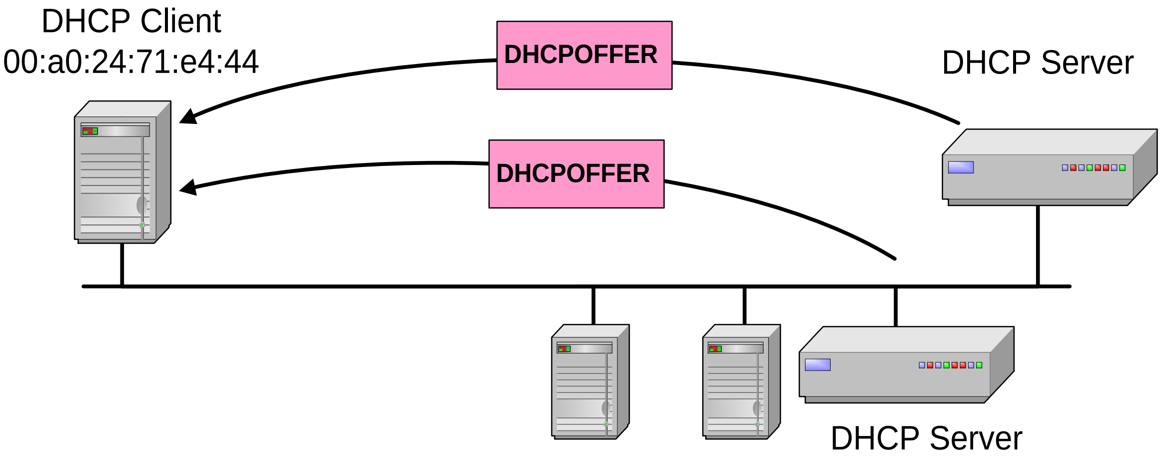
⇒IP address is dependent on who is the service provider

DHCP Operation

DHCP DISCOVER

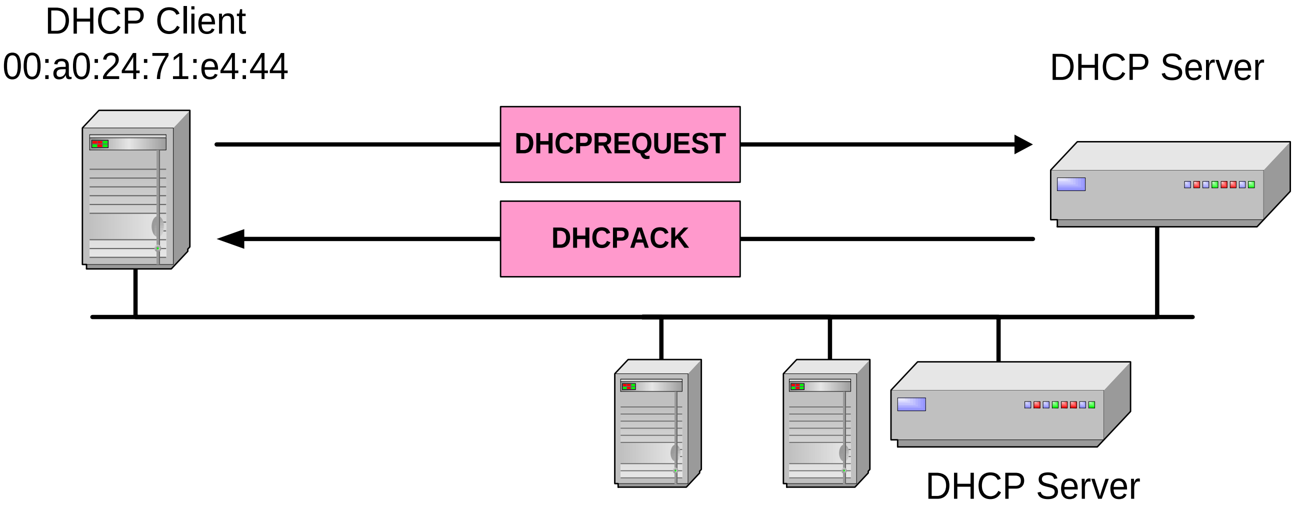


DHCP OFFER



DHCP Operation

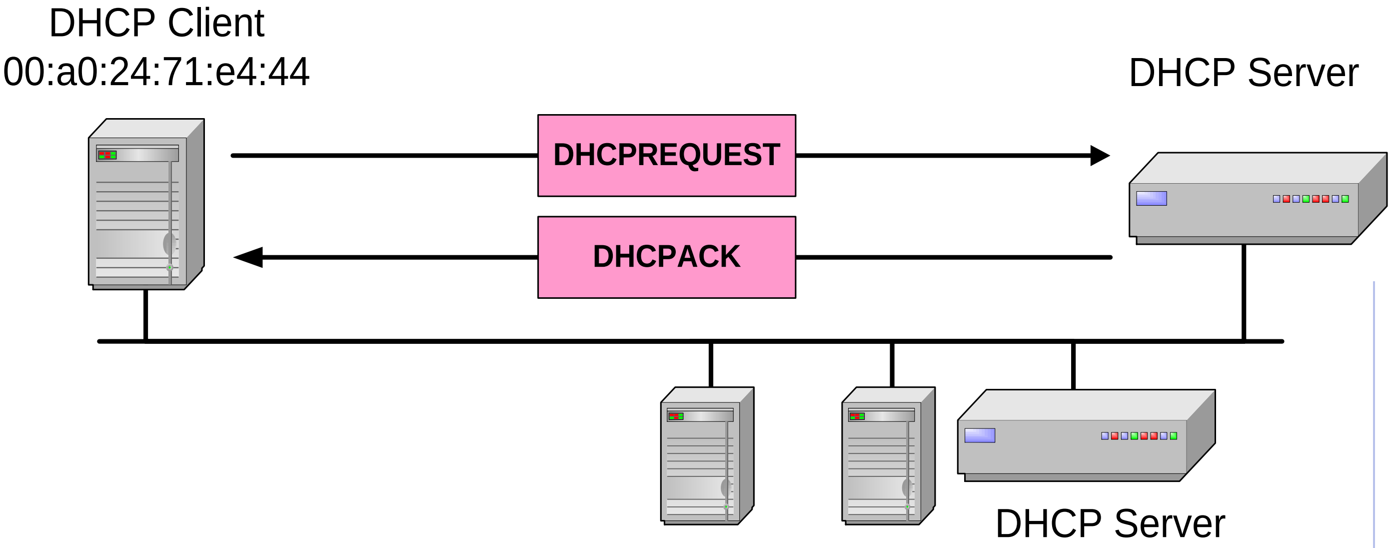
At this time, the DHCP client can start to use the IP address



Renewing a Lease (sent when 50% of lease has expired)

If DHCP server sends DHCPNACK, then address is released.

NOTE: Soft state concept!



MAC vs. IP address

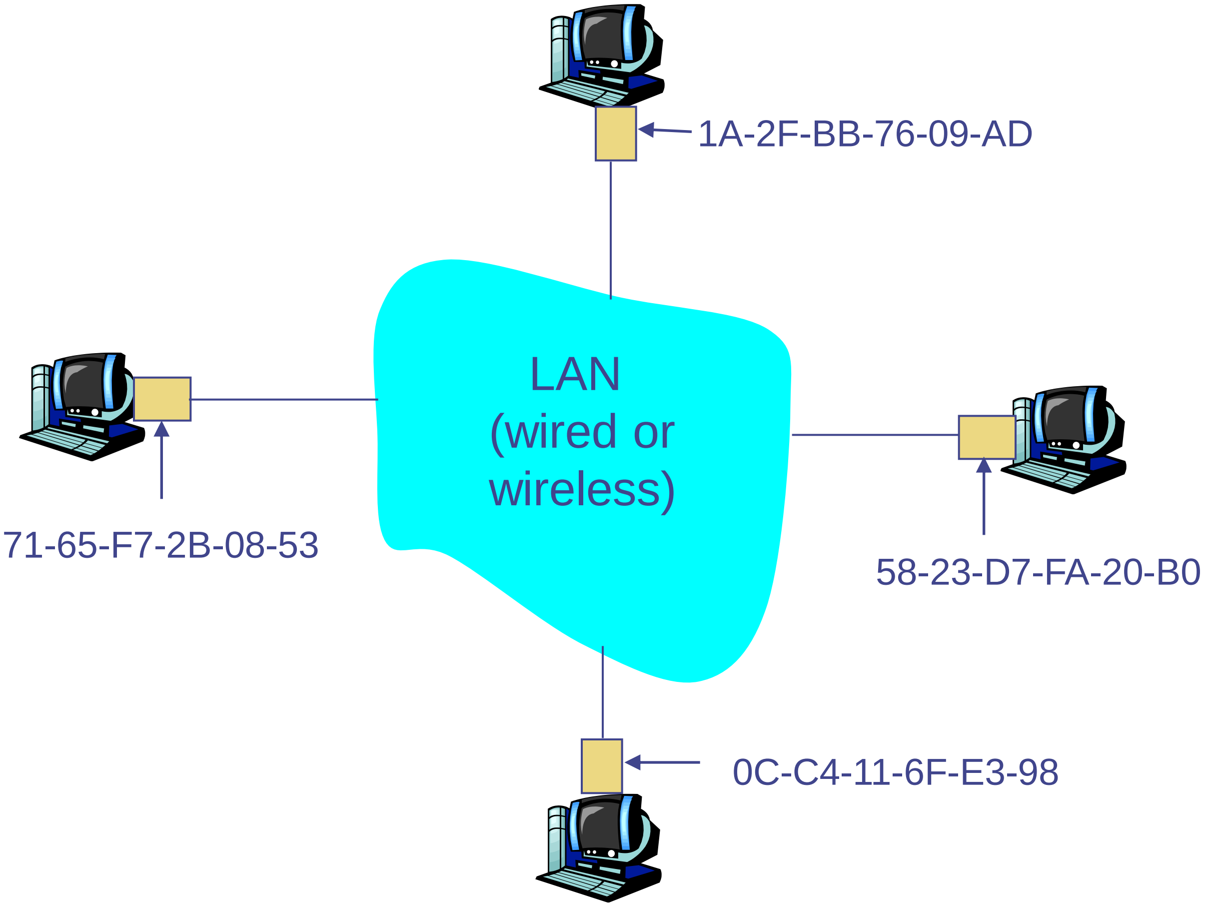
- Analogy
  - MAC address: like Social Security Number
  - IP address: like postal address
- MAC flat address ➜ portability
  - Can move LAN card from one LAN to another
- IP hierarchical address NOT portable
  - Address depends on IP subnet to which node is attached

MAC Addresses and ARP

- 32-bit IP address
  - Network-layer address
  - Used to get datagram to destination IP subnet
- MAC (or LAN or physical or Ethernet) address
  - Function: get frame from one interface to another physically-connected interface (same network)
  - 48 bit MAC address (for most LANs)
    - burned in NIC ROM, also sometimes software settable
- Internet protocols use dynamic assignment for MAC addresses to Internet addresses with ARP (Address Resolution Protocol)

LAN Addresses and ARP

Each adapter on LAN has unique MAC address (also called LAN address)



1A-2F-BB-76-09-AD

LAN
(wired or
wireless)

71-65-F7-2B-08-53

58-23-D7-FA-20-B0

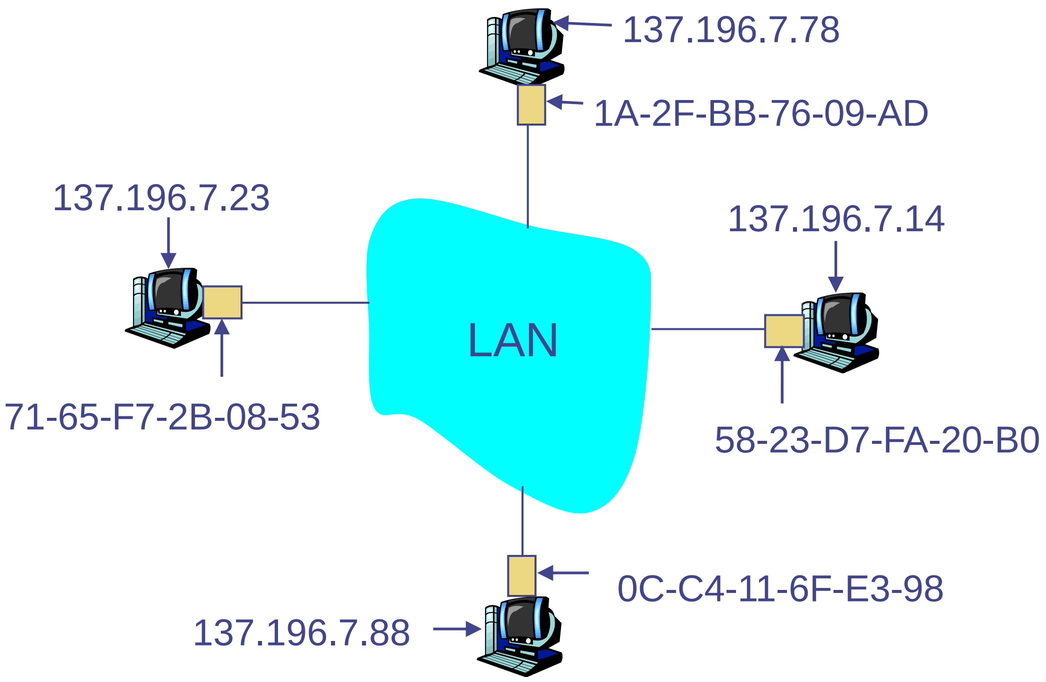
0C-C4-11-6F-E3-98

Broadcast address =
FF-FF-FF-FF-FF-FF

🟨 = adapter

ARP: Address Resolution Protocol

Question: how to determine MAC address of B knowing B's IP address?



- Each IP node (host, router) on LAN has ARP table
- ARP table
  - IP/MAC address mappings for some LAN nodes
  - < IP address; MAC address; TTL>
  - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

ARP protocol: Same LAN (network)

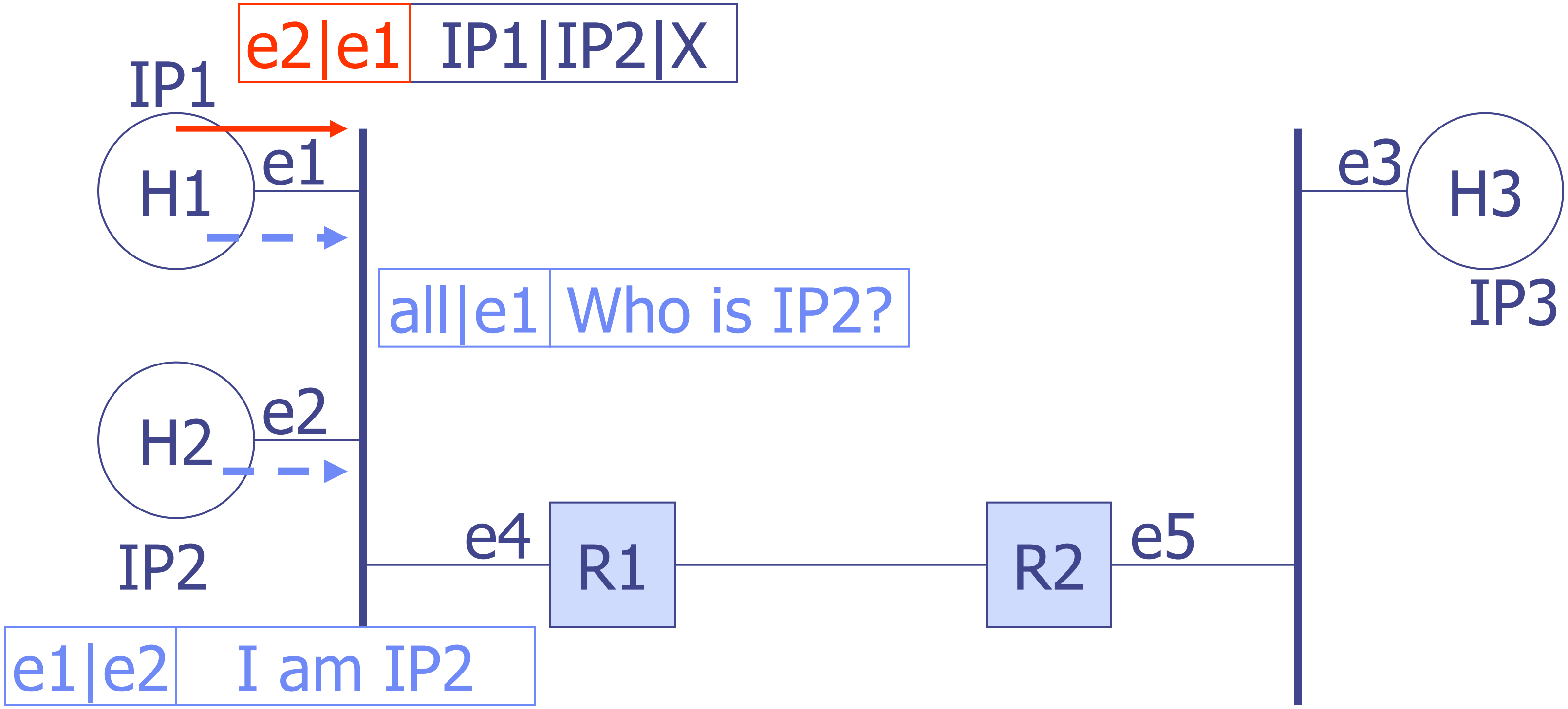
1. A wants to send datagram to B, and B's MAC address not in A's ARP table
2. A broadcasts ARP query packet, containing B's IP address
   - dest MAC address = FF-FF-FF-FF-FF-FF
   - All machines on LAN receive ARP query
3. B receives ARP packet, replies to A with its (B's) MAC address
   - frame sent to A's MAC address (unicast)
4. A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
   - Soft state: information that times out (goes away) unless refreshed

- ARP is "plug-and-play":
  - Nodes create their ARP tables without intervention from net administrator

IP packet delivery in a subnet

[Walrand, 122]

IP1 → IP2 on same subnet



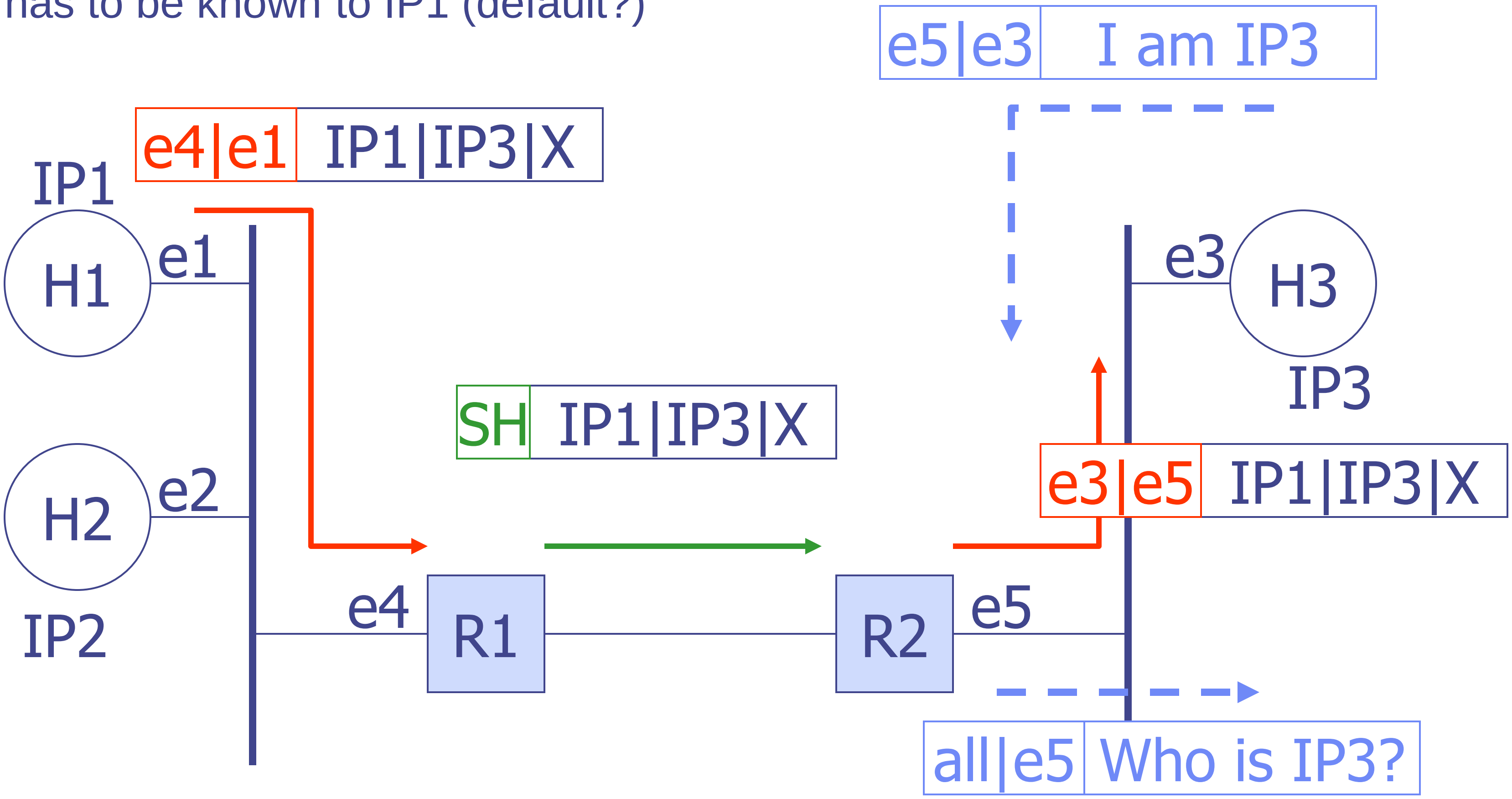
Address Resolution Protocol = Layer 3 Address → Layer 2 Address

IP Packet delivery not on the same subnet

[Walrand, 122]

Note: IP1 has explicitly specified the MAC address e4 – Unlike in bridging – the router has to be known to IP1 (default?)

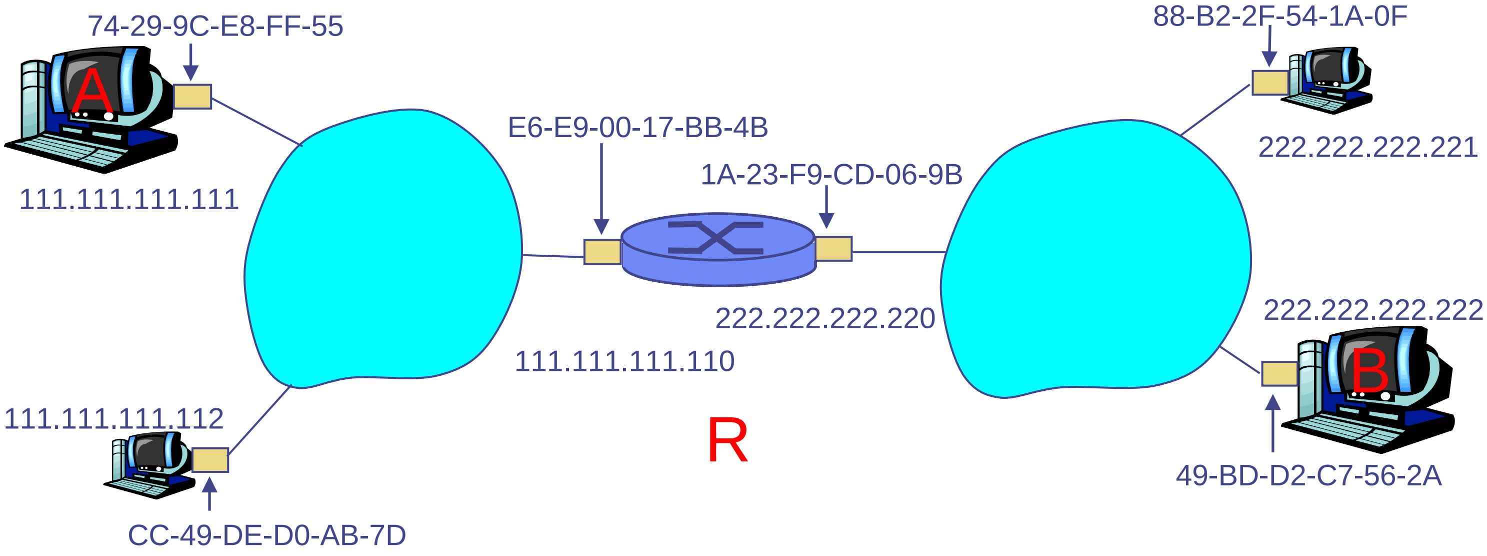
IP1 → IP3 not on same subnet



Note: Fragmentation may be required at R1

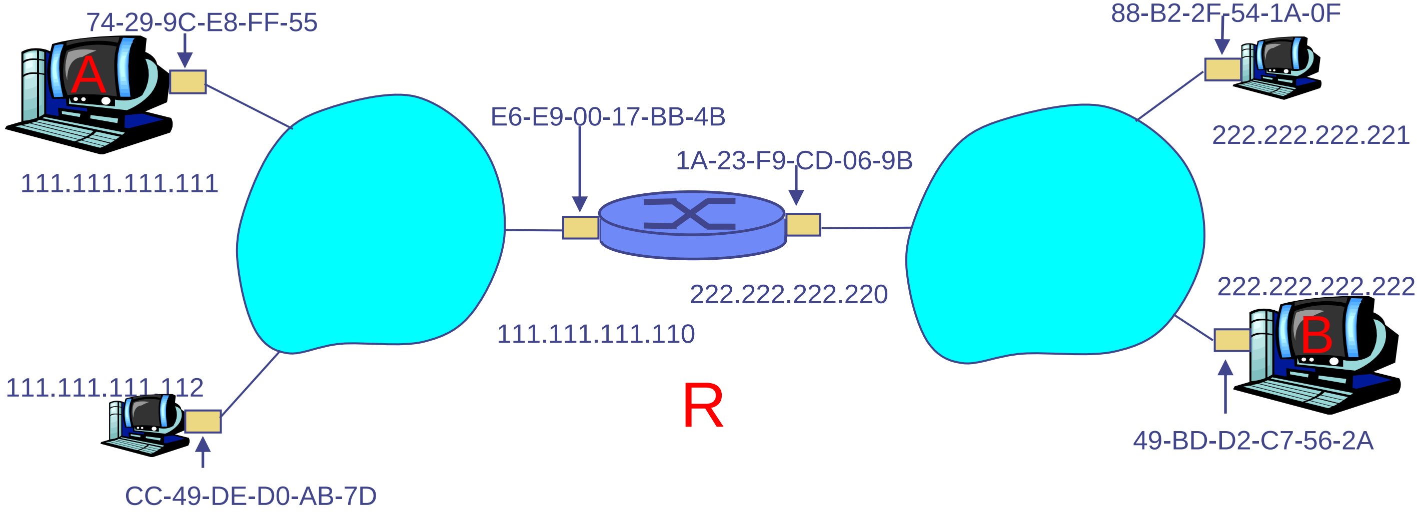
Addressing: routing to another LAN

- Walkthrough
  - send datagram from A to B via R assume A knows B’s IP address
  - Two ARP tables in router R, one for each IP network (LAN)



Example

- A creates IP datagram with source A, destination B
- A uses ARP to get R’s MAC address for 111.111.111.110
- A creates link-layer frame with R's MAC address as destination, frame contains A-to-B IP datagram
- A’s NIC sends frame, R’s NIC receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B’s MAC address
- R creates frame containing A-to-B IP datagram sends to B



Routing

- Building larger networks by simply interconnecting LANs is limited, it does not scale
- To build larger networks, the following questions have to be explicitly solved
  - What are good paths that a packet should take to get from a source node to a destination node?
  - How to represent these paths by *routing tables* and how to construct them efficiently?
  - How to *use* routing tables (once constructed) efficiently?
  - How to organize larger networks with respect to an addressing structure that allows efficient & compact routing tables?

Build LARGE networks: Simple options

- Some simple options for next hop selection
  - Flooding – Each router sends every datagram over all outgoing links except the incoming link
    - Exceptions
      - Datagram addressed for the router (destination reached)
      - Router received datagram earlier
    - Advantage: reachable destination will be accessed via the shortest path
    - Disadvantage: Many useless datagram duplicates
  - Hot potato routing – send to a randomly chosen neighbor
- Simple options not convincing
  - Try to find good, i.e., short routes – few hops
  - Try to learn about the structure of the network, interpreted as a graph

Desirable: Shortest paths!

- Shortest paths
  - Given a source and destination node for a packet, what is the shortest way to deliver the packet?

- Routing by shortest path
  - Computation of shortest paths based on global knowledge of the worldwide network graph
  - Weights: 1 point per hop
  - Continuous, cost-intensive update of the network graph required

- What does “shortest” mean?
  - Fewest hops?
  - Smallest delivery time?
  - Lowest cost?

- Choice to make: to which neighbor to forward a packet?

  ⇒Construct routing tables

Routing tables [Karl, UPB]

- Criteria
  - Good/perfect estimate of real distances, absence of loops, …
- Constructing routing tables
  - Initially, typically empty – how should a new node know anything?
  - Passive: observe ongoing traffic (e.g., from hot potato routing) and try to extract information, successively improve table correctness
  - Actively exchange information between routers to try to learn network structure – routing protocols
- Problem: Size!
  - In large networks, maintaining routing entries for all possible destinations quickly becomes infeasible
  - Solution: hierarchy – treat "similar" nodes identically (divide et impera) → internetworking

Routing tables expressing costs

- Routing table
  - For a given node and all its direct neighbors, express cost to send to any destination

- Construct from routing table: Forwarding table
  - For a given node and any destination, express to which neighbor a packet should be passed on to minimize cost
  - Trivial to construct from routing table, but smaller and quicker to search

Routing table of W

Destination

Neighbor

	M	P	Z
U	**2**	3	4
V	3	**2**	3
X	4	3	**2**
Y	4	4	3

Forwarding table of W

Destination

M	P	Z
U (2)	**V (2)**	**X (2)**

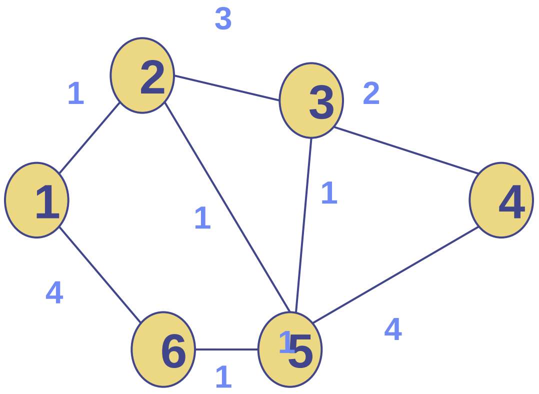
Computing routing tables – A centralized view

- Given a graph G=(V,E) and a cost function c : E → Real
- Compute, for each node $v \in V$, the routing table to each destination $u \in V$ such that
  - for each pair (v,u), the path (v, s_1, …, s_n, u) with the minimal smallest cost can be derived easily from the routing table
    - By simply choosing the neighbor with the smallest entry
    - Cost of a path is the sum of the costs of its edges
- "Single-source shortest path problem"
  - Approach:
    - Compute shortest paths from a given node to all possible destination nodes;
    - do that for all nodes in the network

  ⇒"Shortest path tree"
  - NOT a minimum spanning tree computation

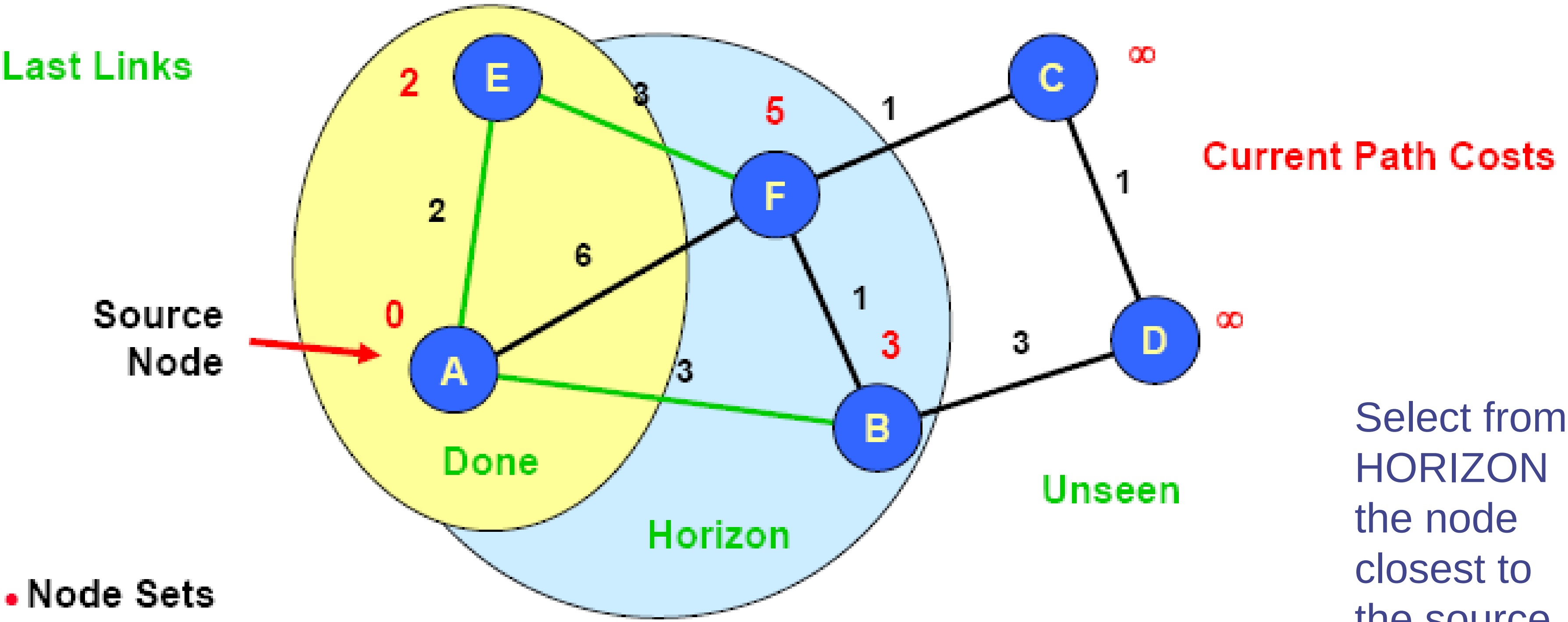
Dijkstra

- Every node knows the graph
  - –All link weights are >= 0
- Goal at node 1: Find the shortest paths from 1 to all the other nodes.
- Strategy: Find the shortest paths in order of increasing path length



Dijkstra – Intuition

[steenkiste]



Select from HORIZON the node closest to the source, and add it to DONE

- **Node Sets**
  - Done
    - Already have least cost path to it
  - Horizon:
    - Reachable in 1 hop from node in Done
  - Unseen:
    - Cannot reach directly from node in Done

- **Label**
  - d(v) = path cost
    - From s to v
- **Path**
  - Keep track of last link in path

Dijkstra

Notation

c(i,j) >=0 :cost of link from (i,j)

D(1,i): Shortest path from 1 to i.

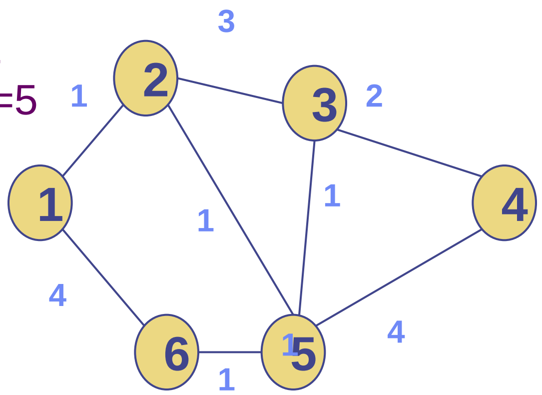
D(1,x,i): Shortest path from 1 to i via x

Let P(k) be the set of nodes k-closest to 1

P(2)={1,2}

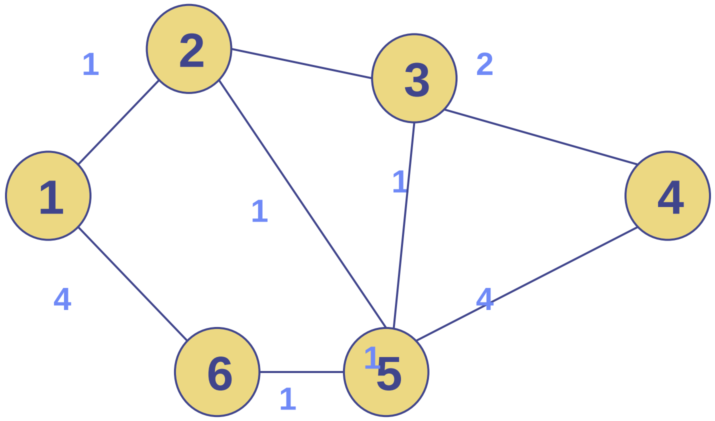
D(1,5)=2

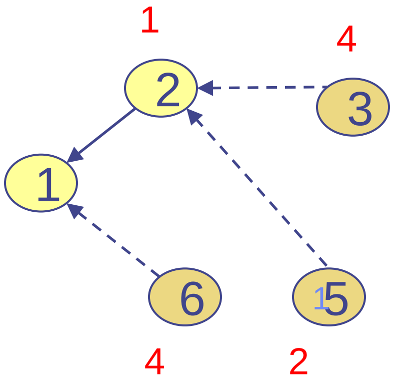
D(1,6,5)=5



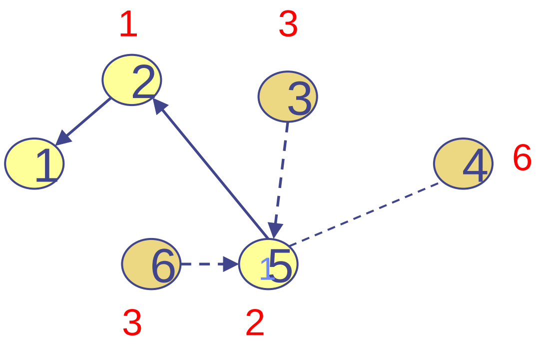
Dijkstra

3

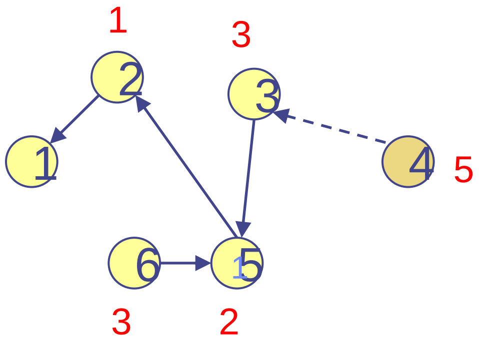




P(2)={1,2}
D(1,2)=1

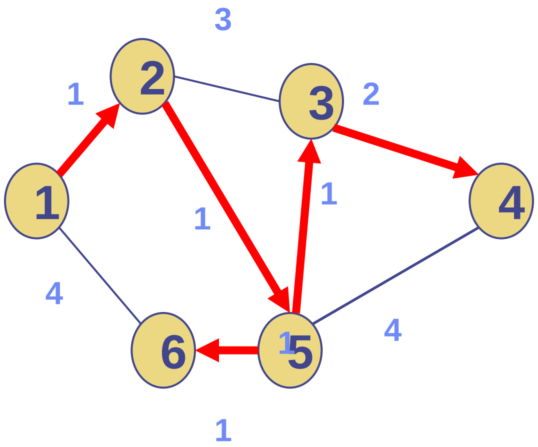


P(3)={1,2,5}
D(1,5)=2



P(4)={1,2,3,5,6}
D(1,3)=3
D(1,6)=3

Dijkstra - Forwarding Table



At node 5

	Outgoing	Cost
1	2	2
2	2	1
3	3	1
4	3	3
6	6	1

Centralized vs. distributed algorithms – Link-state routing

- Dijkstra’s algorithm nice and well. But how to obtain centralized view of the entire network to be able to apply Dijkstra’s algorithm?
  - Assumption: only direct neighbors know the (current) cost of a link or know whether a link has failed/been restored/upgraded/…

- Solution: Have each node distribute this information – state of all its links – in the entire network
  - Then, all nodes know entire network topology & can apply Dijkstra’s algorithm
  - Distribution itself can happen via flooding

- Link-state routing
  - Intuition: Little information (about direct neighbors) is spread over large distances (to the entire network)

OSPF: Topology update – Flooding!!!

[steenkiste]

Transmit link state advertisements

- Originating router
  - Typically, minimum IP address for router
- Link ID *(Link ID and metric repeated for each link)*
  - ID of router at other end of link
- Metric
  - Cost of link
- Link-state age
  - Incremented each second
  - Packet expires when reaches 3600
- Sequence number
  - Incremented each time sending new link information

Cost of the link might be split according to different ***Type of Service*** *metrics:*
e.g. high for delay, low for bit error rate...

OSPF: Topology update – processing

[steenkiste]

- Node X Receives LSA from Node Y
  - With Sequence Number q
  - Looks for entry with same origin/link ID
- Cases
  - No entry present
    - Add entry, propagate to all neighbors other than Y
  - Entry present with sequence number p < q
    - Update entry, propagate to all neighbors other than Y
  - Entry present with sequence number p > q
    - Send entry back to Y
    - To tell Y that it has out-of-date information
  - Entry present with sequence number p = q
    - Ignore it

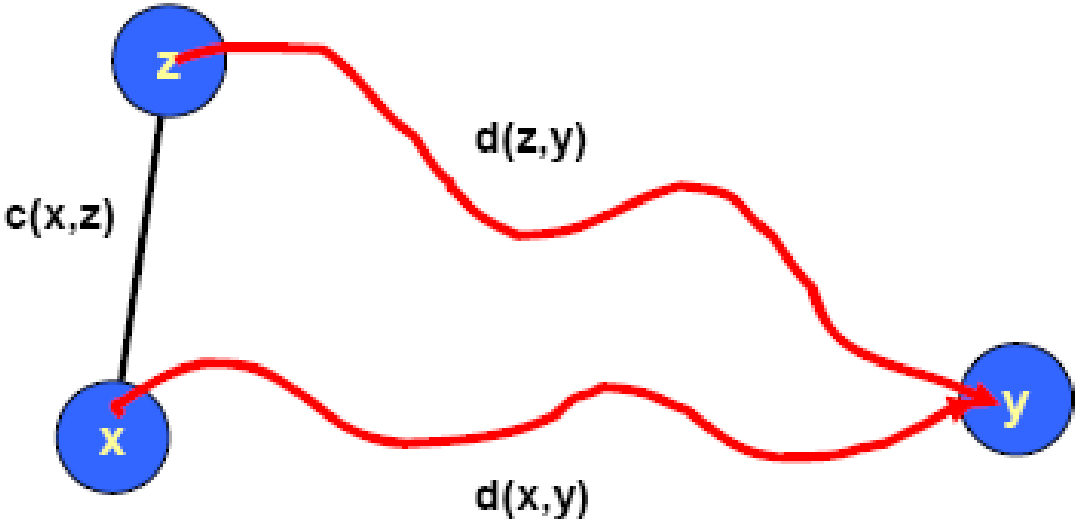
Alternative approach: Distance-vector routing

- Alternative idea to link state routing: Distribute lot's of information over short distances
  - Distribute everything a node currently knows (or believes) about the entire network topology, but only to direct neighbors
  - This information is represented by the routing table (containing outgoing link and cost)
    - If reduced to cost only, also called a distance vector
  - Invented by Bellman & Ford (1957)

- After receiving a routing table from a neighbor, compare whether it contains "good news", i.e., a shorter route than the one currently known
  - Assumption: each router knows cost to each of its direct neighbors

- In practice: It suffices to exchange distance vectors

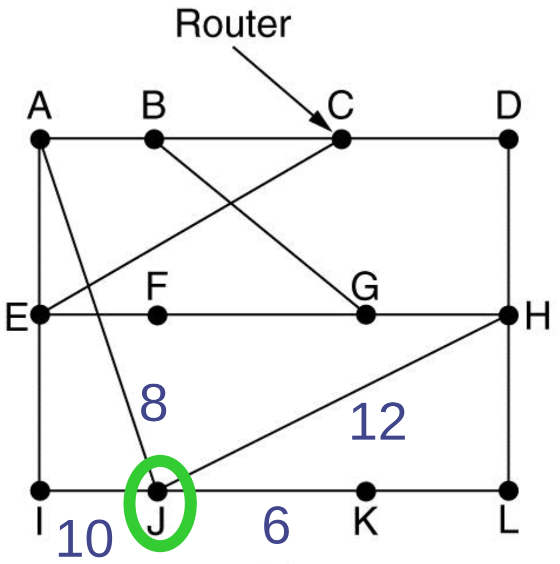
Bellman-Ford Principle



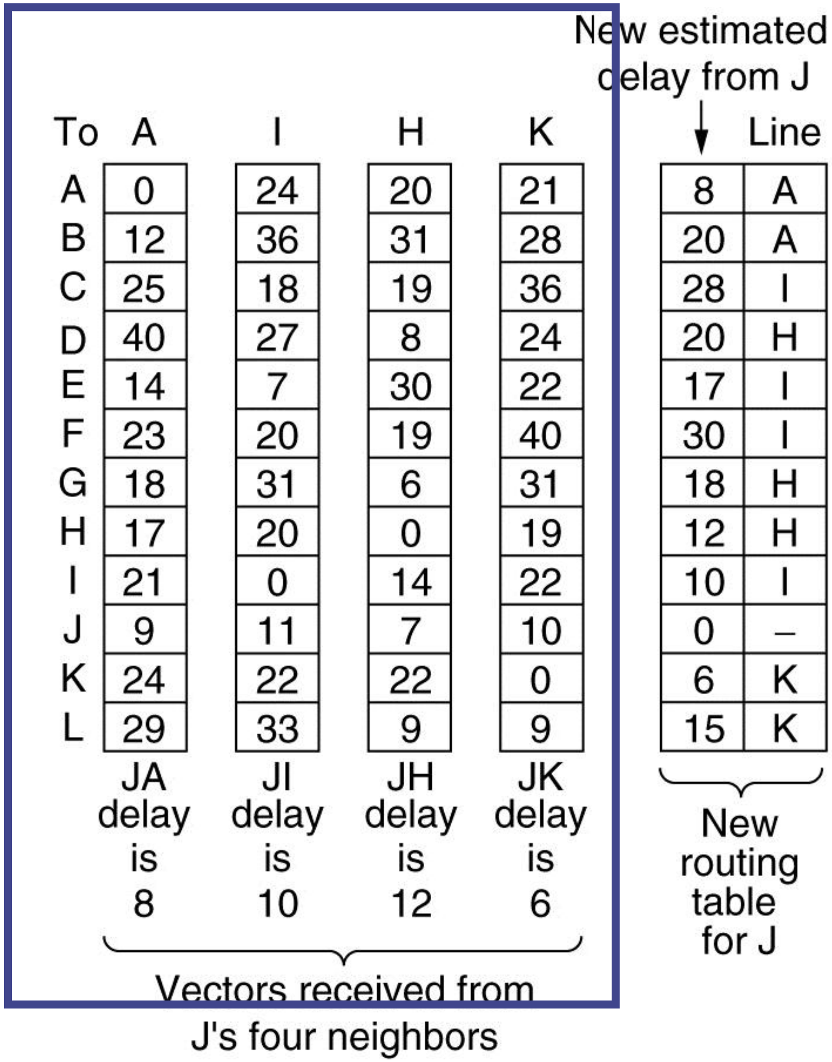
- Update(x,y,z)

```
d ← c(x,z) + d(z,y)              # Cost of path from x to y with first hop z
if d < d(x,y)
    # Found better path
    return d,z                   # Updated cost / next hop
else
    return d(x,y), nexthop(x,y)          # Existing cost / next hop
```

Distance-vector routing – Example



This is the (current version of) routing table!



To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6		

Vectors received from J's four neighbors

New routing table for J

Problems, Problems…

- IP offers unreliable service between COMPUTERS
- Do we need something more
  - Well, we might like to reach individual processes…(UDP)
  - We might want to increase the reliability… and more (TCP)