Network Layer

Computer Networks

Network Layer Topics

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What is inside a router?
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 ICMP

K & R

Network Layer Introduction

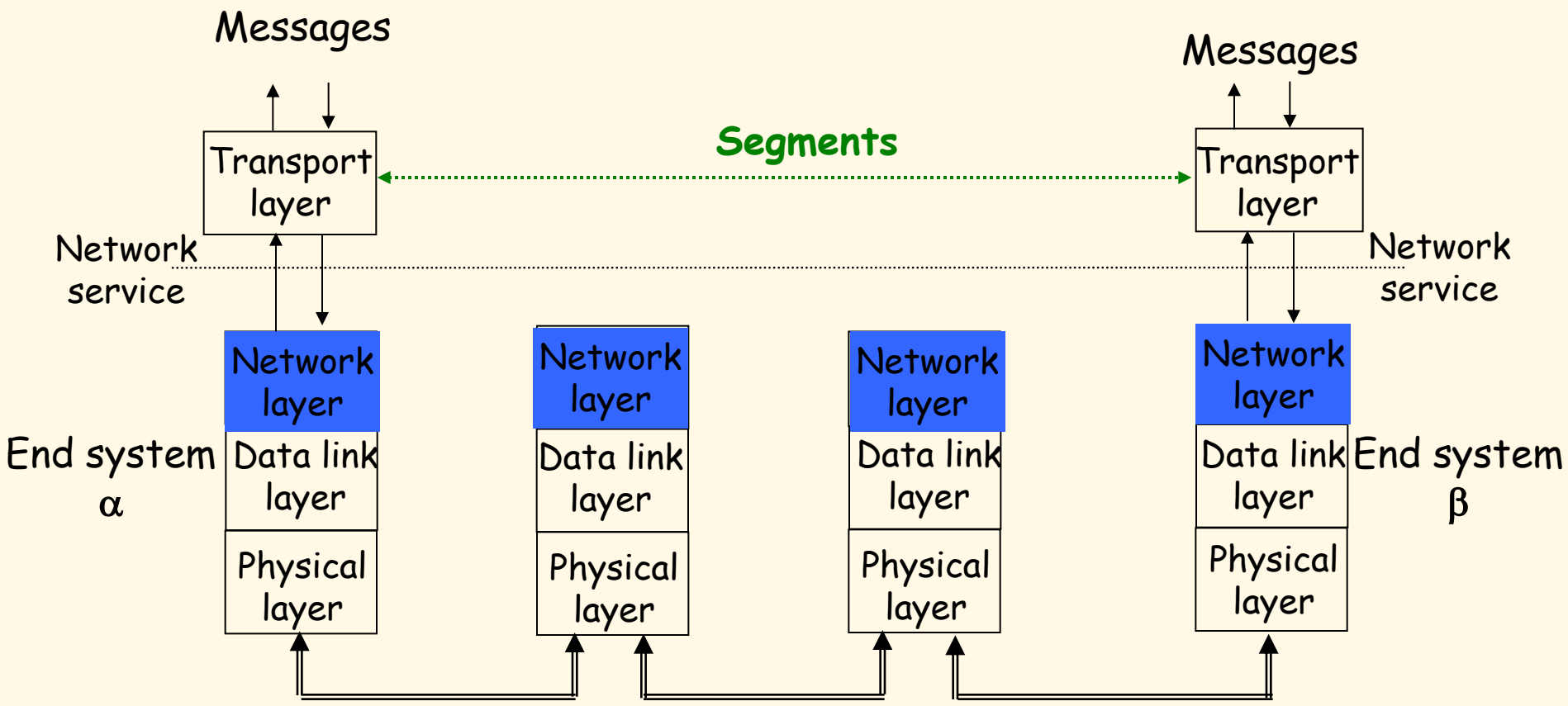
- Concerned with getting packets from source to destination.
- The network layer must know the topology of the subnet and choose appropriate paths through it.
- When source and destination hosts are in different networks, the network layer (IP) must deal with these differences.
- Key issue: What service does the network layer provide to the transport layer?

  connection-oriented or connectionless

Network Layer Design Goals

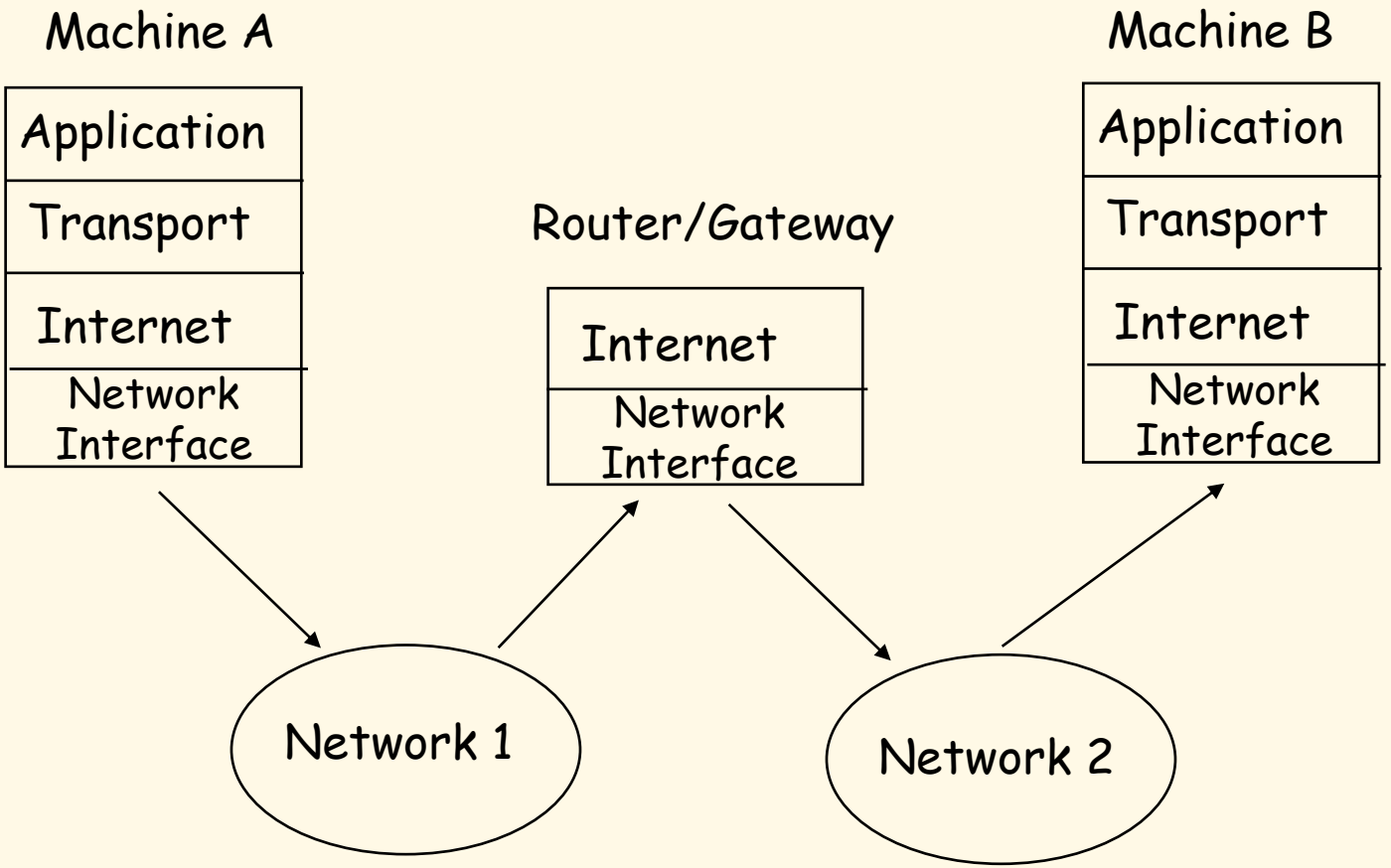
1. The services provided by the network layer should be independent of the subnet topology.
2. The transport layer should be shielded from the number, type and topology of the subnets present.
3. The network addresses available to the transport layer should use a uniform numbering plan (even across LANs, WANs and WLANs).

Network Layer



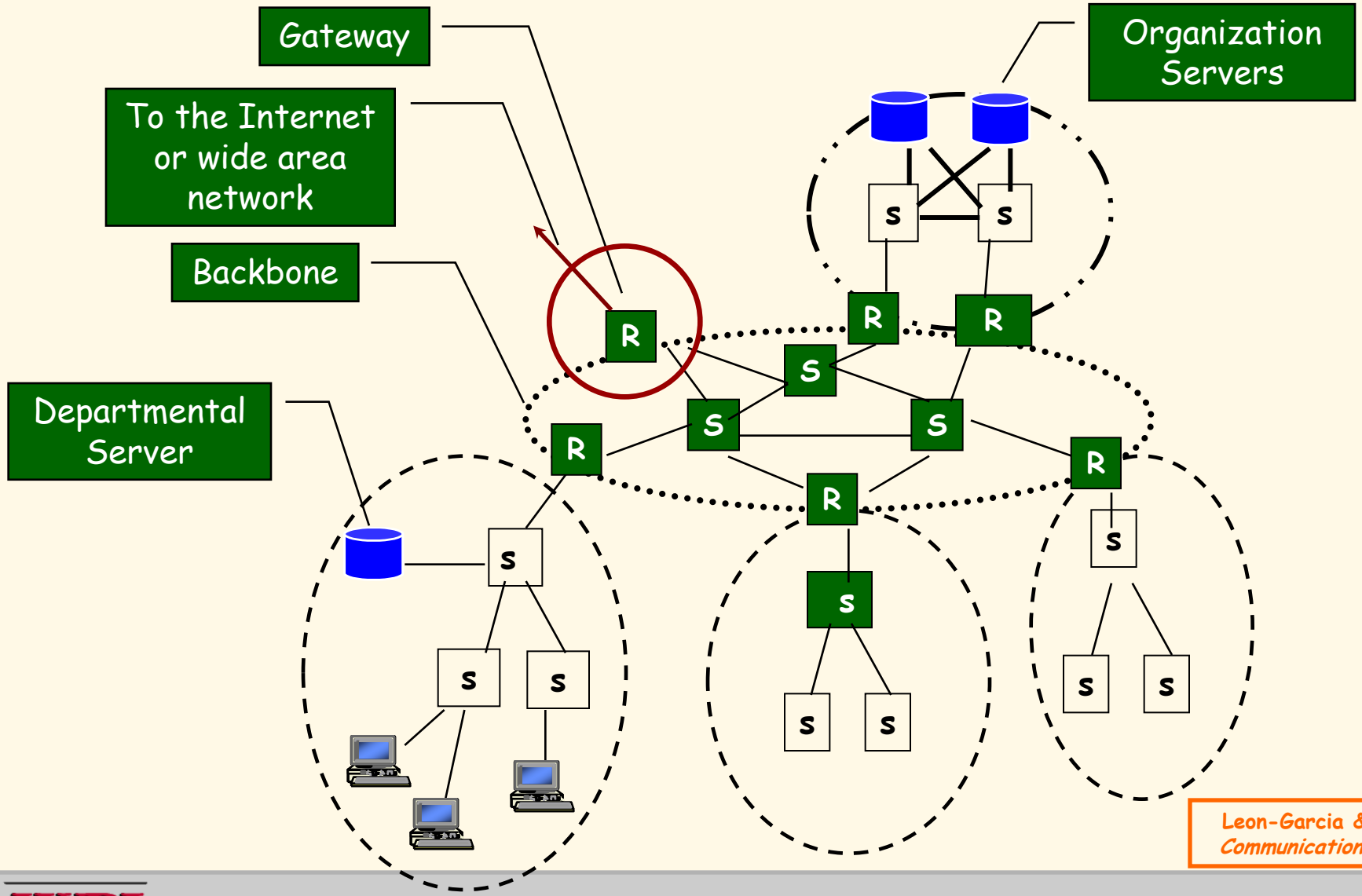
Leon-Garcia & Widjaja: *Communication Networks*

Network Layer



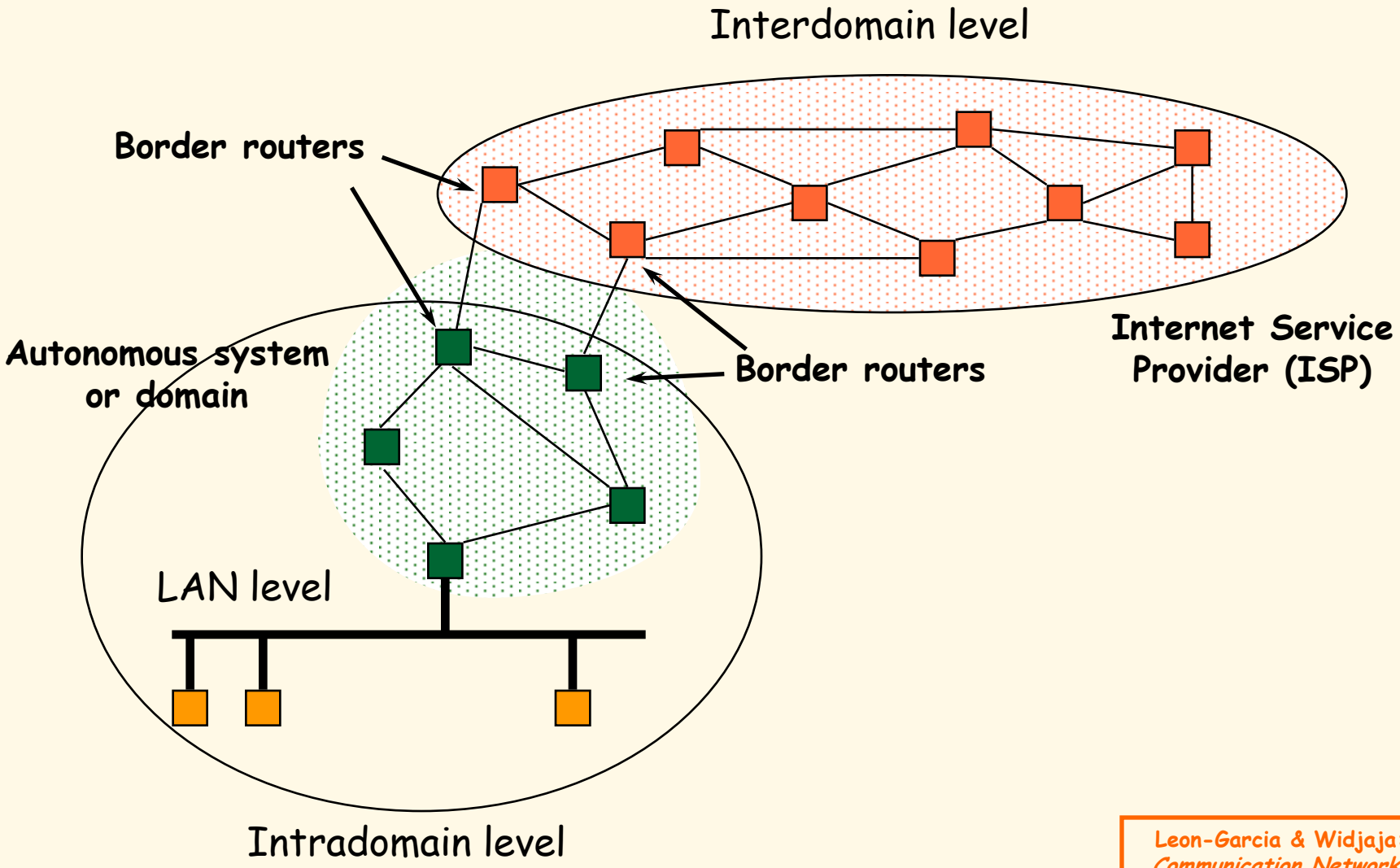
Leon-Garcia & Widjaja: *Communication Networks*

Metropolitan Area Network (MAN)



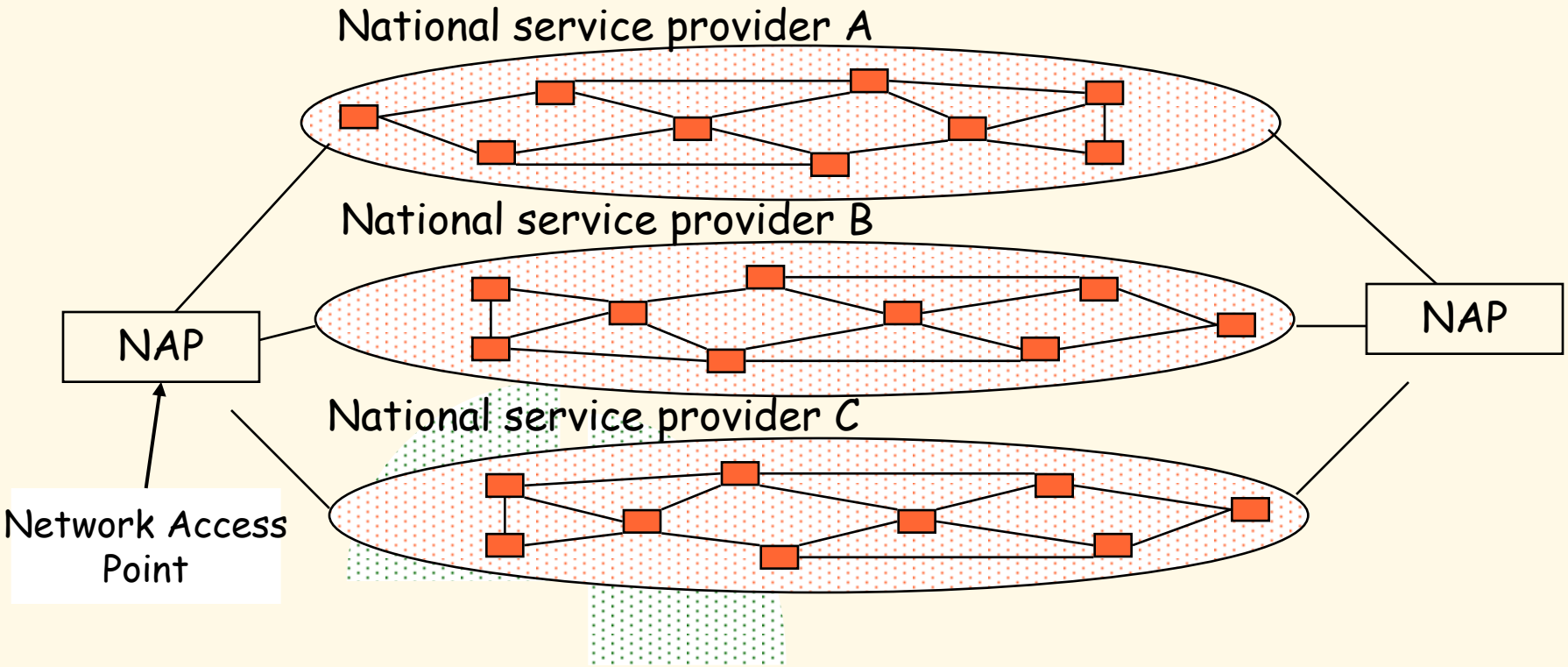
Leon-Garcia & Widjaja: *Communication Networks*

Wide Area Network (WAN)



Leon-Garcia & Widjaja: *Communication Networks*

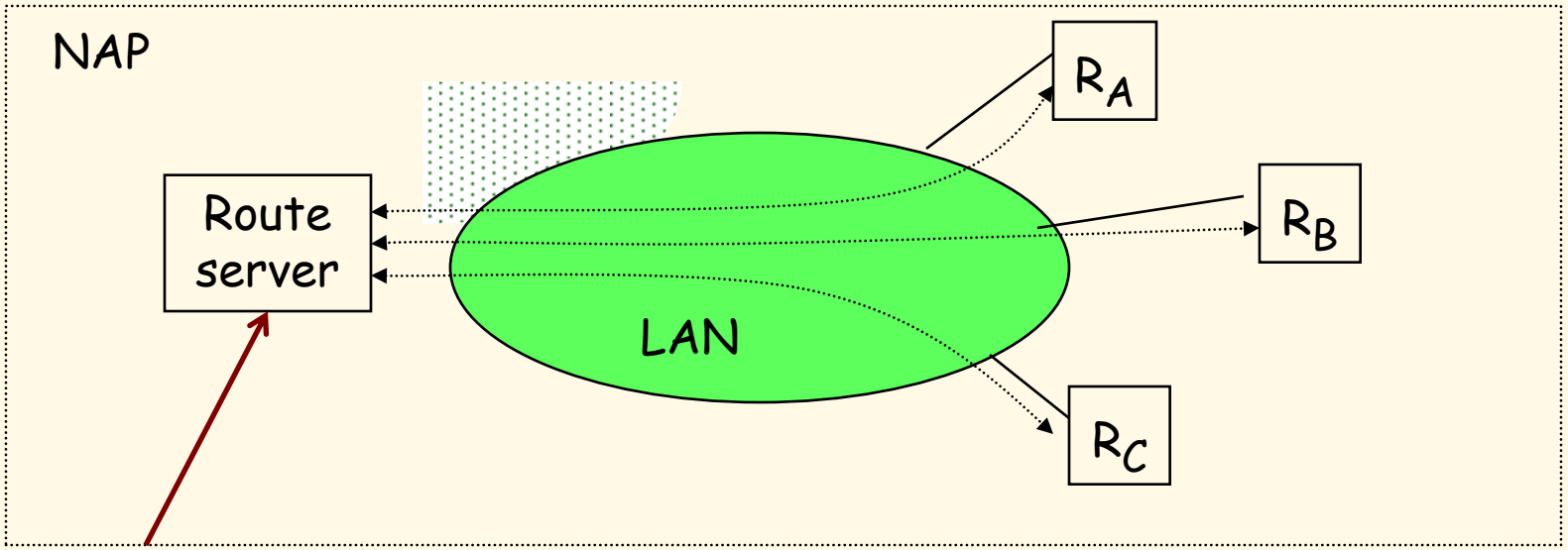
Modern Internet Backbone



National Internet Service Providers

Leon-Garcia & Widjaja: *Communication Networks*

Network Access Point



often called
the default router

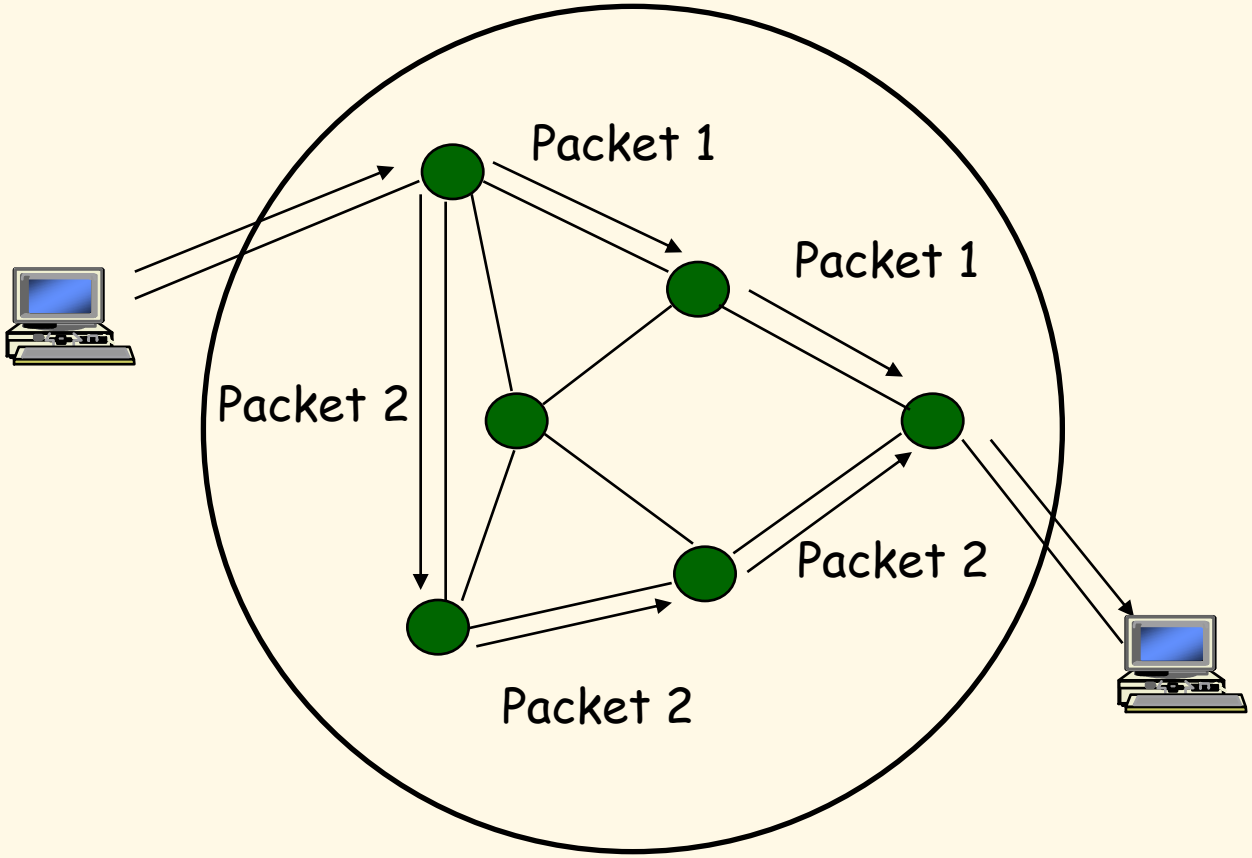
Leon-Garcia & Widjaja:
Communication Networks

Network Layer Topics

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What is inside a router?
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

K & R

Datagram Packet Switching



Leon-Garcia & Widjaja:
Communication Networks

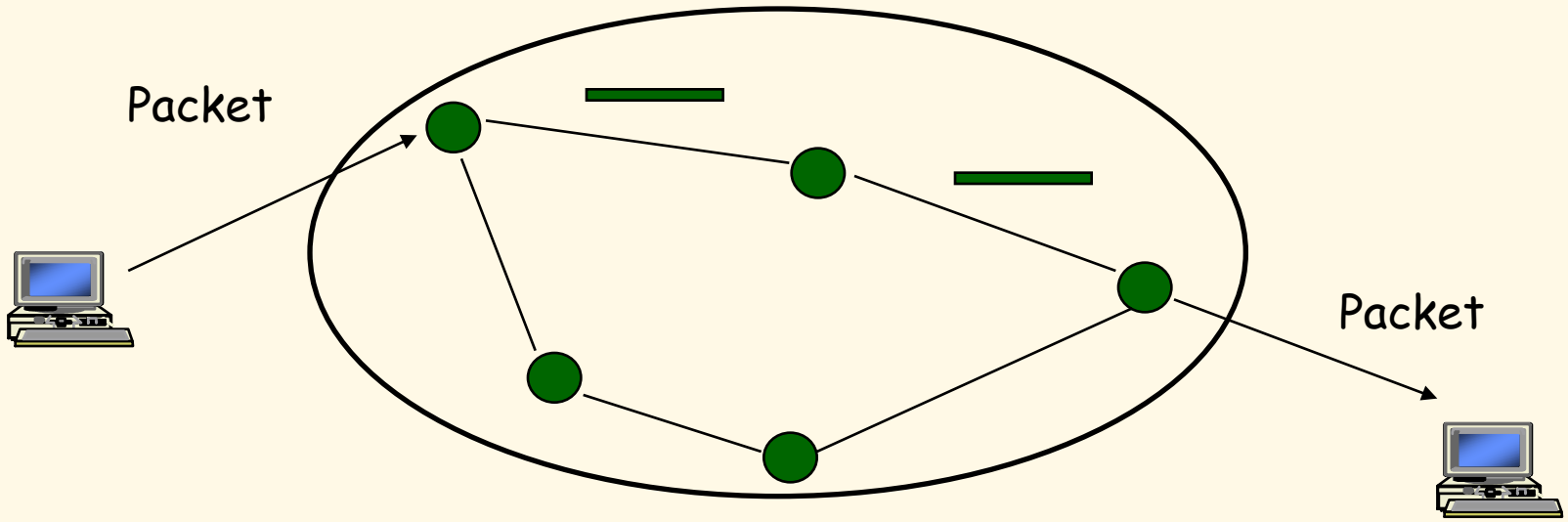
Datagram Routing Table

IP addresses

Destination address	Output port
0785	7
1345	12
1566	6
2458	12

Leon-Garcia & Widjaja: *Communication Networks*

Virtual Circuit Packet Switching



Leon-Garcia & Widjaja:
Communication Networks

Virtual Circuit Routing Table

Identifier	Output port	Next identifier
12	13	44
15	9	23
27	13	16
58	7	34

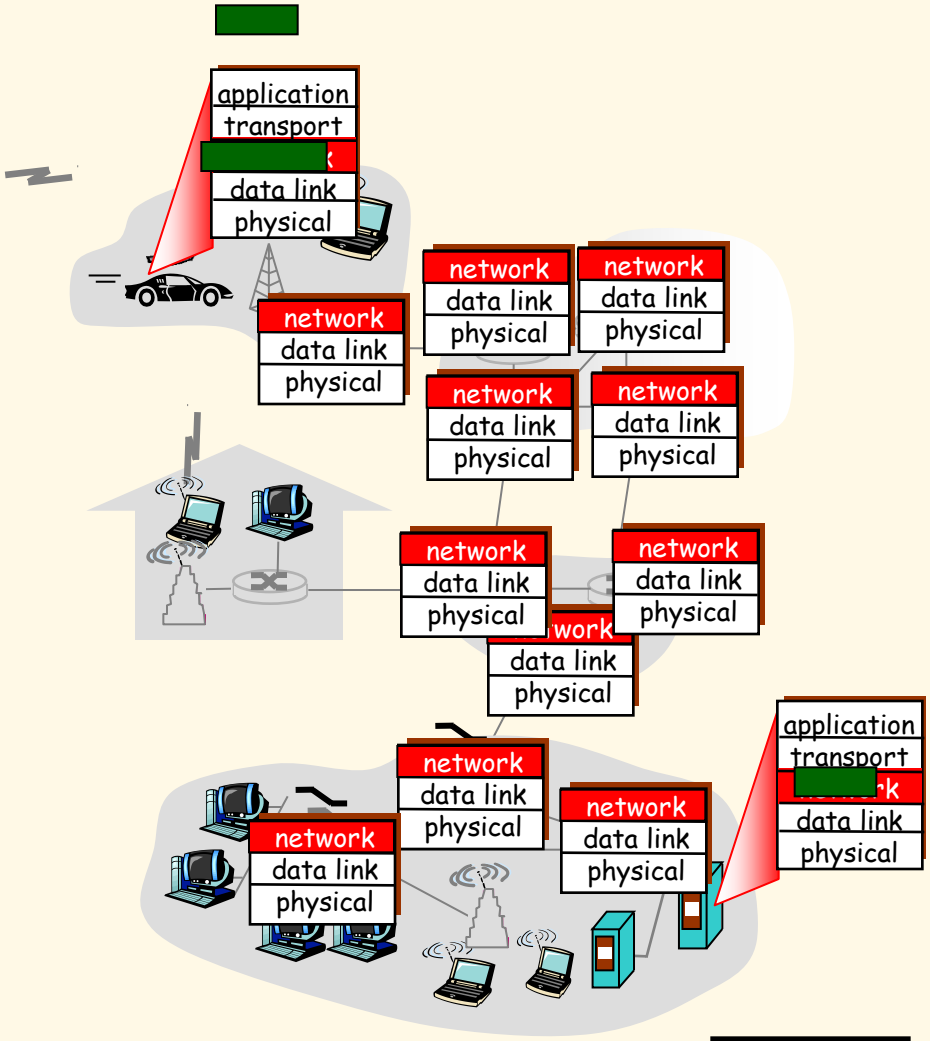
Entry for packets with identifier 15 → (row 15)

← Packet leaves with new identifier 23 (row 15)

Leon-Garcia & Widjaja: *Communication Networks*

Network Layer

- transports **segment** from sending to receiving host.
- encapsulates **segments** on sending side into datagram **packets**.
- delivers **segments** on receiving side to the transport layer.
- network layer protocols exist in *every* host, router.
- router examines header fields in all IP datagrams passing through it.



K & R

Network Layer Topics

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What is inside a router?
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 ICMP

K & R

Two Key Network Layer Functions

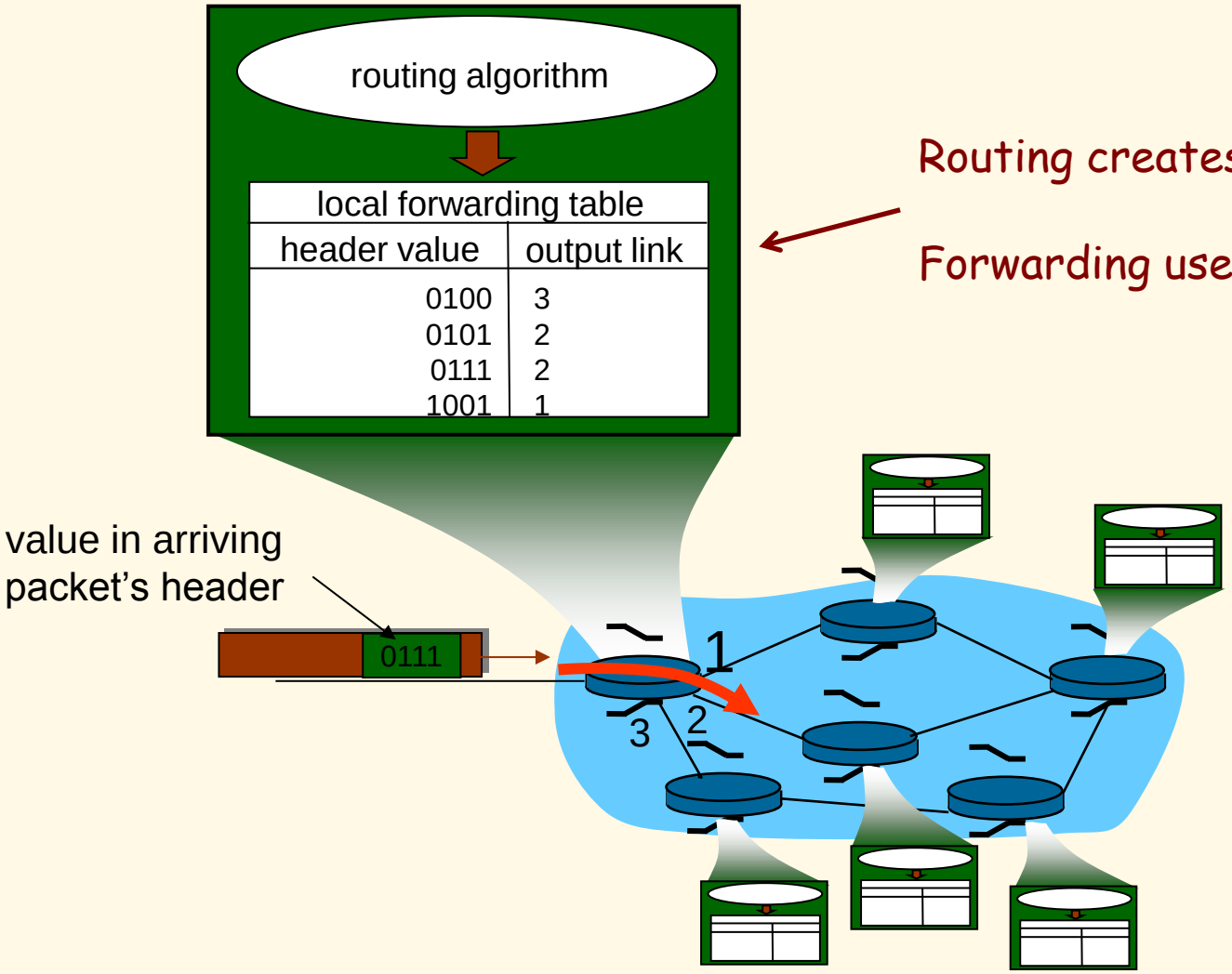
- *forwarding:* move packets from router's input to appropriate router output.
- *routing:* determine route taken by packets from source to destination.

analogy:

- **forwarding:** process of getting through single interchange
- **routing:** process of planning trip from source to destination

K & R

Interplay between Routing and Forwarding

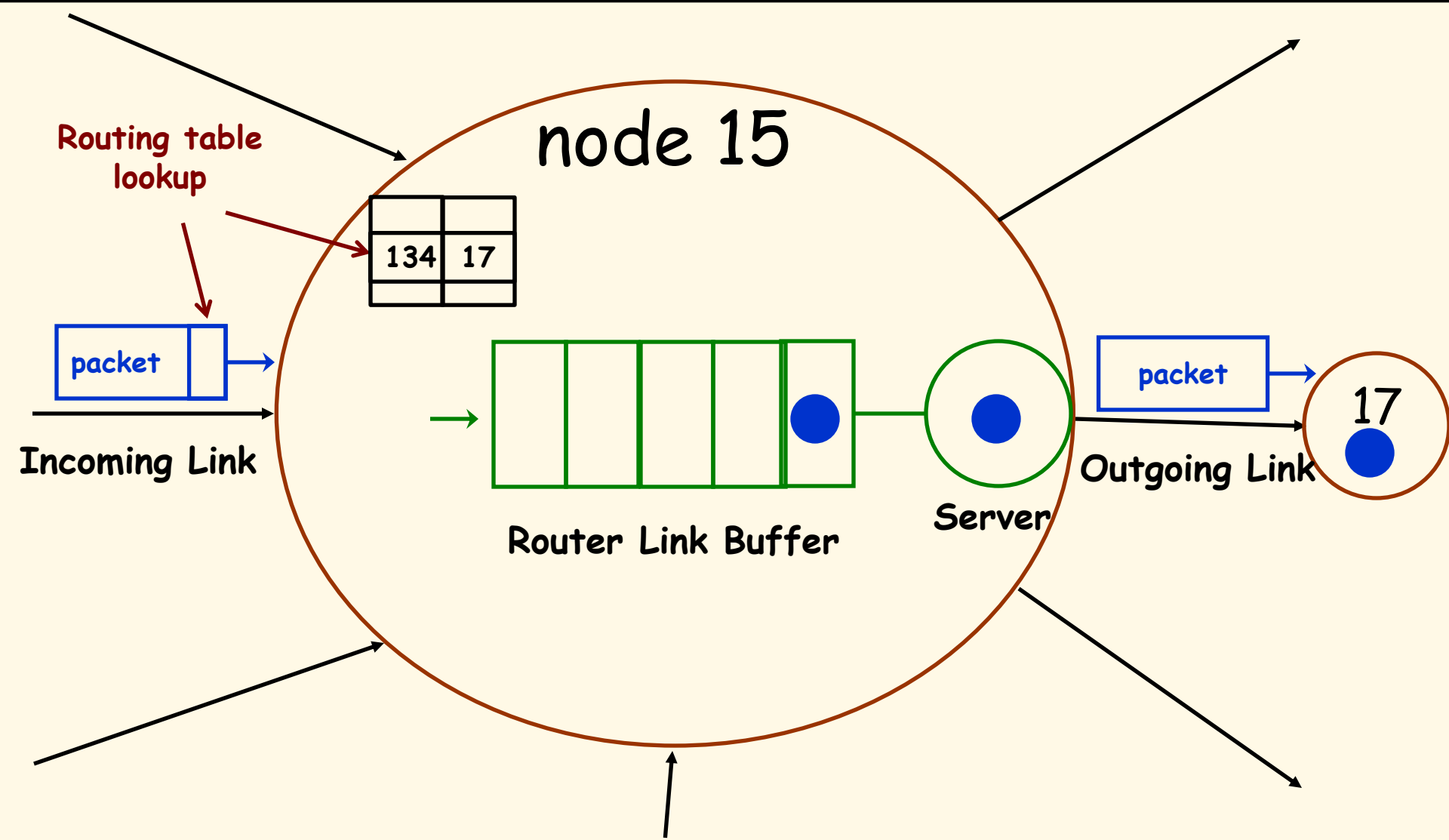


Routing creates the tables.

Forwarding uses the tables.

K & R

Router Node Forwarding



Routing in an internet

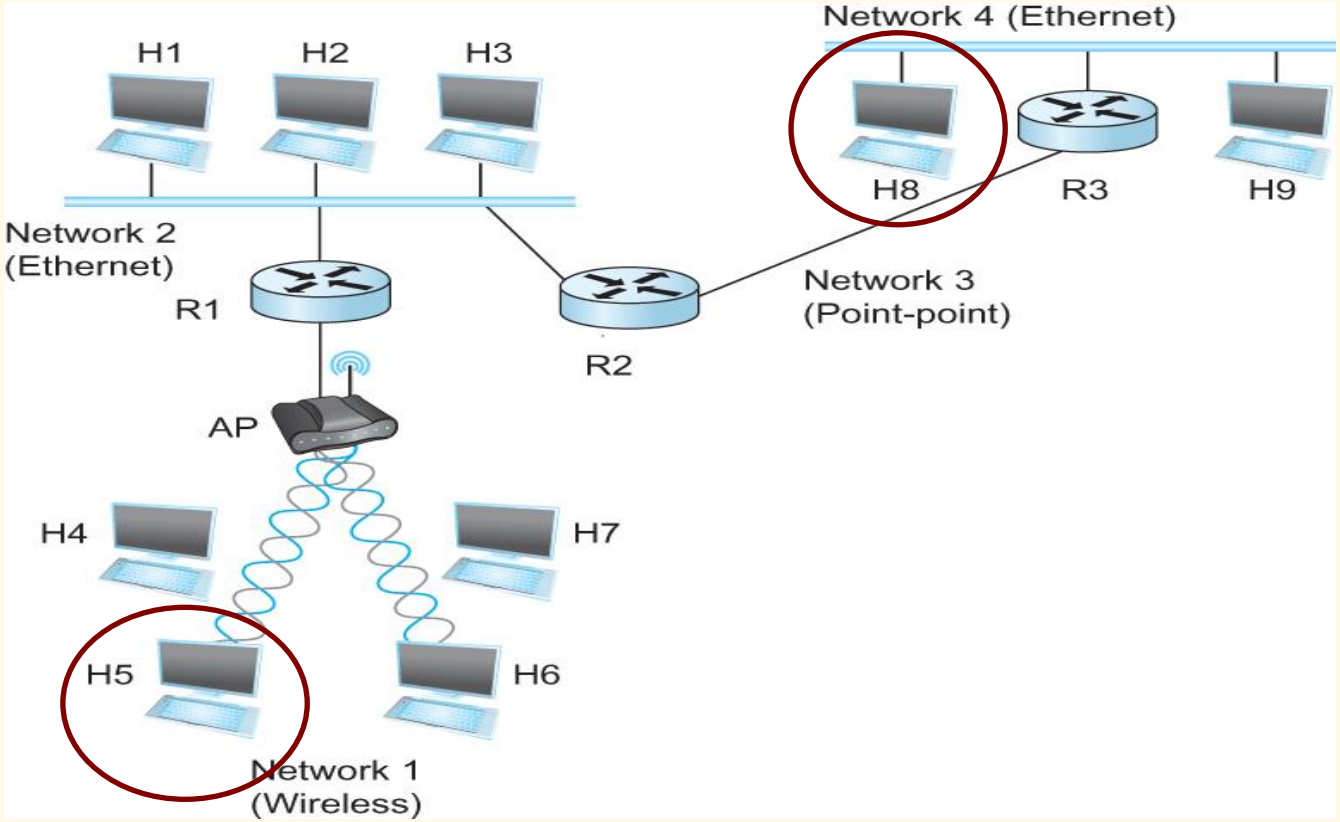


Figure 3.14 A Simple internetwork with Three Routers

Protocol Layers along the Route

- IP runs on all the nodes in a collection of networks and defines the infrastructure that allows these nodes and networks to function as a single logical internetwork.

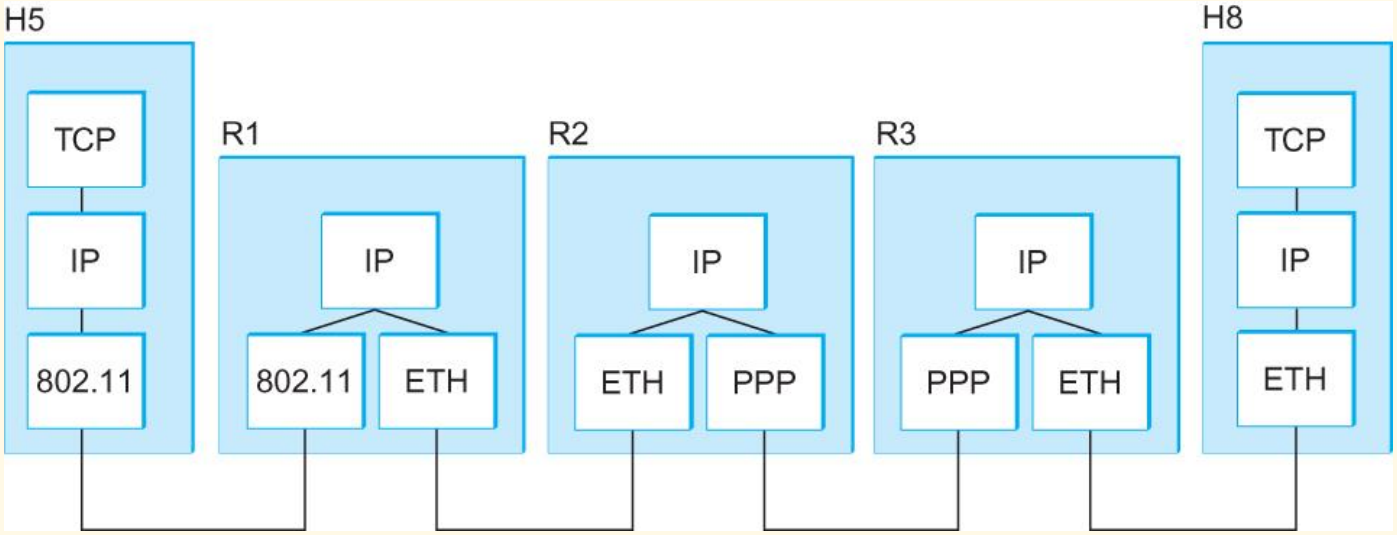


Figure 3.15 Protocol Layers used for a Simple internet

Forwarding Table Example

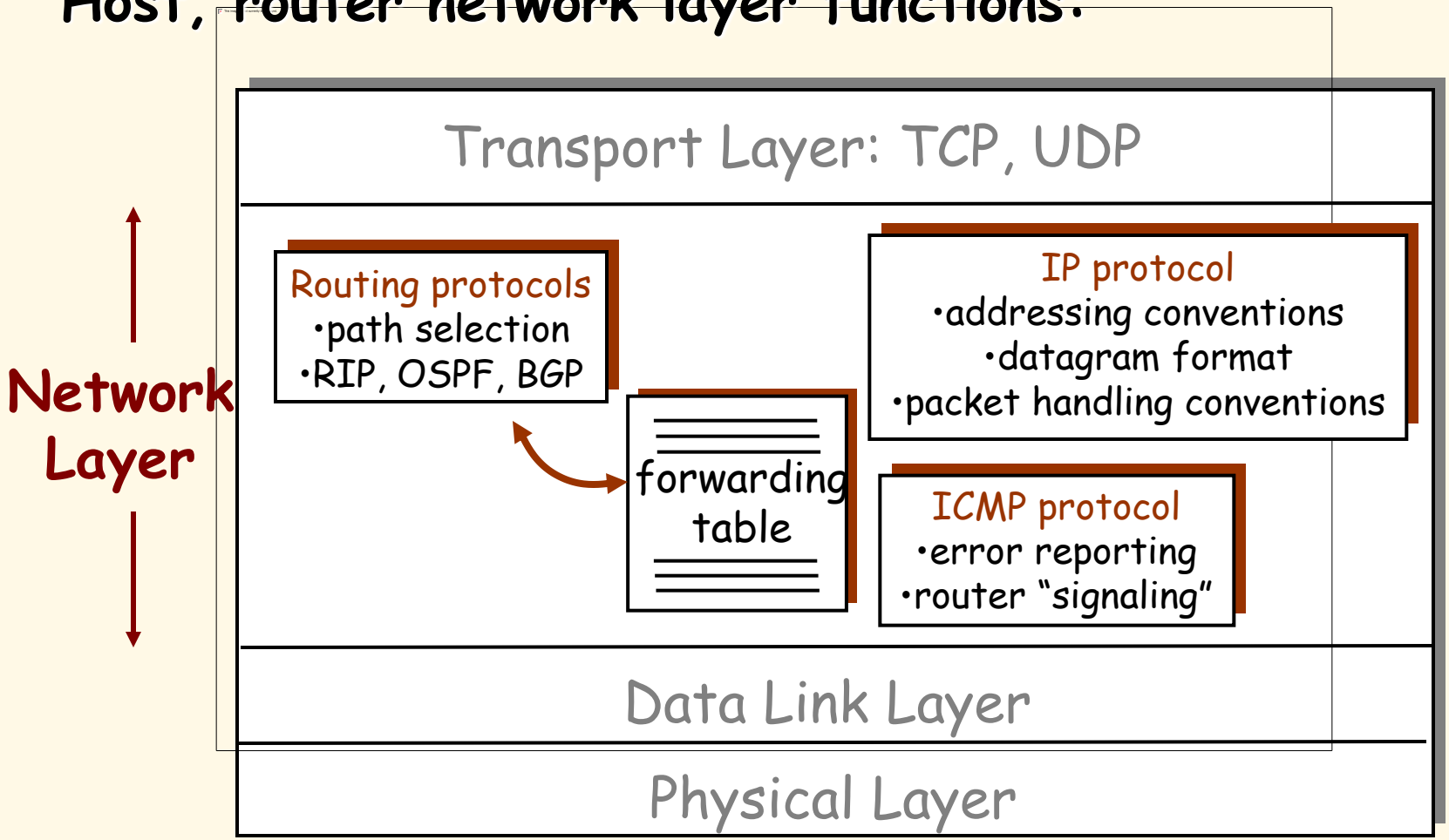
Table 3.6 Complete Forwarding Table for Router R2 in Figure 3.14

Network Number	Next Hop
1	R1
2	Interface 1
3	Interface 0
4	R3

Note – As R2 is on Network 2 and Network 3, this table shows packets headed for H1, H2 and H3 are not forwarded by R2 to another router.

The Internet Network Layer

Host, router network layer functions:



K & R

Network Layer Topics

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What is inside a router?
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 ICMP

K & R

IPv4 Datagram Format

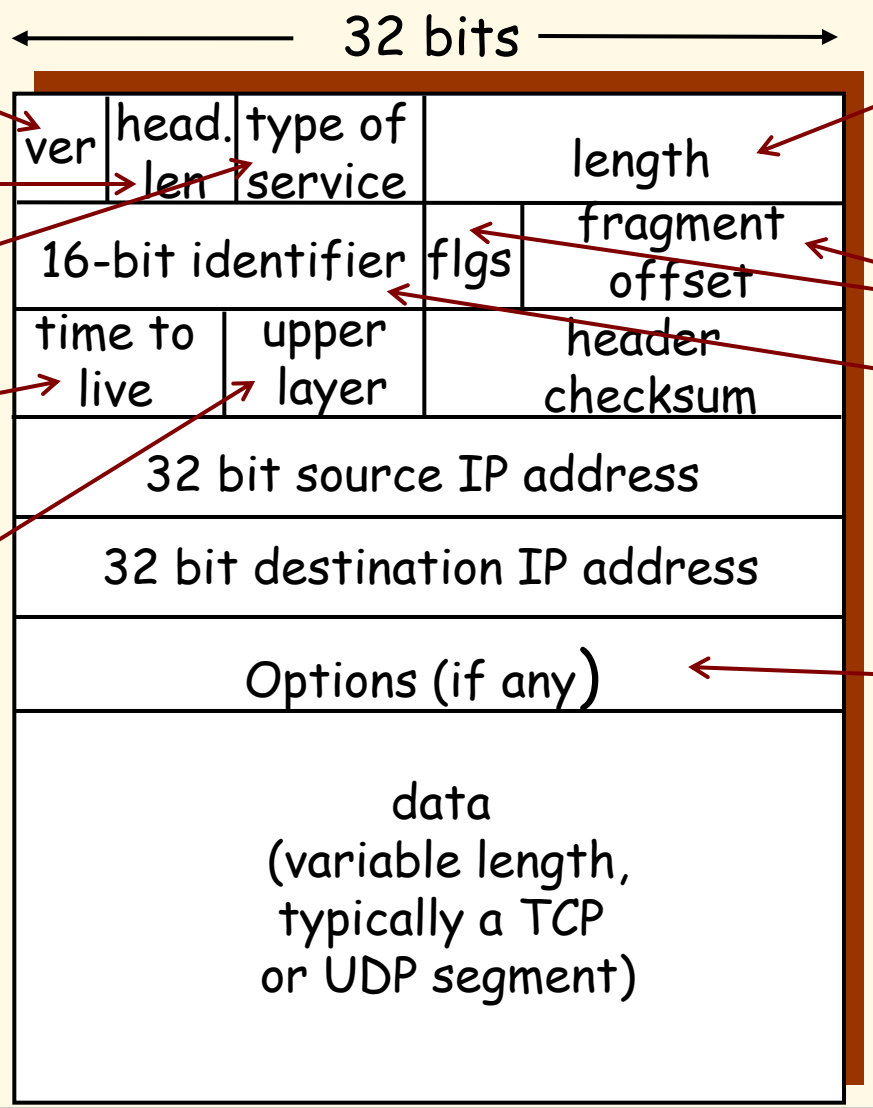
IP protocol version number

header length (bytes)

TOS:: data type

TTL:: max hops remaining (each router decrements)

upper layer protocol to deliver payload to



total datagram length (bytes)

Three fields for fragmentation/ reassembly

E.g. timestamp, record route taken, specify list of routers to visit.

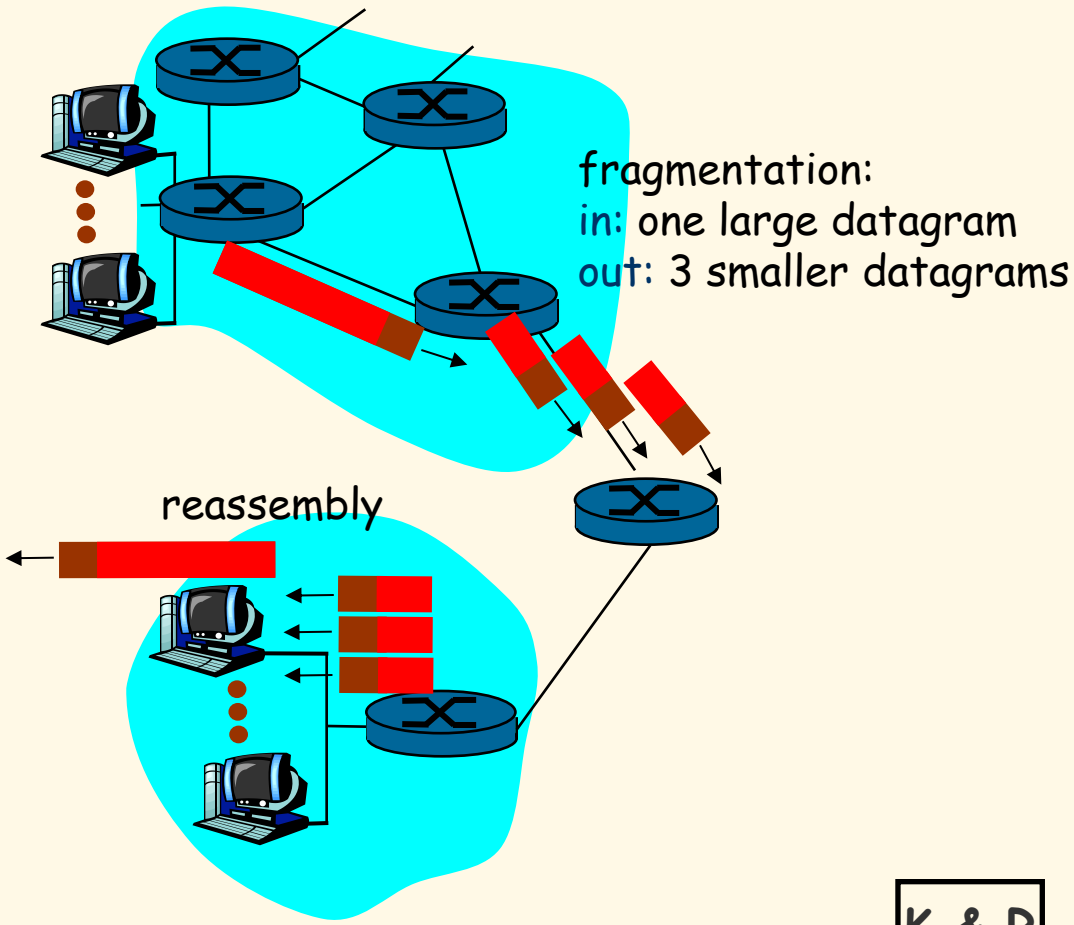
how much overhead with TCP?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

K & R

IP Fragmentation & Reassembly

- network links have MTU (max.transfer size) - largest possible link-level frame.
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams.
  - "reassembled" only at final destination.
  - IP header bits used to identify, order related fragments.



K & R

IP Fragmentation & Reassembly

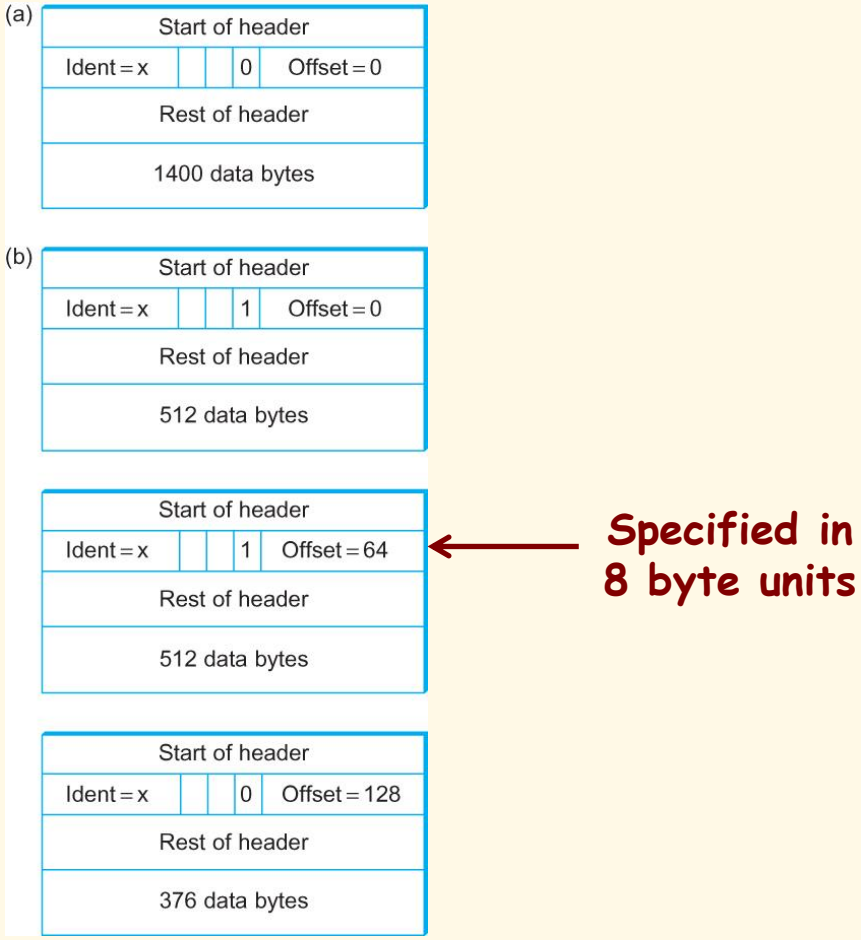


Figure 3.18 Header fields used in IP fragmentation: (a) unfragmented packet; (b) fragmented packets.

IP Fragmentation for Figure 3.14

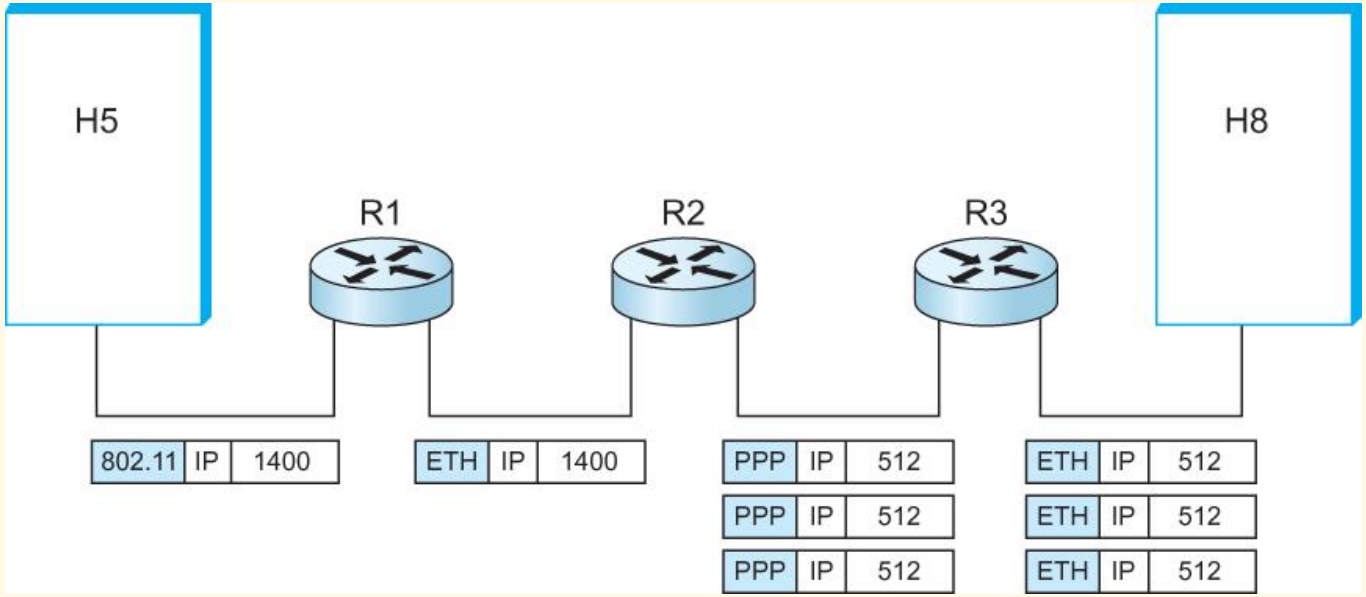


Figure 3.17 IP datagrams traversing the sequence of physical networks graphed in Figure 3.14.

Network Layer Topics

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What is inside a router?
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

K & R

Global Internet

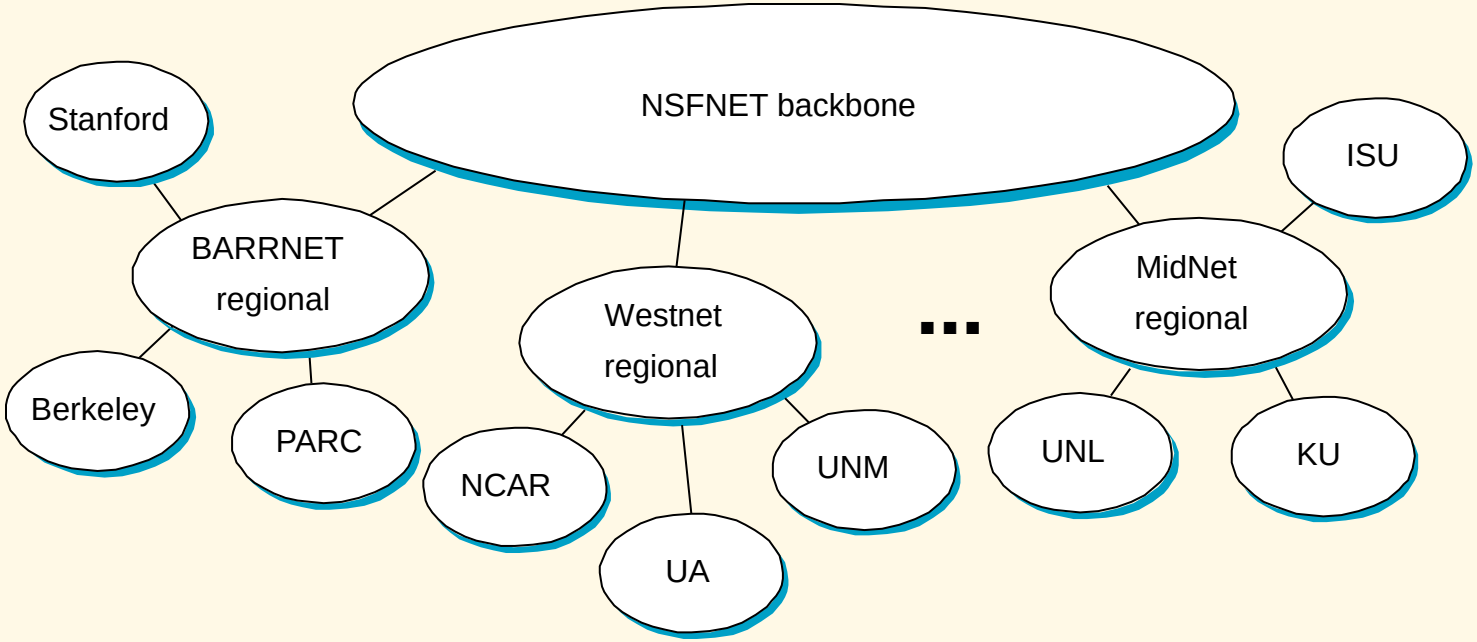


Figure 4.1 The tree structure of the Internet in 1990

Global Internet

- Each provider network is regional and a single autonomous system (AS)
- Major issues are:
  - Scalability of routing
  - Address utilization
    - Now out of IPv4 addresses
- Hierarchy is used to improve scalability.
  - Namely, utilize subnets with masks.

IPv4 Network Classes

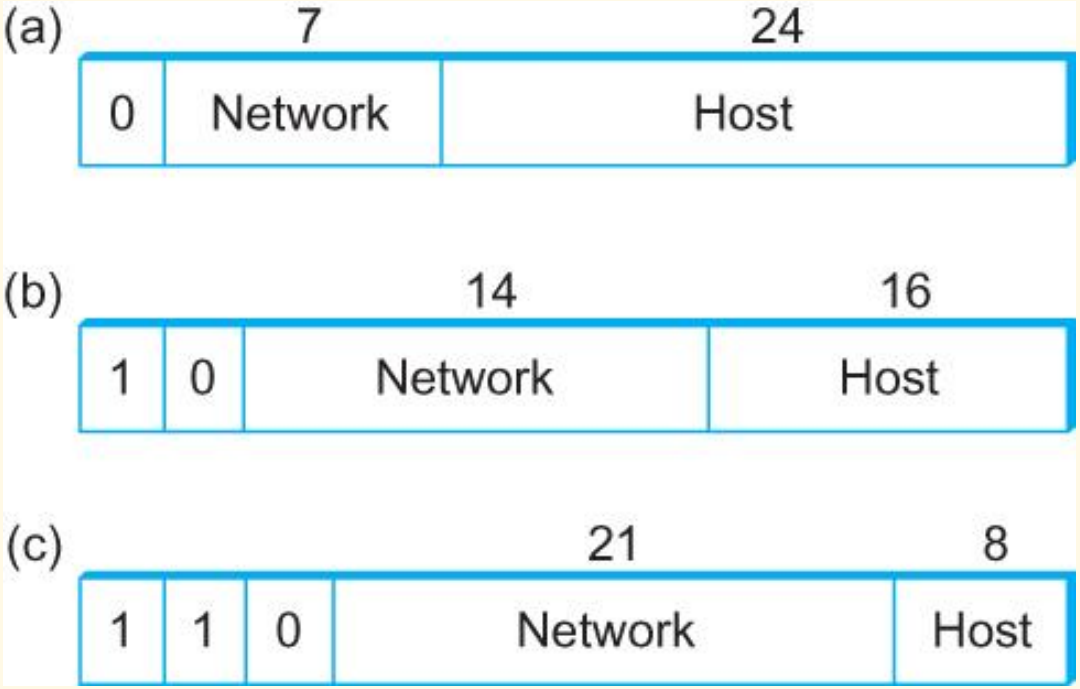


Figure 3.19 IP addresses: (a) class A; (b) class B; (c) class C.

Network Number Problems

- Assigning one network number per physical network uses up IP address too fast!
- Adding more network numbers also increases forwarding table size.

Subnet solution ::

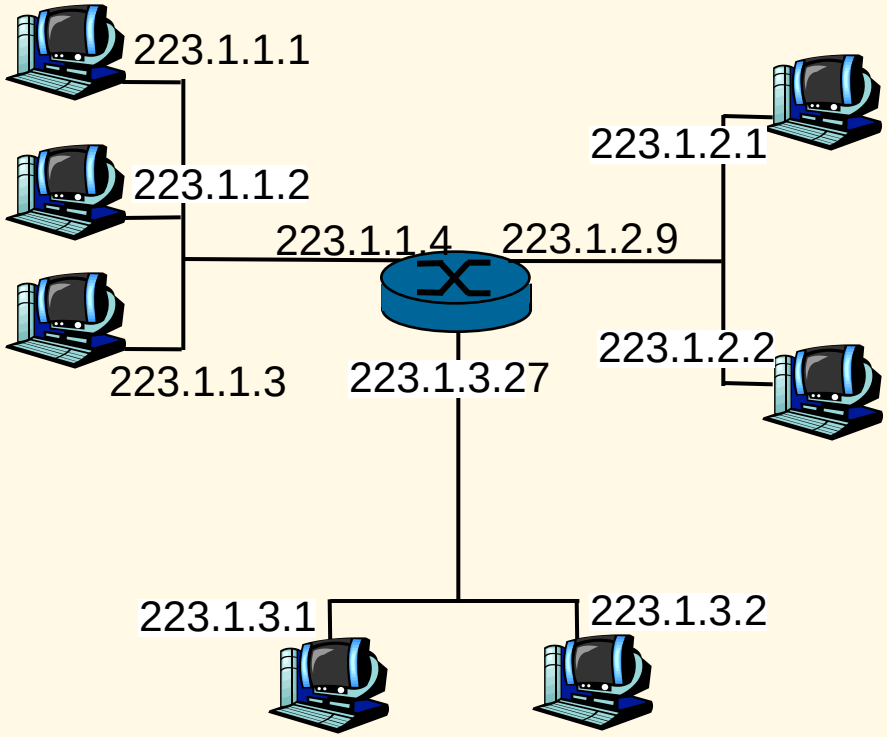
- The idea is to take a single IP network number and allocate the IP addresses with that network number to several physical networks which are referred to as subnets.
- The subnets need to be physically close to each other for routing purposes.

Subnetting

- The mechanism by which a single network number can be shared among multiple networks involves configuring all the nodes on each subnet with a subnet mask.
- The subnet mask enables introduction of a single subnet number which provides for another level of hierarchy into the IP address.
- All hosts on a given subnet are configured with the same mask, i.e., there is one subnet mask per subnet.

IP Addressing: Introduction

- **IP address::** 32-bit identifier for host, router *interface*.
- **interface::** connection between host/router and physical link
  - router's typically have multiple interfaces.
  - host typically has one interface.
  - IP addresses are associated with each interface.



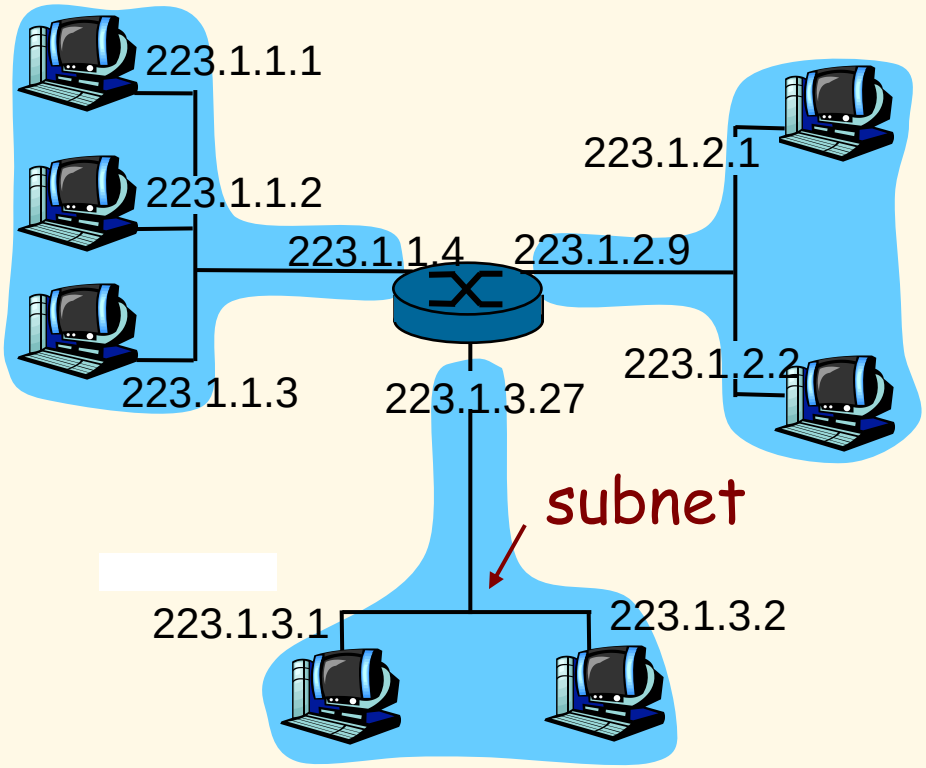
223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

K & R

Subnets

- **IP address:**
  - subnet part (high order bits)
  - host part (low order bits)
- **What is a subnet ?**
  - device interfaces with same subnet part of IP address.
  - can physically reach each other without an intervening router.



network consisting of **three** subnets

K & R

Subnet Masks

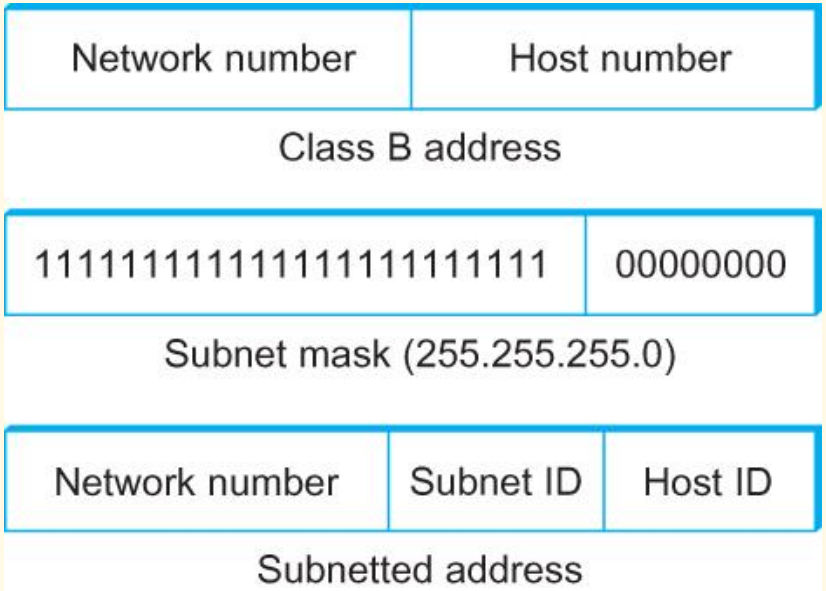


Figure 3.20 Subnet Addressing

Subnetting Example

Figure 3.21
An Example of Subnetting

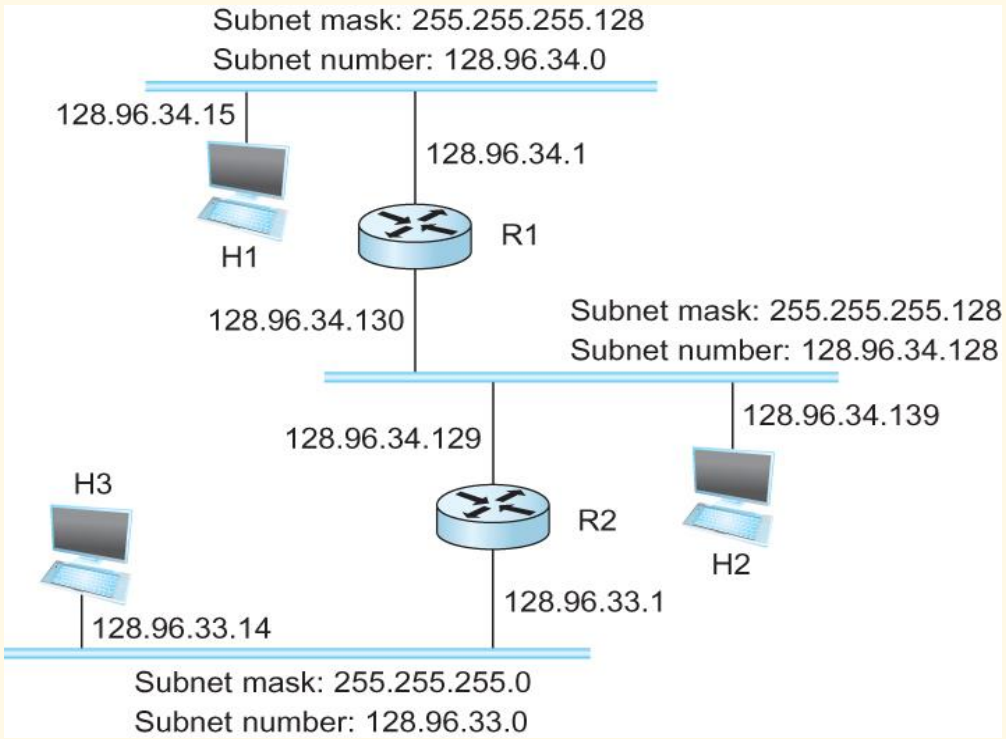


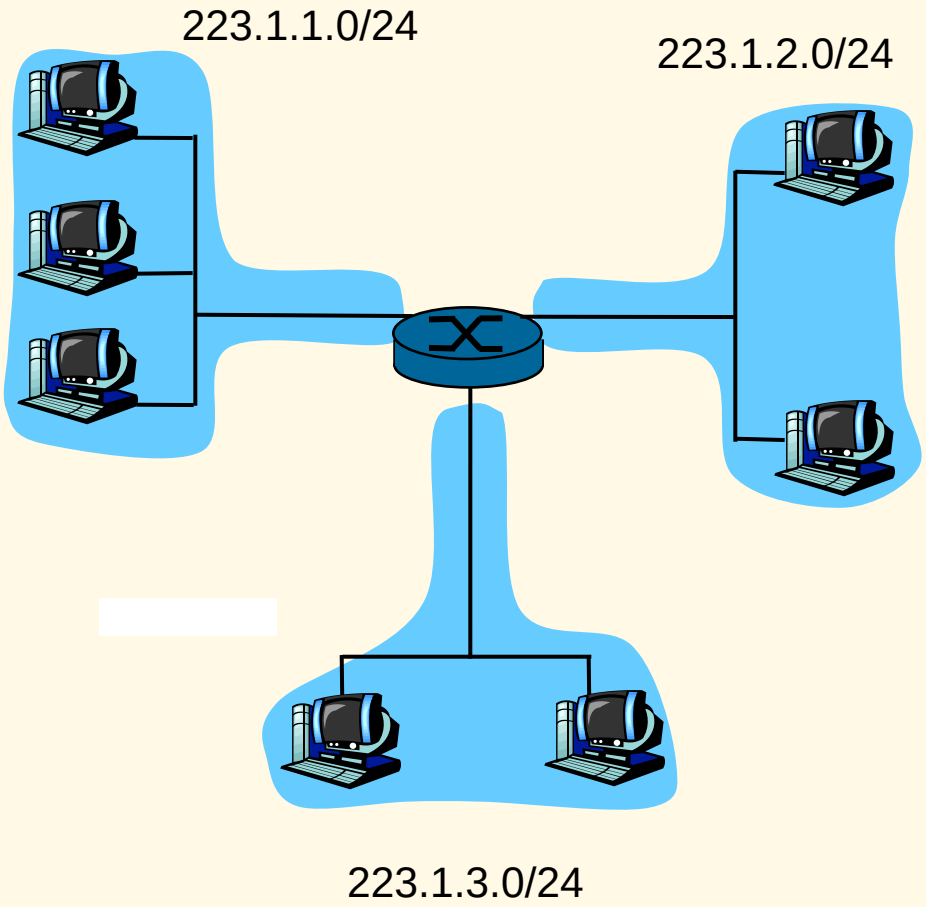
Table 3.7
Forwarding Table at Router R1

SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2

Subnets

Subnet Concepts

- To determine the subnets, detach each interface from its host or router, creating islands of isolated networks. Each isolated network is called a subnet.
- Sending host does bitwise AND between subnet mask and destination address to determine whether packet needs to be routed or not.

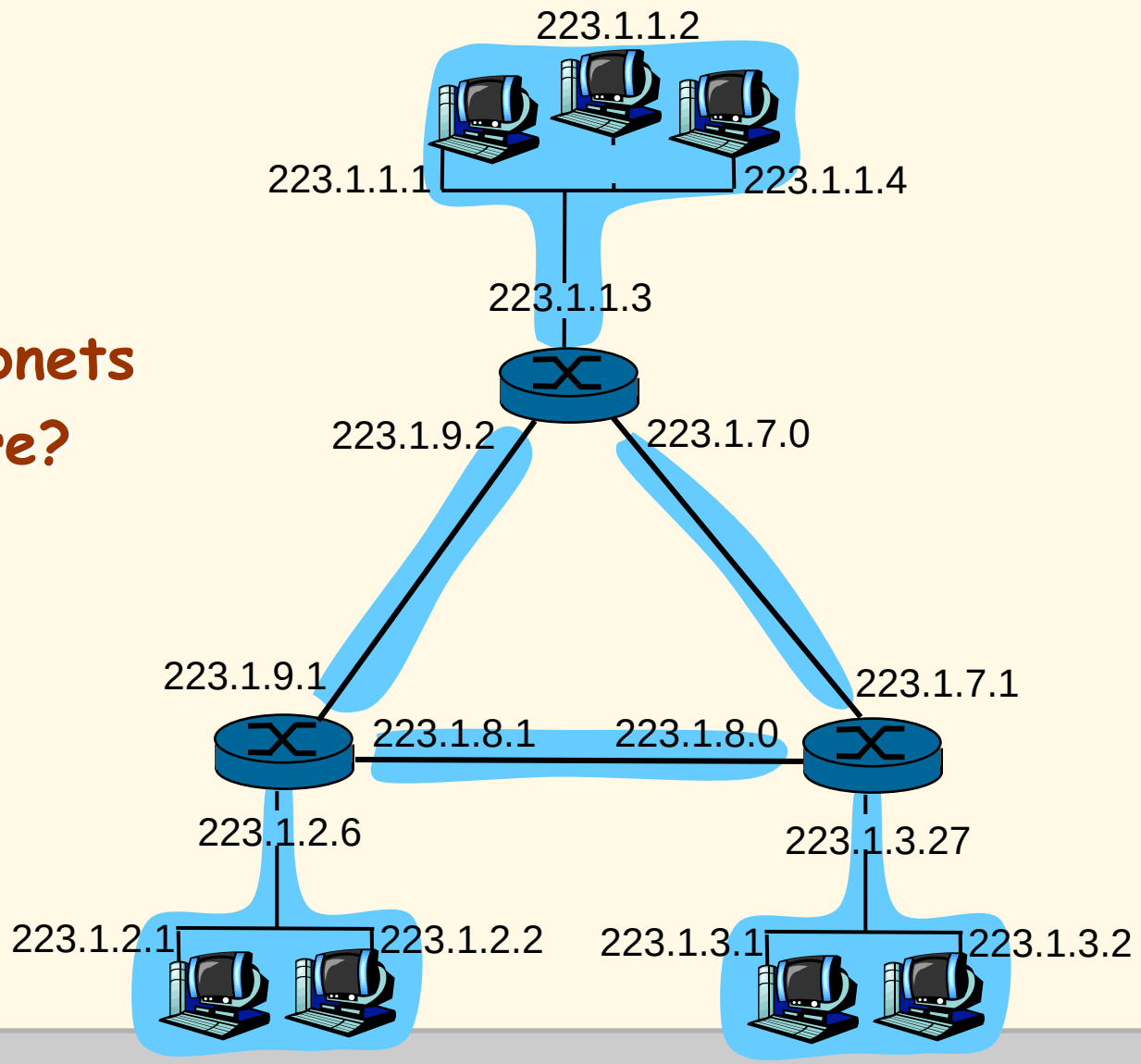


Subnet mask: /24 :: defined by the leftmost 24 bits.

K & R

Subnets

How many subnets in the figure?



K & R

IP Addressing: CIDR

CIDR: Classless InterDomain Routing

- Allows a subnet portion of address of arbitrary length.
- address format: a.b.c.d/x, where x is number of bits in subnet portion of address.

subnet part			host part
11001000	00010111	00010000	00000000

200.23.16.0/23

K & R

Classless Routing (CIDR)

- CIDR helps aggregate routes by breaking up rigid boundaries between classes.
- Handing out Class C addresses in contiguous blocks by address makes it possible for addresses to share a common prefix.

  ➔ allocate Class C networks as a power of 2.

- We need a protocol that understands these rules, e.g., BGP
- Network numbers are represented by (length,value) where length is the length of the prefix {similar to a mask}.

Classless Routing (CIDR)

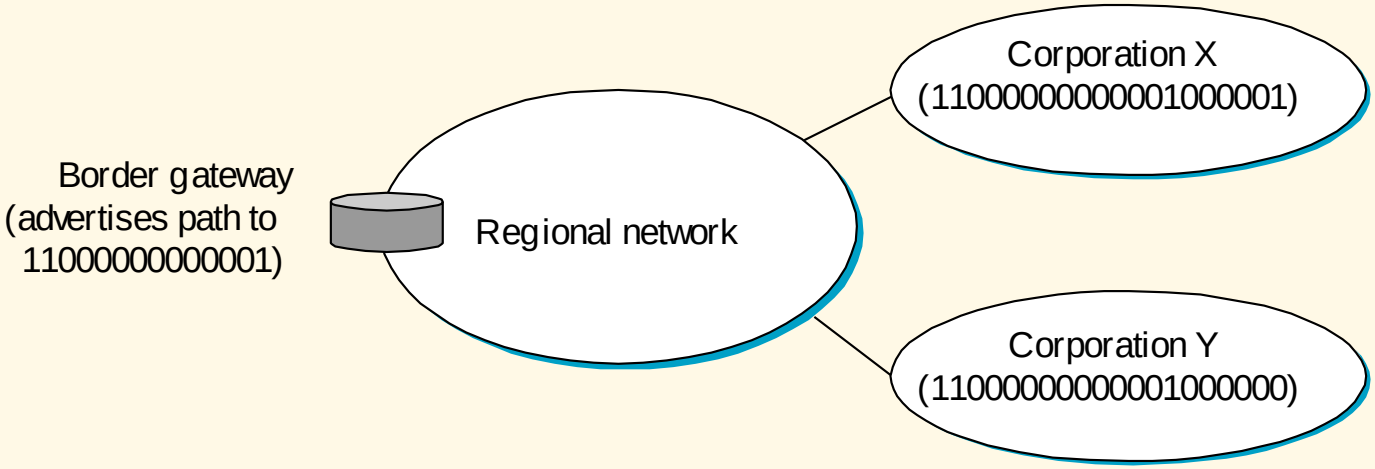
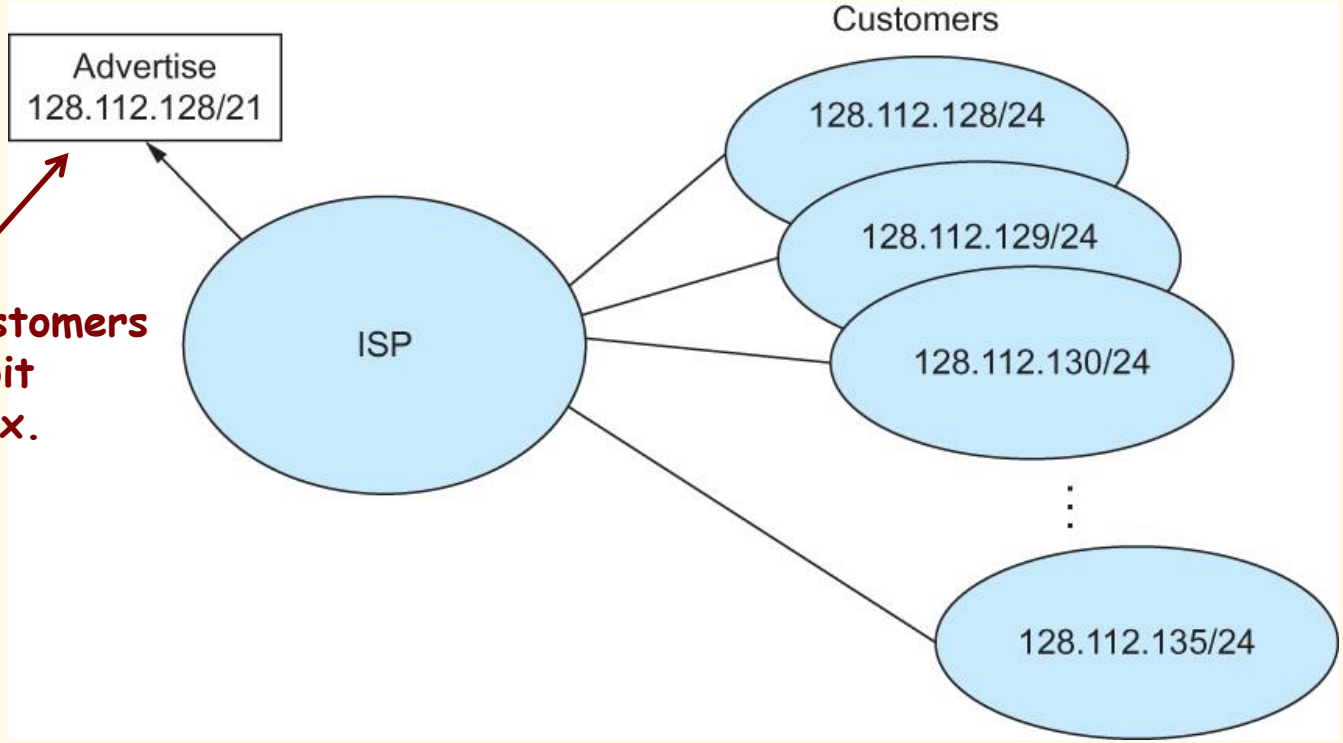


Figure 4.27 Route Aggregation with CIDR

P&D 4th

CIDR Route Aggregation



Eight ISP customers share a 21-bit common prefix.

Figure 3.22 Route Aggregation with CIDR

Network Layer Topics

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What is inside a router?
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 ICMP

K & R

Routing Algorithm Classification

Routing

Routing algorithm:: that part of the Network Layer responsible for deciding on which output line to transmit an incoming packet.

- Remember: For virtual circuit subnets the routing decision is made ONLY at set up.

Algorithm properties:: correctness, simplicity, robustness, stability, fairness, optimality, and scalability.

Routing Classification

Adaptive Routing

based on current measurements of traffic and/or topology.

1. centralized
2. isolated
3. distributed

Non-Adaptive Routing

routing computed in advance and off-line

1. flooding
2. static routing using shortest path algorithms

Flooding

- Pure flooding :: every incoming packet to a node is sent out on every outgoing line.
  - Obvious adjustment – do not send out on arriving link (assuming full-duplex links).
  - The routing algorithm can use a hop counter (e.g., TTL) to dampen the flooding.
  - Selective flooding :: only send on those lines going "approximately" in the right direction.

Routing is Graph Theory Problem

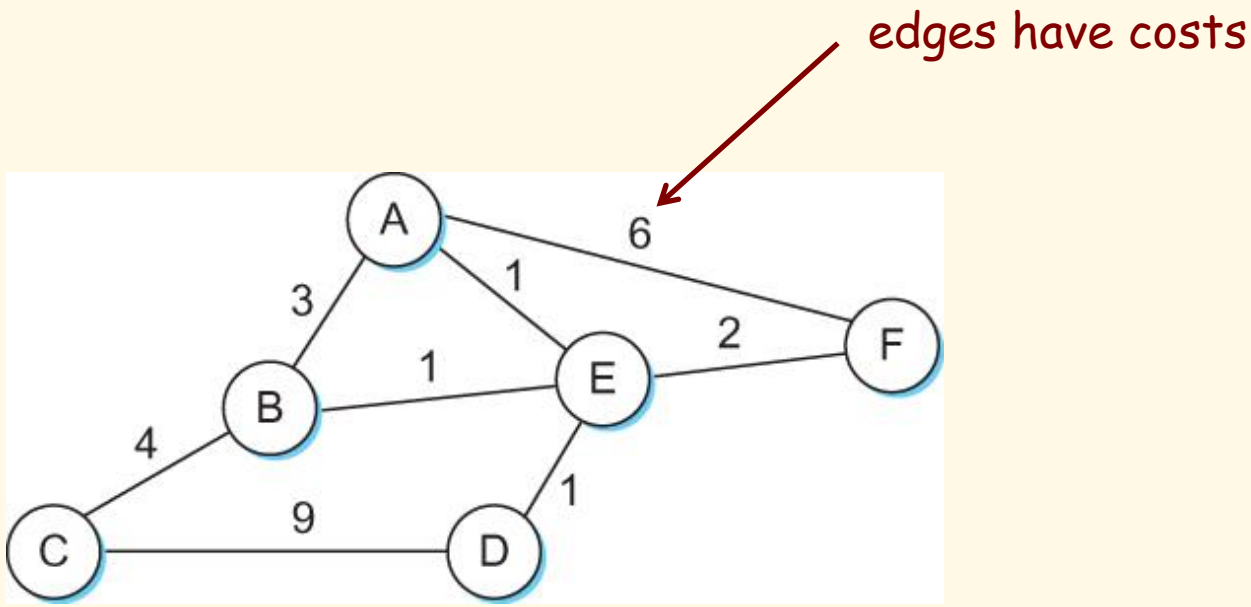


Figure 3.28 Network represented as a graph.

Shortest Path Routing

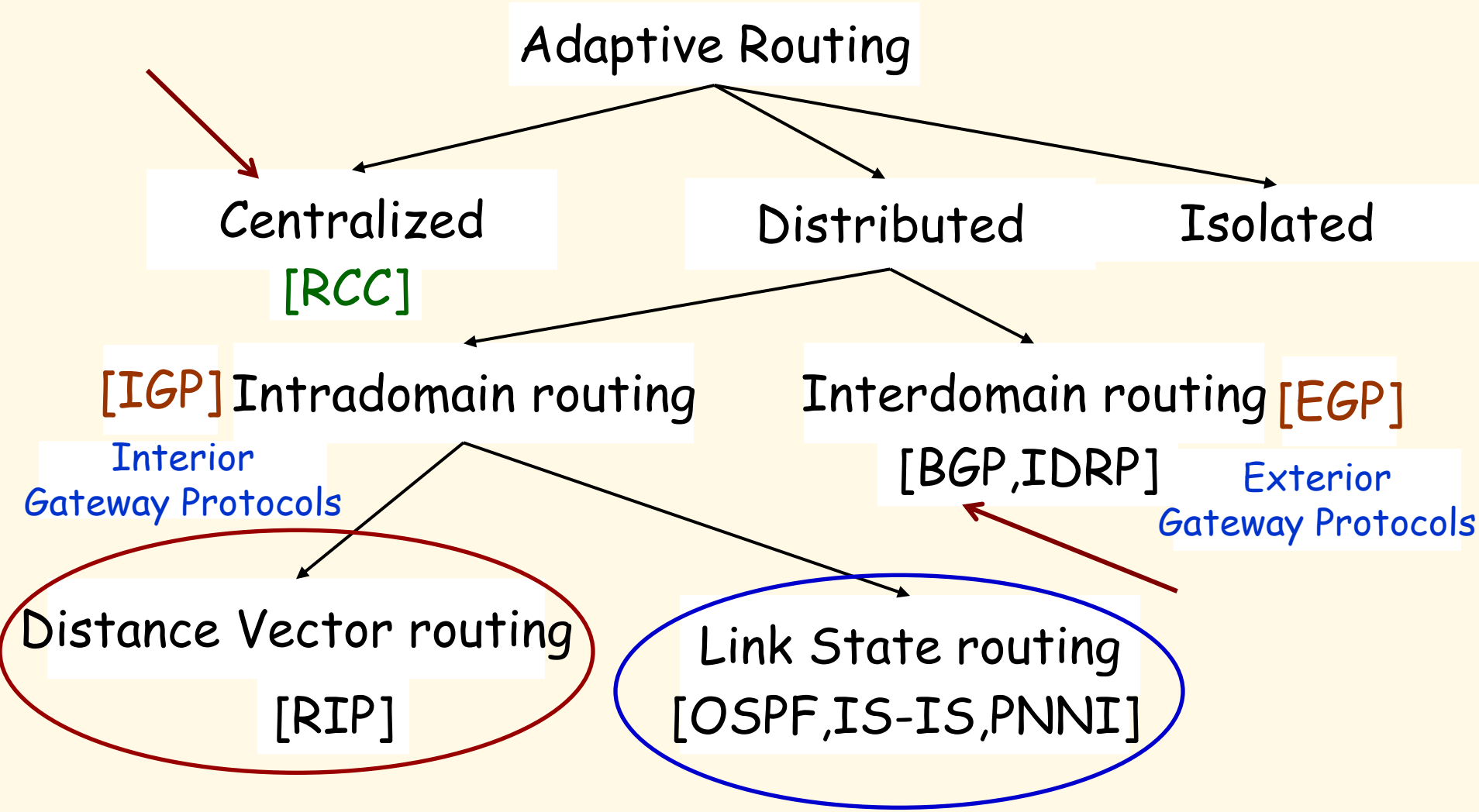
1. Bellman-Ford Algorithm [Distance Vector]
2. Dijkstra's Algorithm [Link State]

What does it mean to be the shortest (or optimal) route?

We need a cost metric (edges in graph):

a. Minimize the number of hops along the path.
b. Minimize the mean packet delay.
c. Maximize the network throughput.

Internetwork Routing [Halsall]



Adaptive Routing Design

Design Issues:

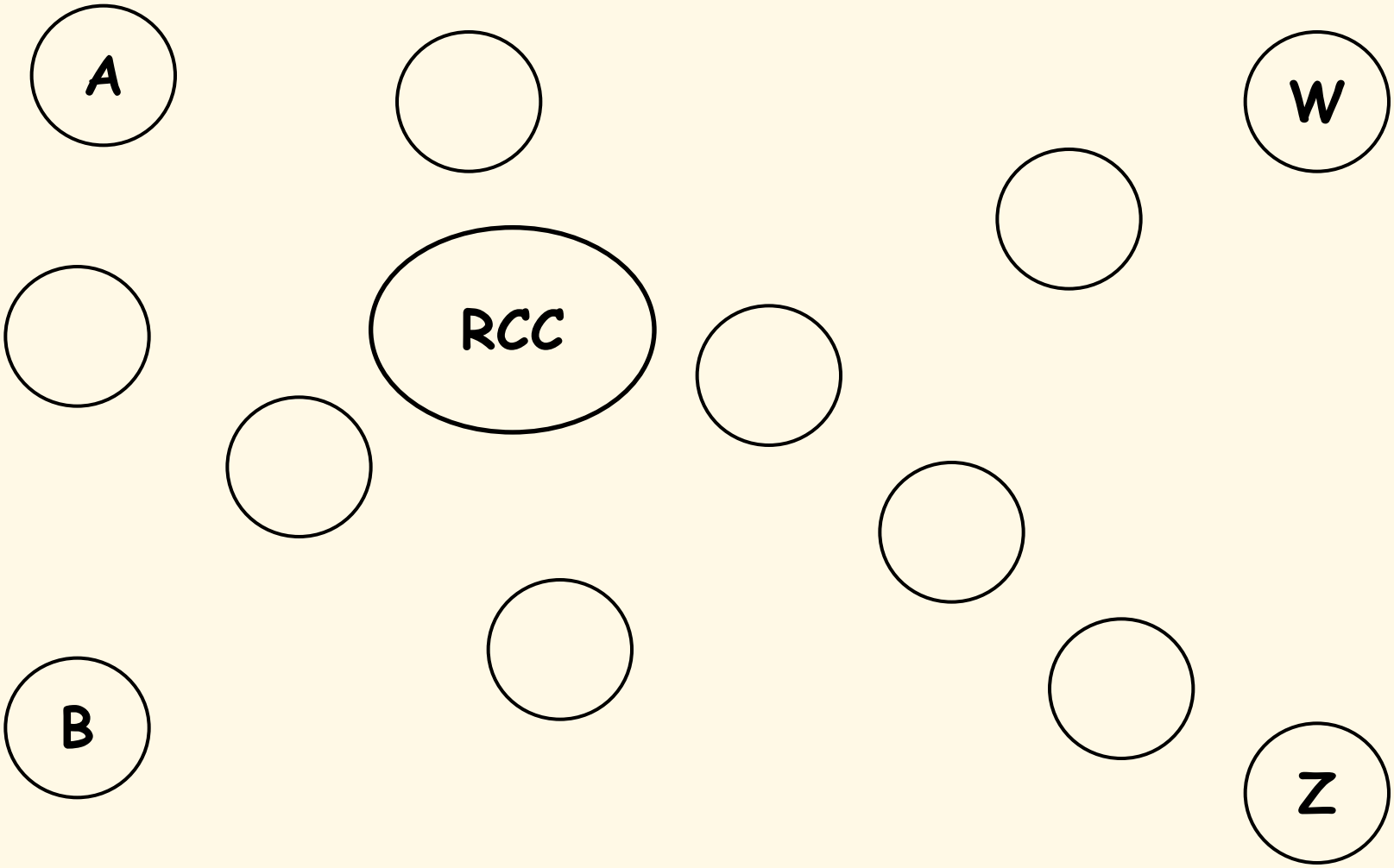
1. How much overhead is incurred due to gathering the routing information and sending *routing packets*?
2. What is the time frame (i.e, the frequency) for sending *routing packets* in support of adaptive routing?
3. What is the complexity of the routing strategy?

Adaptive Routing

Basic functions:

1. Measurement of pertinent network data {e.g. the cost metric}.
2. Forwarding of information to where the routing computation will be done.
3. Compute the routing tables.
4. Convert the routing table information into a routing decision and then dispatch the data packet.

Centralized Routing



Network Layer Topics

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What is inside a router?
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 ICMP

K & R

Distance Vector Routing

{Tanenbaum & Perlman version}

Computer Networks

Network Layer

Distance Vector Routing

Historically known as the old ARPANET routing algorithm {or known as Bellman-Ford (BF) algorithm}.

BF Basic idea: each router maintains a Distance Vector table containing the *distance* between itself and ALL possible destination nodes.

Distances, based on the chosen metric, are computed using information from the neighbors' distance vectors.

Distance Metric: usually hops or delay

Distance Vector Routing

Information kept by DV router

1. each router has an ID
2. associated with each link connected to a router, there is a link cost (static or dynamic).

Distance Vector Table Initialization

Distance to itself = 0

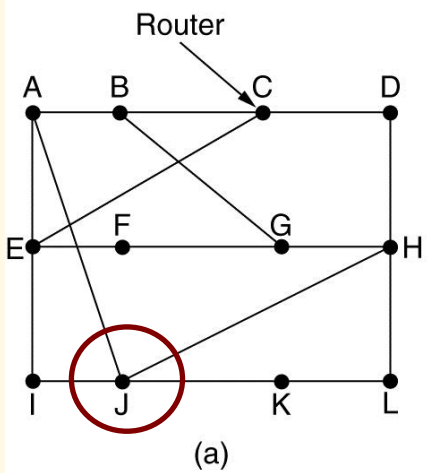
Distance to ALL other routers = infinity number

Distance Vector Algorithm [Perlman]

1. A router transmits its distance vector to <u>each of its neighbors</u> in a routing packet.
2. Each router receives and saves the most recently received distance vector from each of its neighbors.
3. A router recalculates its distance vector when:
   a. It receives a distance vector from a neighbor containing different information than before.
   b. It discovers that a link to a neighbor has gone down (i.e., a topology change).

- The DV calculation is based on minimizing the cost to each destination.

Distance Vector Example



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	New routing table for J	

Vectors received from J's four neighbors

(b)

Figure 5-9.(a) A subnet. (b) Input from A, I, H, K, and the new routing table for J.

Tanenbaum

Distance Vector Routing

{Kurose & Ross version}

Distance Vector Algorithm

Bellman-Ford Equation (dynamic programming)

Define

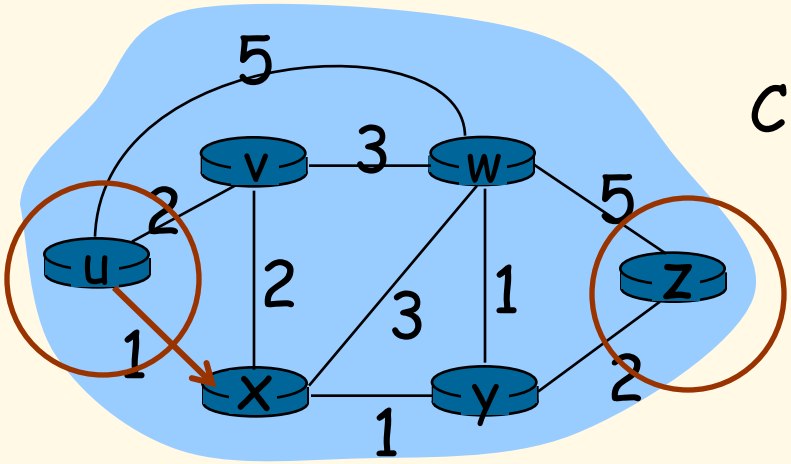
$d_x(y)$:= cost of least-cost path from x to y

Then

$$\boxed{d_x(y) = \min_v \{c(x,v) + d_v(y)\}}$$

where min is taken over all neighbors v of x.

Bellman-Ford Example



Clearly, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

B-F equation says:

$$
\begin{aligned}
d_u(z) &= \min \{ c(u,v) + d_v(z), \\
&\qquad c(u,x) + d_x(z), \\
&\qquad c(u,w) + d_w(z) \} \\
&= \min \{2 + 5, \\
&\qquad 1 + 3, \\
&\qquad 5 + 3\} = 4
\end{aligned}
$$

The node that achieves minimum is next hop in shortest path ➜ forwarding table.

Namely, packets from u destined for z are forwarded out link between u and x.

Distance Vector Algorithm (3)

- $D_x(y)$ = estimate of least cost from x to y
- Node x knows cost to each neighbor v:

  $c(x,v)$
- Node x maintains distance vector

  $D_x = [D_x(y): y \in N]$
- Node x also maintains its neighbors' distance vectors
  - For each neighbor v, x maintains

    $D_v = [D_v(y): y \in N]$

Distance Vector Algorithm (4)

DV Basic idea:

- From time-to-time, each node sends its own distance vector estimate to neighbors.
- Asynchronous
- When a node x receives a new DV estimate from any neighbor v, it saves v's distance vector and it updates its own DV using B-F equation:

$$D_x(y) \leftarrow \min_v\{c(x,v) + D_v(y)\} \quad \text{for each node } y \in N$$

- Under minor, natural conditions, the estimate $D_x(y)$ converges to the actual least cost $d_x(y)$.

Distance Vector Algorithm (5)

Iterative, asynchronous: each local iteration caused by:

- local link cost change
- DV update message from neighbor

Distributed:

- each node notifies neighbors *only* when its DV changes
  - neighbors then notify their neighbors if necessary.

Each node:

wait for (change in local link cost or msg from neighbor)

recompute estimates

if DV to any destination has changed, *notify* neighbors

$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\}$
$= \min\{2+0\ ,\ 7+1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\}$
$= \min\{2+1\ ,\ 7+0\} = 3$

node x table

from \ cost to	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

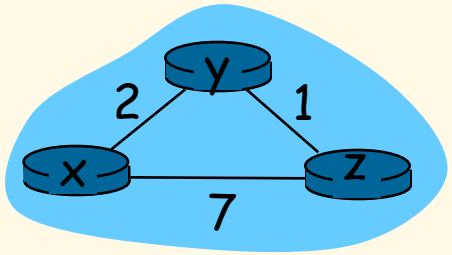
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

node y table

from \ cost to	x	y	z
x		∞	∞
y	2	0	1
z	∞	∞	∞

node z table

from \ cost to	x	y	z
x	∞	∞	∞
y		∞	∞
z	7	1	0



time

$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$
$= \min\{2+0 , 7+1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$
$= \min\{2+1 , 7+0\} = 3$

node x table

from \ cost to	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node y table

from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

from \ cost to	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

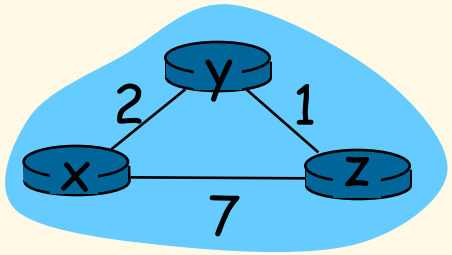
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node z table

from \ cost to	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

from \ cost to	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

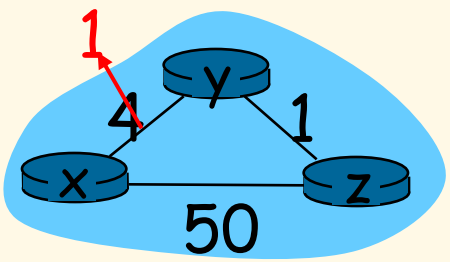


time

Distance Vector: Link Cost Changes

Link cost changes:

- node detects local link cost change.
- updates routing info, recalculates distance vector.
- if DV changes, it notifies neighbors



"good news travels fast"

- At time t_0, y detects the link-cost change, updates its DV, and informs its neighbors.

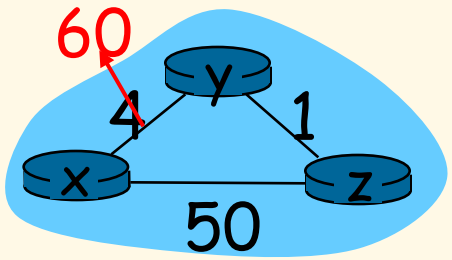
  At time t_1, z receives the update from y and updates its table. It computes a new least cost to x and sends its neighbors its DV.

  At time t_2, y receives z's update and updates its distance table. y's least costs do not change and hence y does *not* send any message to z.

Distance Vector: Link Cost Changes

Link cost changes:

- good news travels fast
- bad news travels slow - "count to infinity" problem!
- 44 iterations before algorithm stabilizes: see P&D page 248!



Possible solutions:

1. Keep 'infinity ' small {depends on graph diameter}.
2. Split Horizon: node does not send those routes learned from a neighbor back to that neighbor.
3. Split Horizon with Poison Reverse:
   - If z routes through y to get to x, z tells y its (z's) distance to x is infinite (so y won't route to x via z).

- Does this solve count to infinity problem?

Network Layer Topics

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What is inside a router?
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 ICMP

K & R

Link State Algorithm

1. Each router is responsible for meeting its neighbors and learning their names.
2. Each router constructs a link state packet (LSP) which consists of a list of names and cost to reach each of its neighbors.
3. The LSP is transmitted to ALL *other routers.* Each router stores the most recently generated LSP from each other router.
4. Each router uses complete information on the network topology to compute the shortest path route to each destination node.

Reliable Flooding

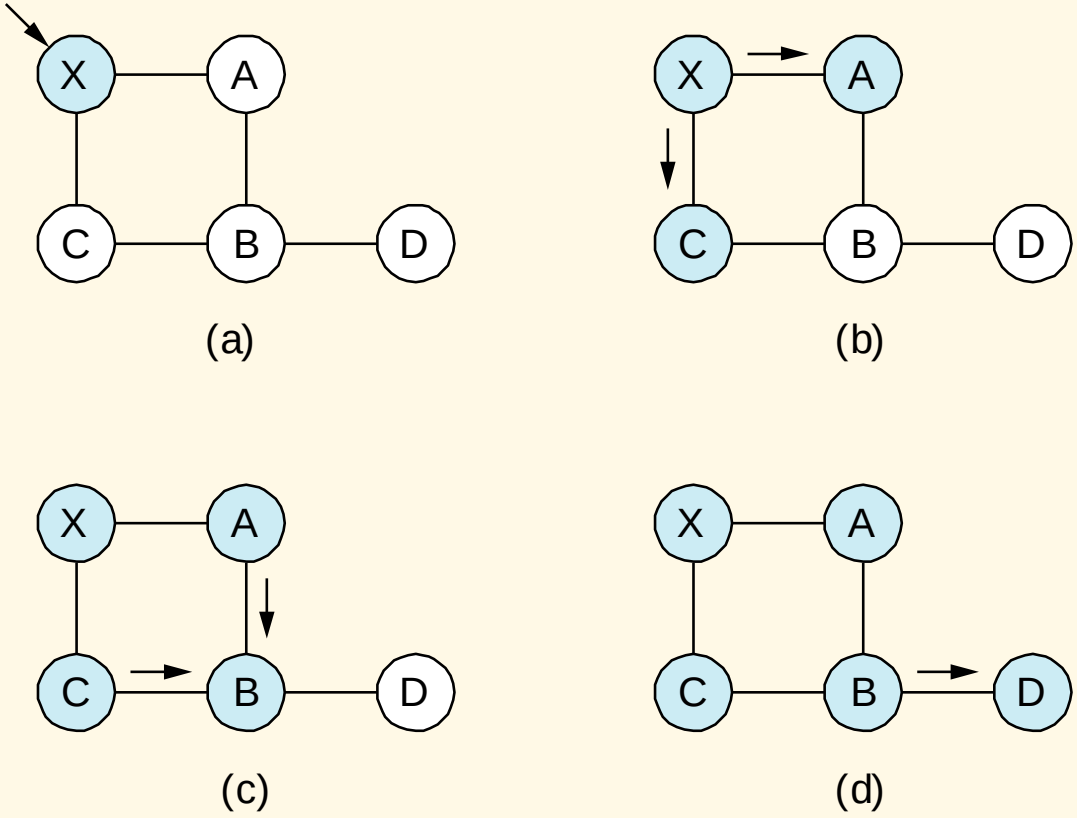


Figure 3.32 Reliable LSP Flooding

Reliable Flooding

- The process of making sure all the nodes participating in the routing protocol get a copy of the link-state information from all the other nodes.
- **LSP** contains:
  - Sending router's node ID
  - List of connected neighbors with the associated link cost to each neighbor
  - Sequence number
  - Time-to-live (TTL) {*an aging mechanism*}

Reliable Flooding

- First two items enable route calculation.
- Last two items make process reliable
  - ACKs and checking for duplicates is needed.
- Periodic **Hello** packets used to determine the demise of a neighbor.
- The sequence numbers are not expected to wrap around.

  ➔ this field needs to be large (64 bits) !!

A Link-State Routing Algorithm

Dijkstra's algorithm

- net topology, link costs known to all nodes
  - accomplished via "link state broadcast".
  - all nodes have same info.
- computes least cost paths from one node ('source") to all other nodes
  - gives forwarding table for that node.
- iterative: after k iterations, know least cost path to k destinations.

Notation:

- **c(x,y):** link cost from node x to y; = ∞ if not direct neighbors.
- **D(v):** current value of cost of path from source to destination v
- **p(v):** predecessor node along path from source to v
- **N':** set of nodes whose least cost path is definitively known.

K & R

Dijsktra's Algorithm [K&R]

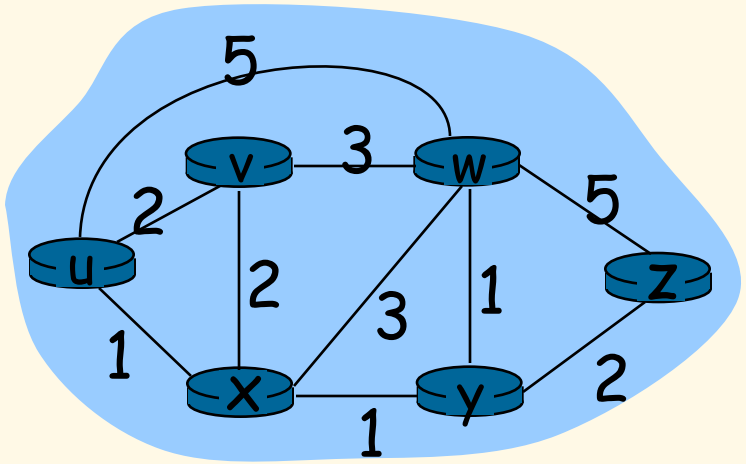
```
Initialization:
  N' = {u}
  for all nodes v
    if v adjacent to u
        then D(v) = c(u,v)
    else D(v) = ∞

Loop
   find w not in N' such that D(w) is a minimum
    add w to N'
    update D(v) for all v adjacent to w and not in N' :
        D(v) = min( D(v), D(w) + c(w,v) )
    /* new cost to v is either old cost to v or known
     shortest path cost to w plus cost from w to v */
until all nodes in N'
```

K & R

Dijkstra's Algorithm: Example

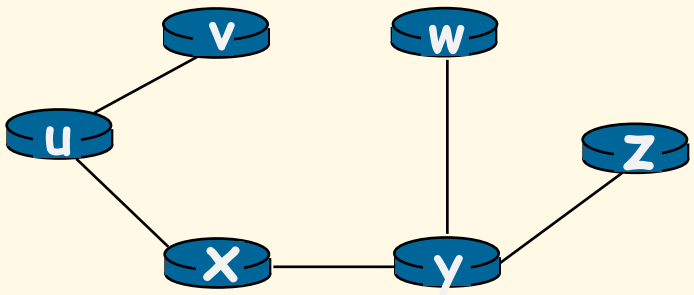
Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



K & R

Dijkstra's Algorithm: Example (2)

Resulting shortest-path tree from u:



Resulting forwarding table in u:

destination	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

K & R

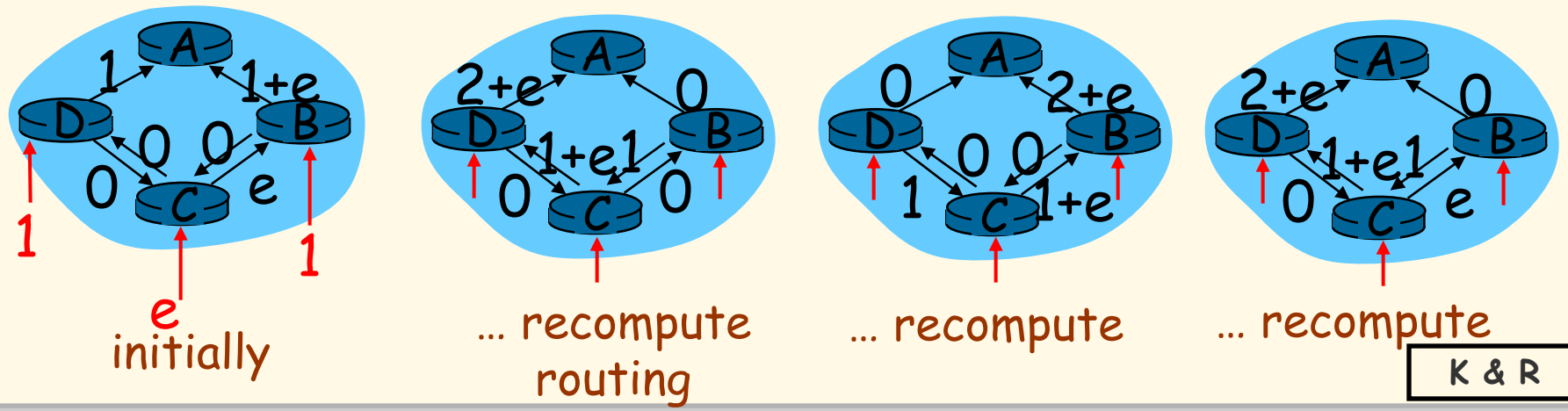
Dijkstra's Algorithm, Discussion

Algorithm complexity: n nodes

- each iteration: need to check all nodes, w, not in N
- $n(n+1)/2$ comparisons: $O(n^2)$
- more efficient implementations possible: $O(n\log n)$

Oscillations possible:

- e.g., link cost = amount of carried traffic



K & R

Network Layer Topics

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 ICMP

K & R

Hierarchical Routing

- Our routing study thus far - idealization
- all routers identical
- network "flat"
- … *not* true in practice

scale: with 200 million destinations:

- can't store all destinations in routing tables!
- routing table exchange would swamp links!

administrative autonomy

- internet = network of networks
- each network admin may want to control routing in its own network.

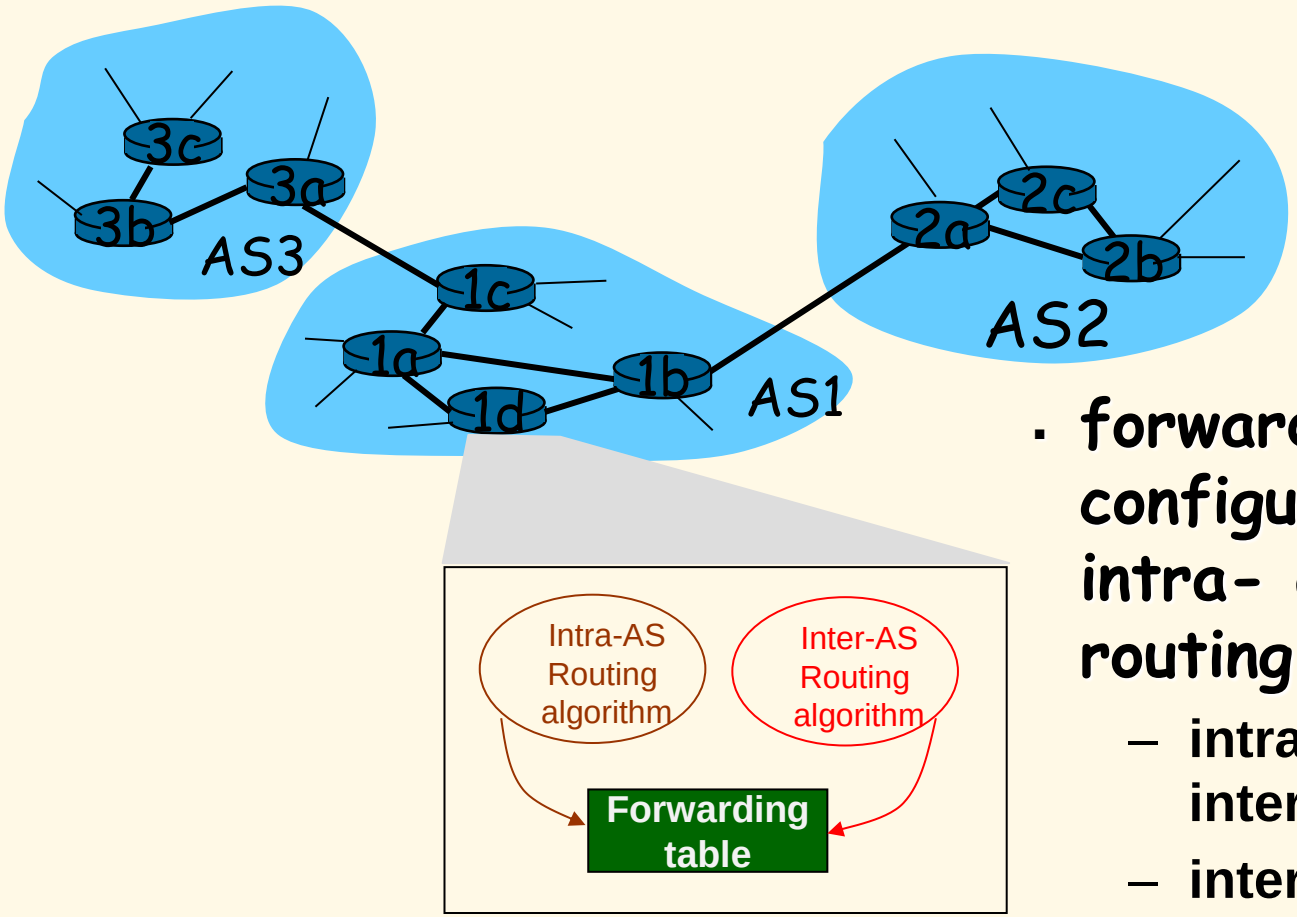
Hierarchical Routing

- aggregate routers into regions, "autonomous systems" (AS)
- routers in same AS run same routing protocol
  - "intra-AS" routing protocol
  - routers in different AS can run different intra-AS routing protocol

<u>Gateway router</u>

- Direct link to router in another AS

Interconnected AS's



- forwarding table configured by both intra- and inter-AS routing algorithm
  - intra-AS sets entries for internal destinations.
  - inter-AS & intra-AS sets entries for external destinations.

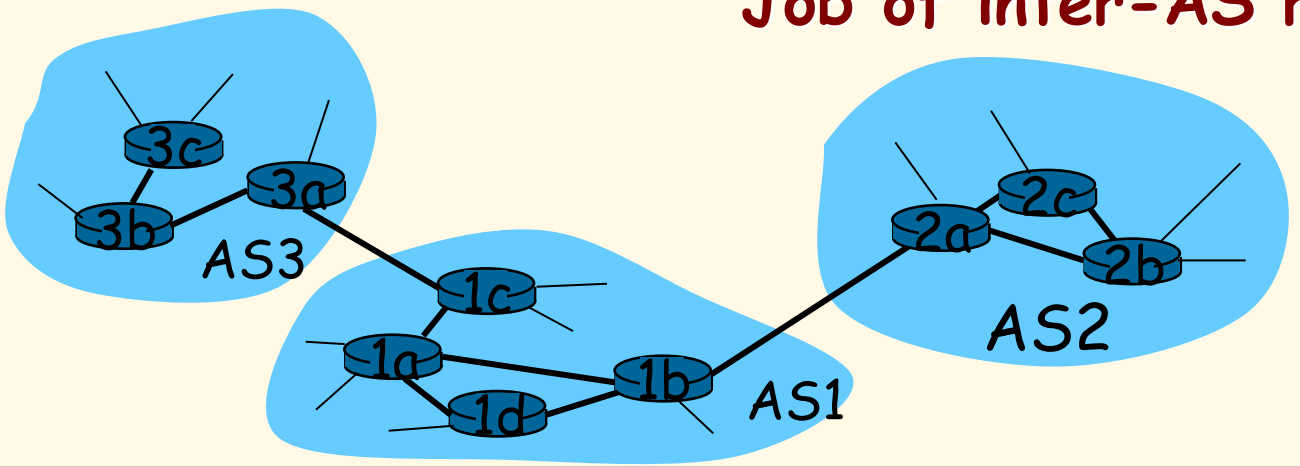
Inter-AS Tasks

- suppose router in AS1 receives datagram destined outside of AS1:
  - router should forward packet to gateway router, but which one?

AS1 must:

1. learn which dests are reachable through AS2, which through AS3
2. propagate this reachability info to all routers in AS1

Job of inter-AS routing!



Network Layer Topics

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What is inside a router?
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 ICMP

Intra-AS Routing

- also known as Interior Gateway Protocols (IGP)
- most common Intra-AS routing protocols:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary)

Network Layer Topics

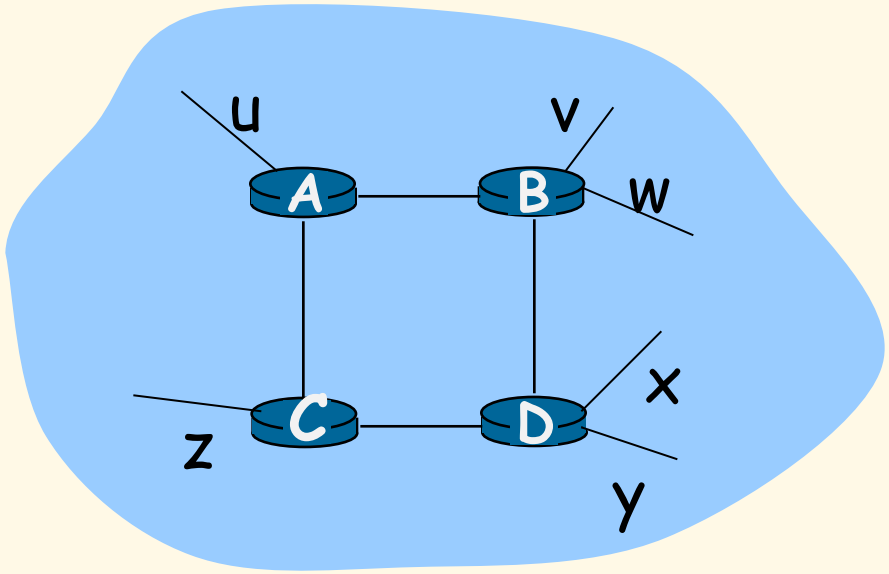
- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What is inside a router?
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 ICMP

Routing Information Protocol (RIP)

- RIP had widespread use because it was distributed with BSD Unix in "routed", a router management daemon in 1982.
- RIP - most used Distance Vector protocol.
- RFC1058 in June 1988
- Runs over UDP.
- Metric = hop count
- Big problem is max. hop count =16

  ➔ RIP limited to running on small networks (or AS's that have a small diameter)!!

Routing Information Protocol (RIP)



From router A to subnets:

destination	hops
u	1
v	2
w	2
x	3
y	3
z	2

- Sends DV packets every 30 seconds (or faster) as Response Messages (also called advertisements).
- each advertisement: list of up to 25 destination subnets within AS.
- Upgraded to RIPv2

RIP Packets

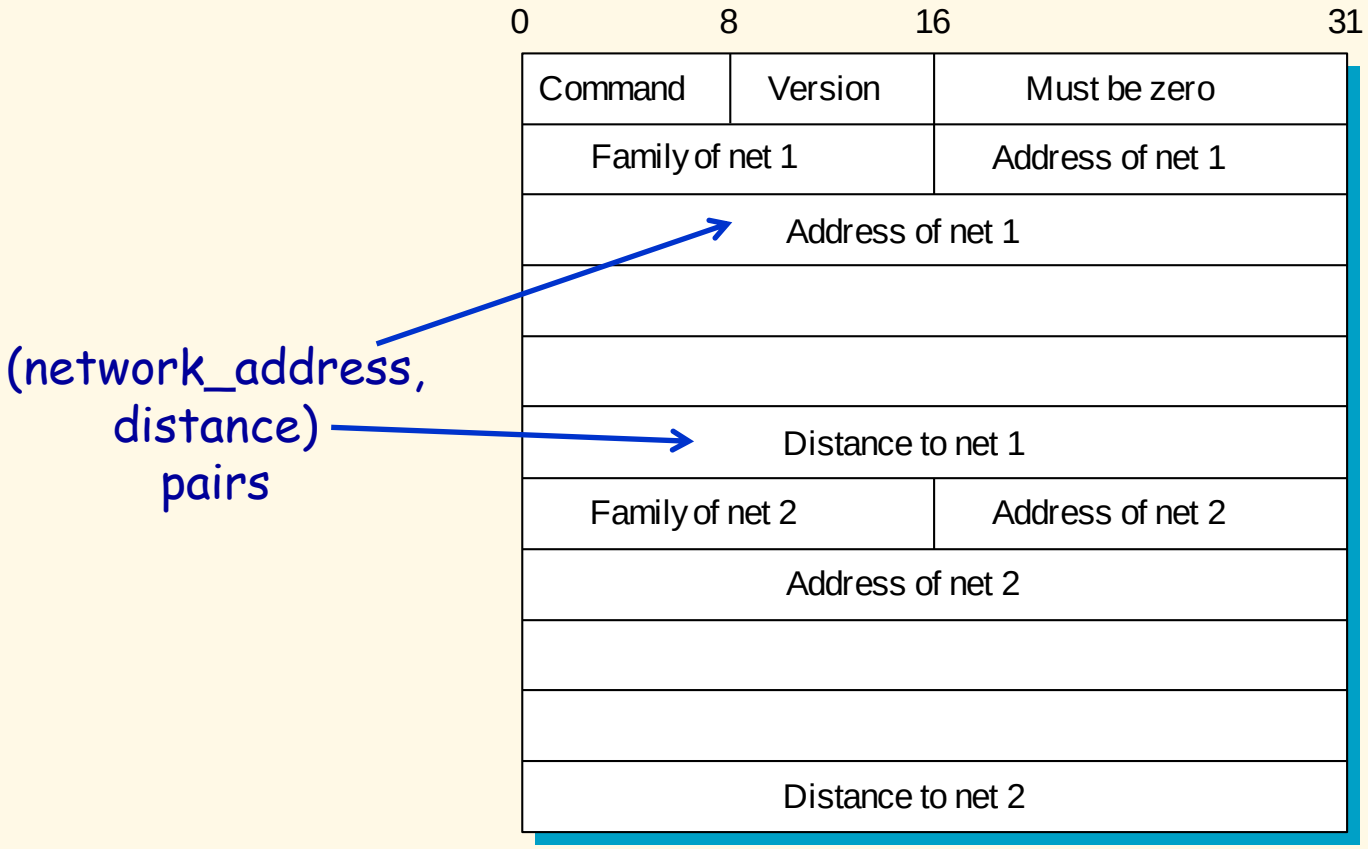


Figure 4.17 RIP Packet Format

P&D 4th

RIPv2

- Allows routing on a subnet (subnet masks)
- Has an authentication mechanism
- Tries to deal with multicast
- Uses route tags
- Has the ability for router to announce routes on behalf of another router.

RIPv2 Packets

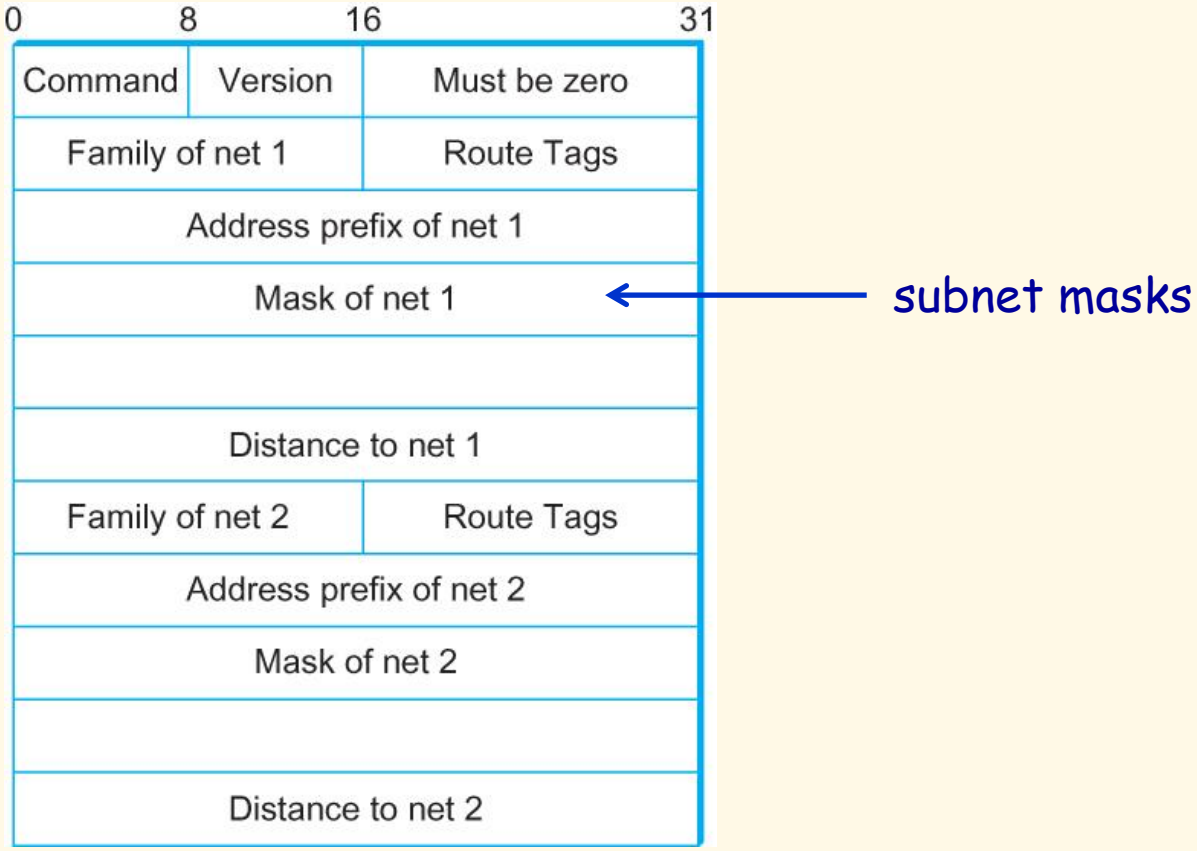


Figure 3.31 RIPv2 Packet Format

Network Layer Topics

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 ICMP

K & R

OSPF (Open Shortest Path First)

- "open" :: publicly available (due to IETF)
- uses Link State algorithm
  - LS packet dissemination
  - topology map at each node
  - route computation uses Dijkstra's algorithm.
- OSPF advertisement carries one entry per neighbor router.
- advertisements disseminated to entire AS (via flooding*).
  - carried in OSPF messages directly over IP (rather than TCP or UDP).

* However hierarchy (partitioning domains into areas) reduces flooding impact.

Partitioning Domains

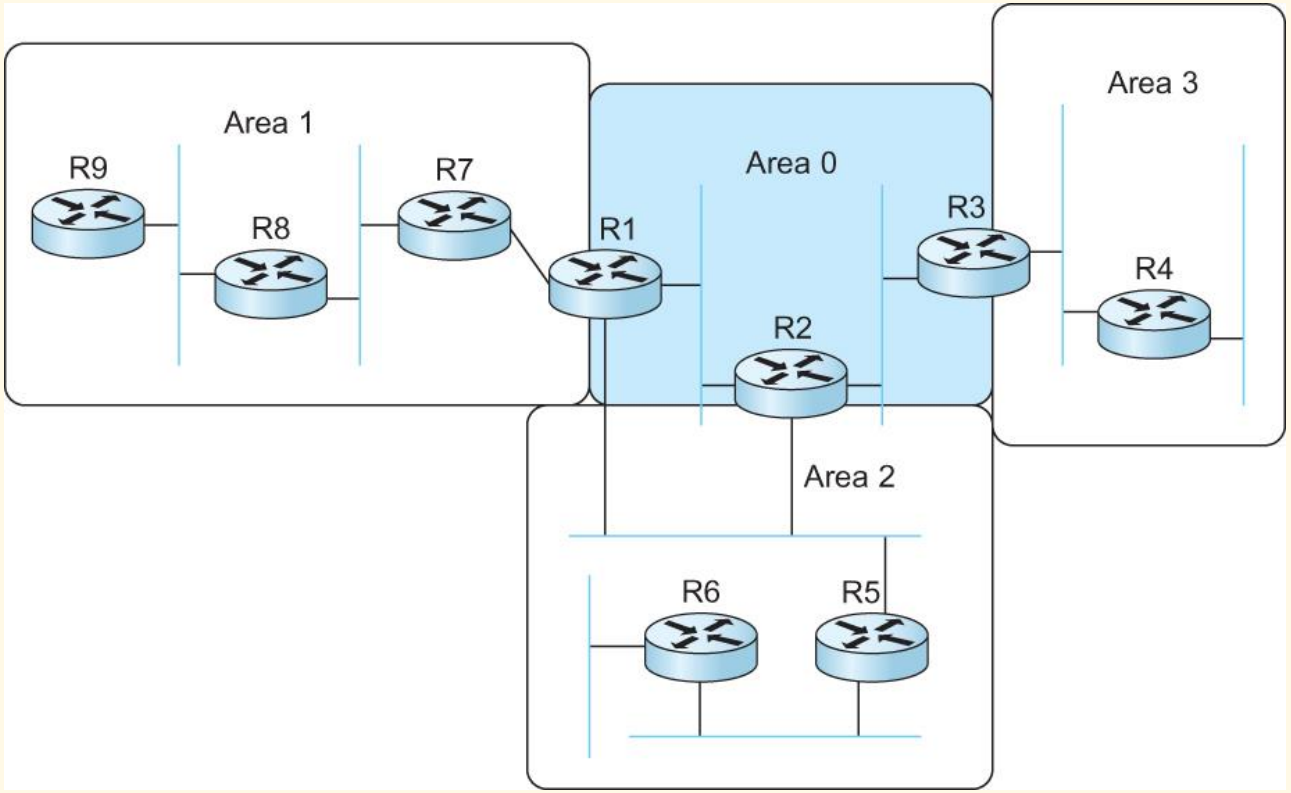
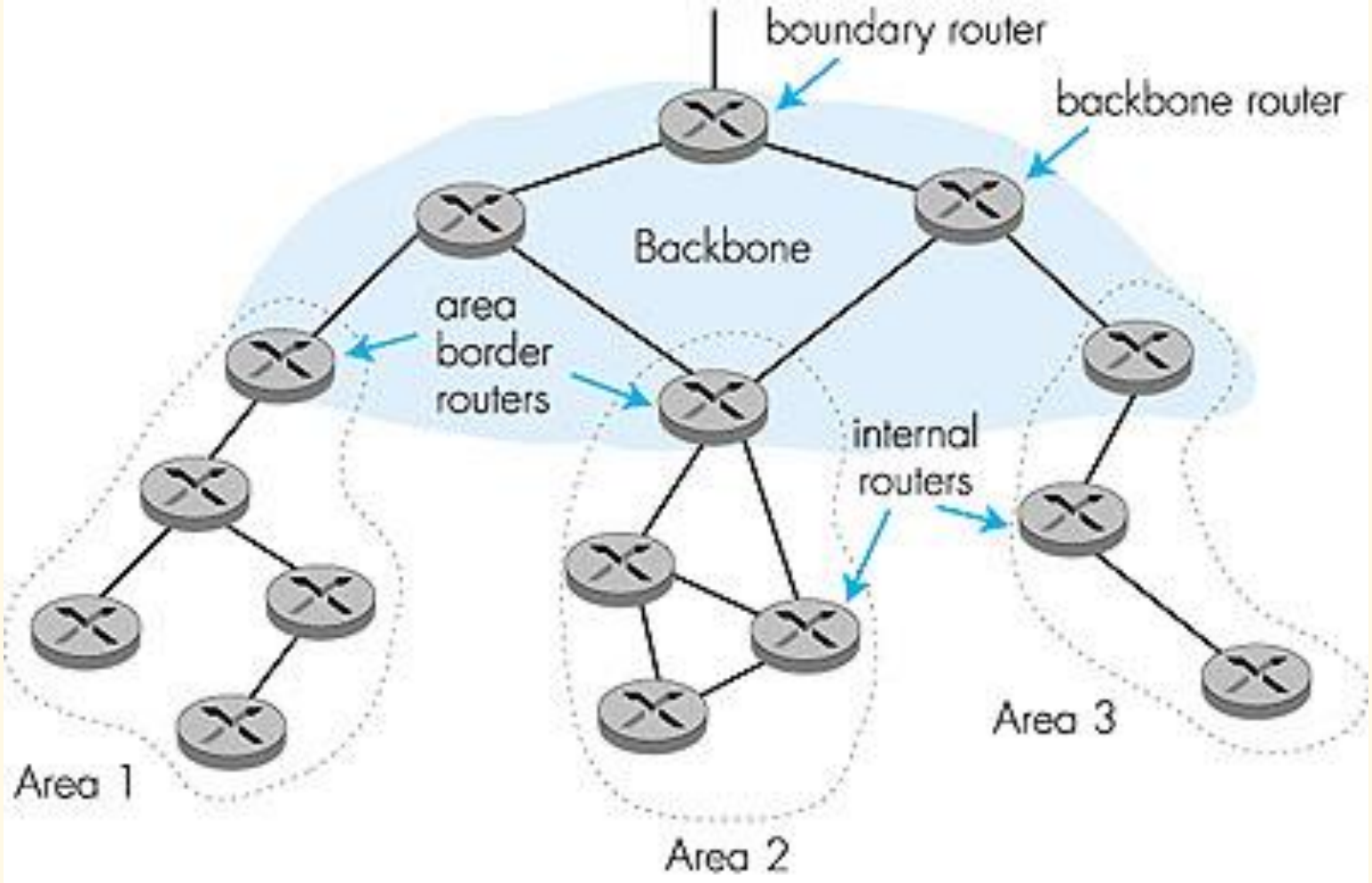


Figure 4.2 A domain divided into areas

Hierarchical OSPF



Hierarchical OSPF

- **Two-level Hierarchy::** local area, backbone.
  - Some Link-State Advertisement (LSA) types are only sent into one area.
  - Each node has detailed area topology; only knows direction (shortest path) to nets in other areas.
- *area border routers:* "summarize" distances to nets in own area, advertise to other Area Border routers.
- *backbone routers:* run OSPF routing limited to backbone.
- *boundary routers:* connect to other AS's.

OSPF LSA Types

1. Router link advertisement [Hello message]
2. Network link advertisement
   - identifies connected networks.
3. Network summary link advertisement
4. AS border router's summary link advertisement
5. AS external link advertisement

OSPF Header

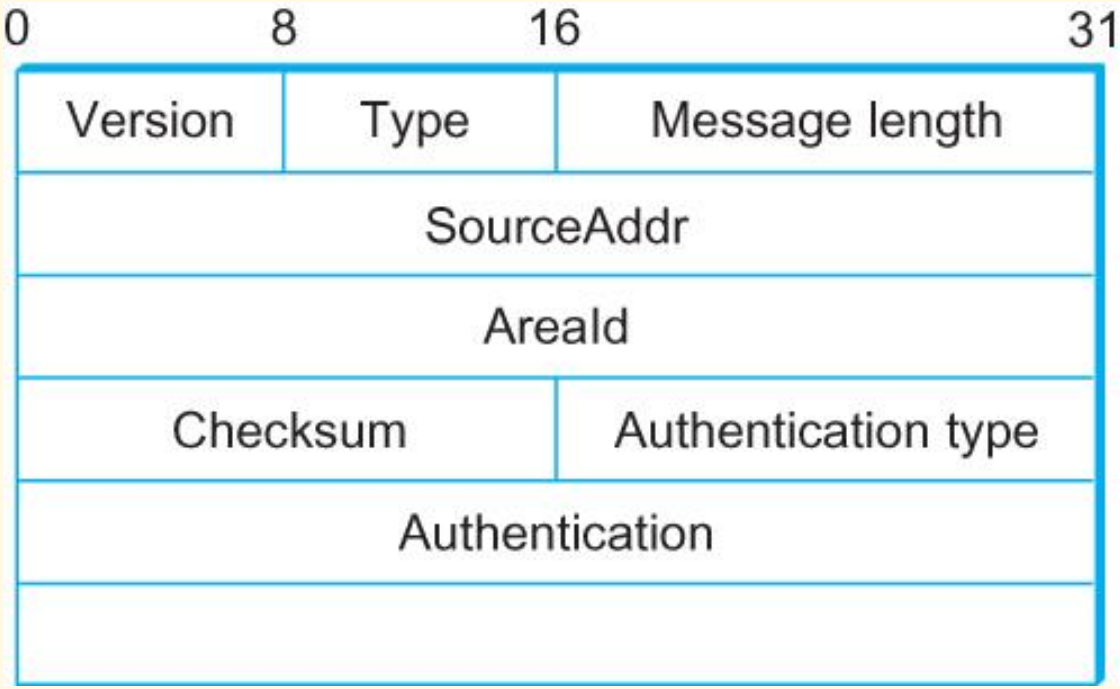


Figure 3.34 OSPF Header Format

Type 1 OSPF LSA

LS Age			Options	Type=1
Link-state ID				
Advertising router				
LS sequence number				
LS checksum			Length	
0	Flags	0	Number of links	
Link ID				
Link data				
Link type		Num_TOS	Metric	
Optional TOS information				
More links				

Figure 3.3 OSPF Link-State Advertisement (LSA)

OSPF "Advanced" Features (not in RIP)

- security: all OSPF messages authenticated (to prevent malicious intrusion).
- multiple same-cost paths means distributed load balancing of traffic over routes (only one path in RIP).
- For each link, multiple cost metrics for different TOS (e.g., satellite link cost set "low" for best effort; high for real time).
- integrated uni- and multicast support:
  - Multicast OSPF (MOSPF) uses same topology data base as OSPF.
- hierarchical OSPF is used in large domains.

OSPF

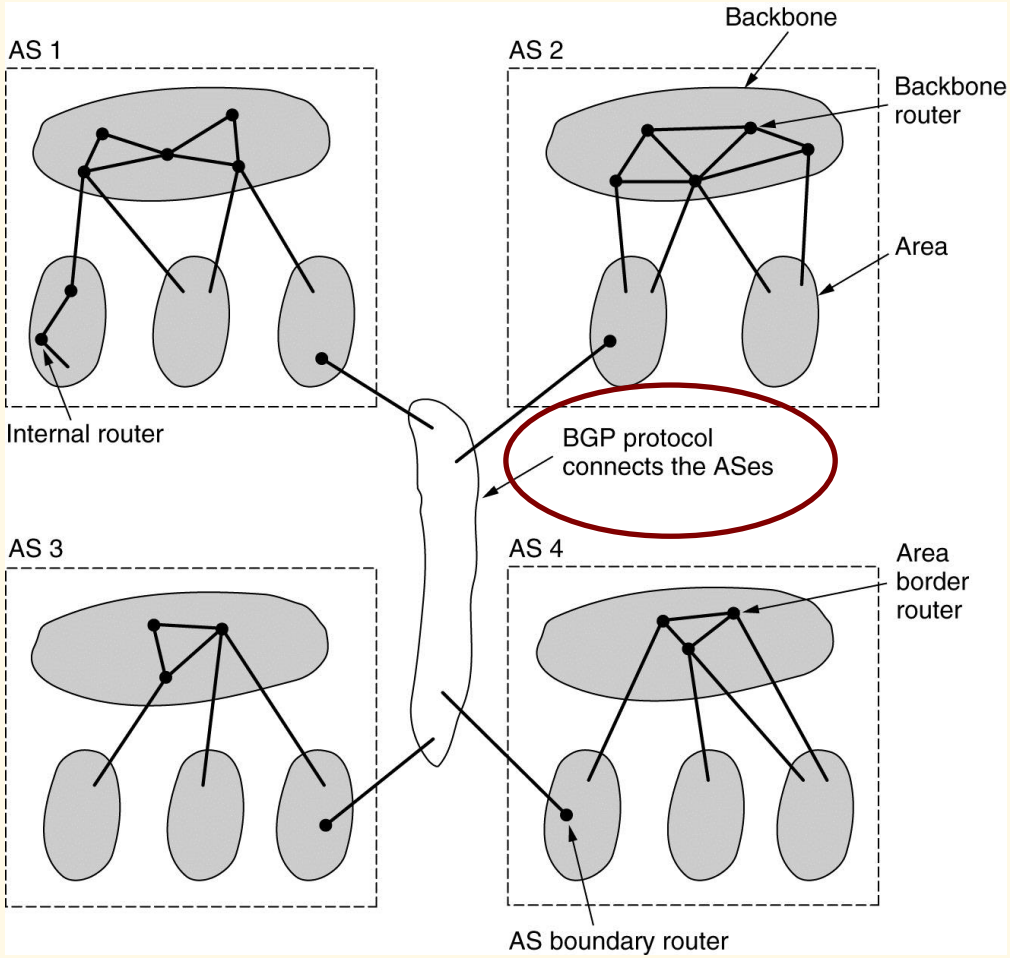


Figure 5-65.The relation between AS's, backbones, and areas in OSPF

Tanenbaum

Network Layer Topics

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What is inside a router?
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP &DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 ICMP

K & R

Internet Inter-AS Routing: BGP

- BGP (Border Gateway Protocol): de facto standard interdomain routing protocol.
- BGP provides each AS a means to:
  1. Obtain subnet reachability information from neighboring ASs.
  2. Propagate reachability information to all AS-internal routers.
  3. Determine "good" loop-free routes to subnets based on reachability information and policy.
- allows subnet to advertise its existence to rest of Internet: "I am here!"

BGP Assumes an Arbitrary Interconnection of AS's

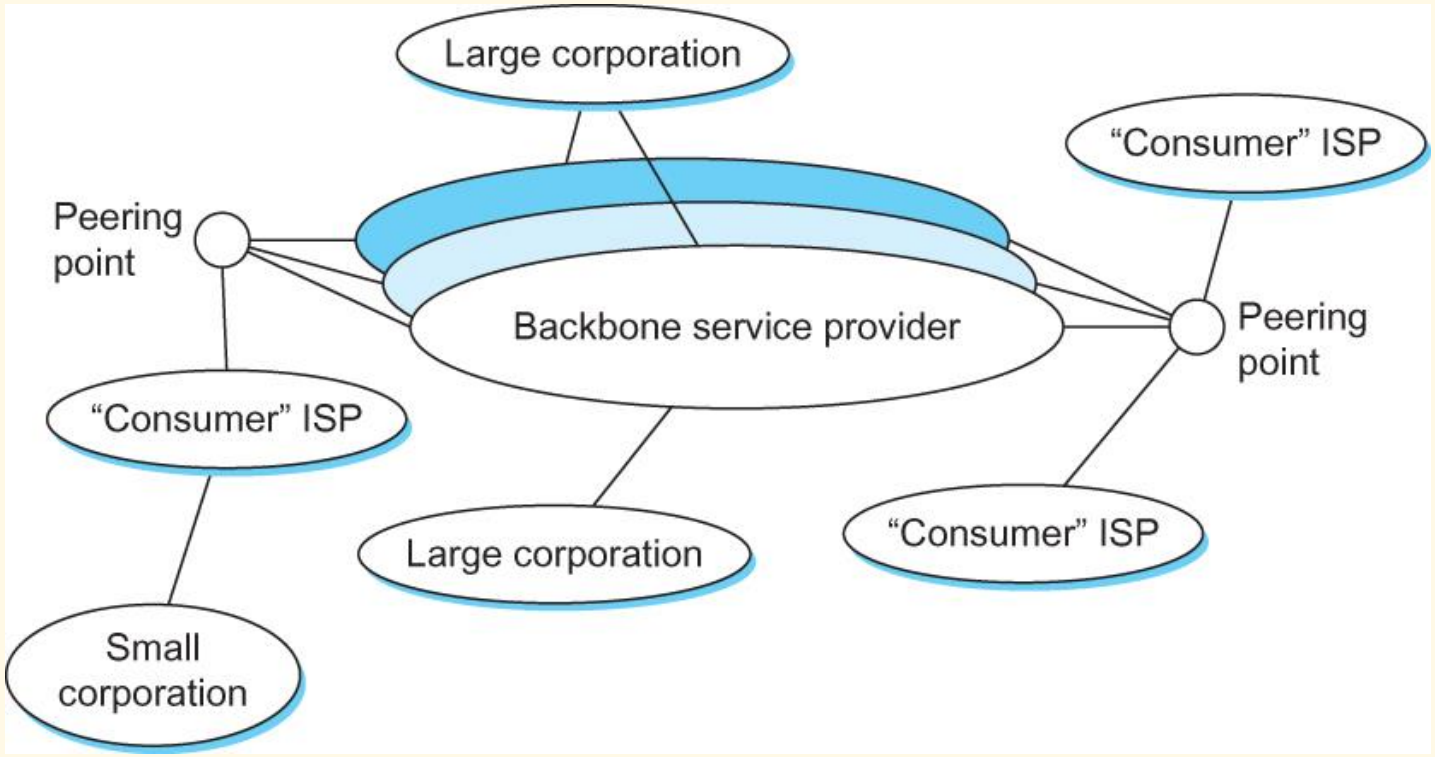


Figure 4.4 A simple multi-provider Internet

BGP Issues/Concerns

- **Scalability:** An Internet backbone router must be able to forward any packet destined anywhere in the Internet.
  - Having a routing table that will provide a match for any valid IP address.
- **Autonomous nature of the domains**
  - It is impossible to calculate meaningful path costs for a path that crosses multiple ASs.
  - A cost of 1000 across one provider might imply a great path but it might mean an unacceptable bad one from another provider.
- **Issues of trust**
  - Provider A might be unwilling to believe certain advertisements from provider B.

BGP-4: Border Gateway Protocol

- Define local traffic as traffic that originates at or terminates on nodes within an AS, and transit traffic as traffic that passes through an AS.

- Classify AS's into three types:
  - Stub AS:: an AS that has only a single connection to one other AS; such an AS will only carry local traffic (*small corporation in Figure 4.4*).

  - Multihomed AS:: an AS that has connections to more than one other AS, but refuses to carry transit traffic (*large corporation at the top in Figure 4.4*).

  - Transit AS:: an AS that has connections to more than one other AS, and is designed to carry both transit and local traffic (*backbone providers in Figure 4.4*).

BGP

- BGP does not belong to either of the two main classes of routing protocols (distance vectors and link-state protocols).
- BGP advertises *complete paths* as an enumerated lists of ASs to reach a particular network. Hence, often referred to as a *path-vector protocol.*
  - BGP paths need to be loop-free!!
  - ➔ Carried AS numbers need to be unique.

BGP Interdomain Routing

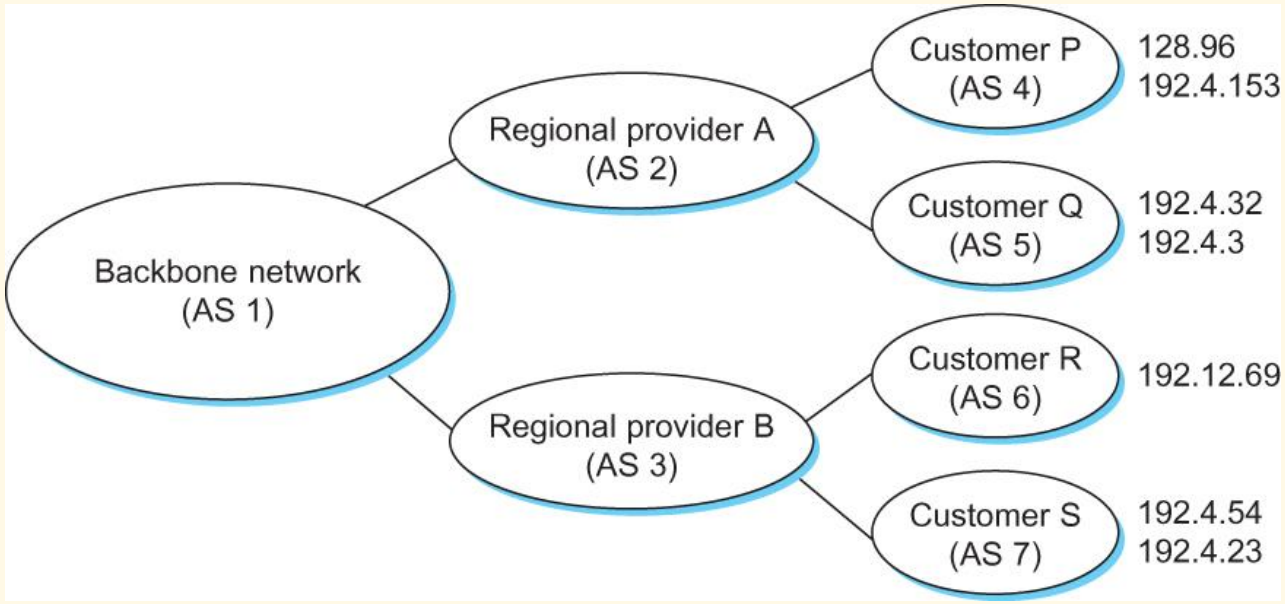


Figure 4.5 Example of a network running BGP

Loops in AS Topology

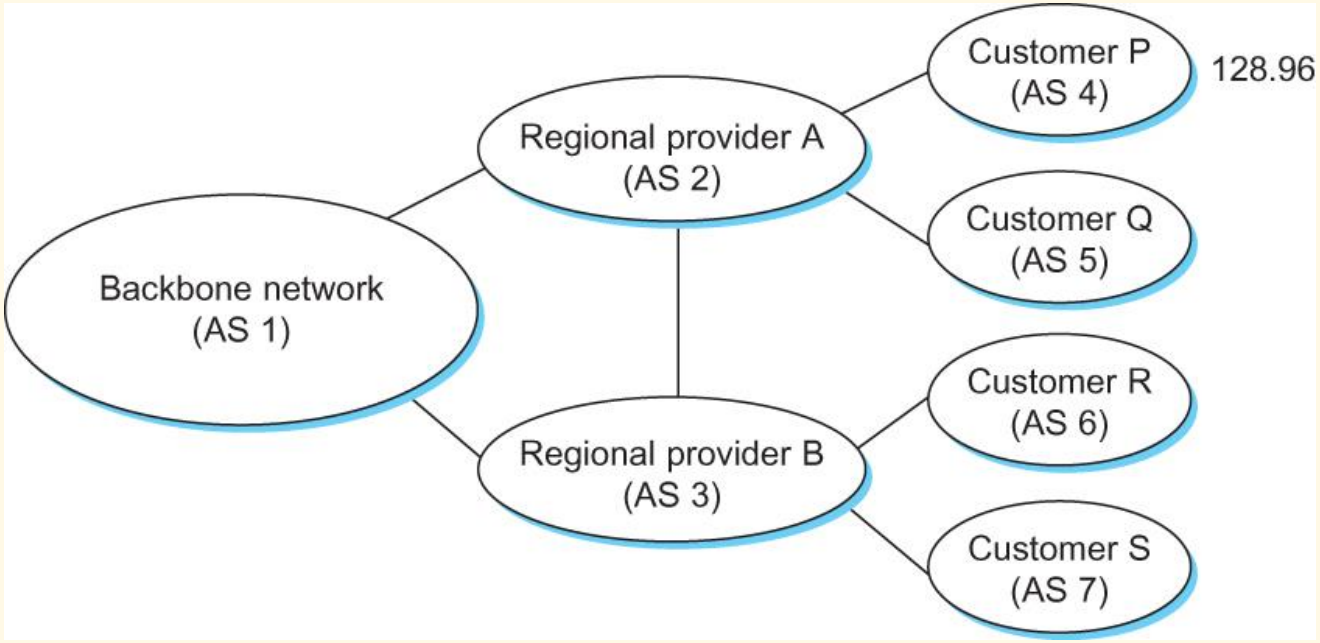


Figure 4.6 Example of loop among autonomous systems

Network Layer Topics

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What is inside a router?
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 ICMP

K & R

Address Resolution Protocol (ARP)

- A mechanism is needed to translate IP addresses into DL layer addresses that make sense on that network (e.g., 48-bit Ethernet addresses on an Ethernet LAN).

  Address pair == (hostID, Network Point of Attachment address)

where

hostID == IP address

NPA address == Ethernet address

- This translation is needed by a local interior router (e.g., R1 in Figure 3.14) and by hosts on the LAN (e.g., H1- H3 in Figure 3.14)

Routing in an internet

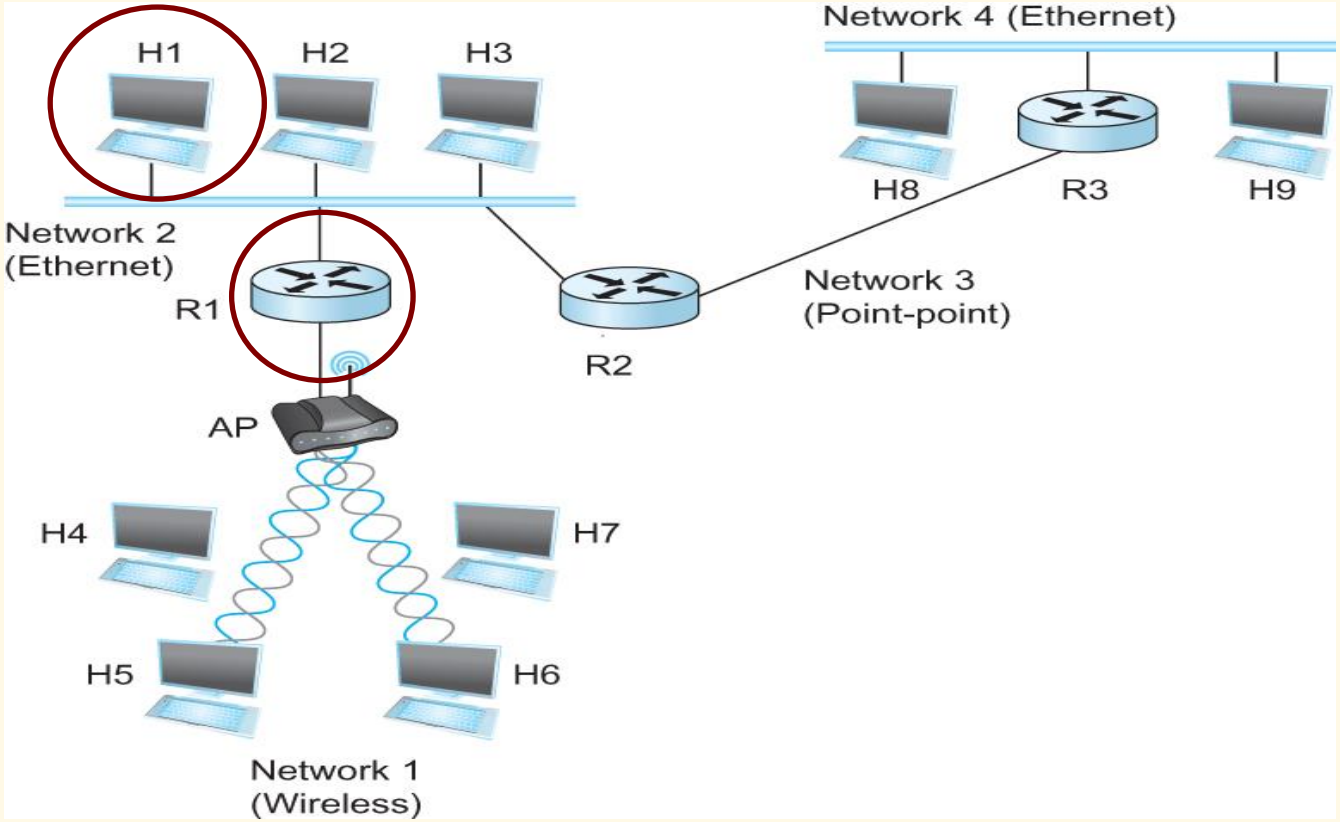


Figure 3.14 A Simple internetwork with Three Routers

Address Resolution Protocol (ARP)

- One solution :: have the router maintain a table of address pairs for all hosts attached to the LAN. Table is then copied to each host on the LAN.
  - This will generate lots of extra traffic on the LAN.

* a better solution:: each host dynamically learns the contents of this address pairs table using the LAN.

- This is the function performed by ARP (Address Resolution Protocol).

Address Resolution Protocol (ARP)

- Each node maintains an address pairs table in its cache.
- If IP address of destination host is NOT found in the cache, it invokes ARP over the LAN.
- An ARP query for desired IP address is broadcast on the LAN. The query includes address pair of sending host.
- Each IP host receives the query and checks to see if it matches its own IP address.

Address Translation Protocol (ARP)

- The target host sends an ARP reply with its DL layer address and adds sending address pair to its local table.
- Query originator updates information in its cached ARP table.
- Note - target hosts will 'refresh' their cache entry if originator is in local cache, but will not make a new table entry otherwise.
- ARP cache entries time out periodically (e.g., every 15 minutes) and non-refreshed entries are discarded.

ARP Packet Format

0	8	16 … 31
Hardware type = 1		ProtocolType = 0x0800
HLen = 48	PLen = 32	Operation
SourceHardwareAddr (bytes 0–3)		
SourceHardwareAddr (bytes 4–5)		SourceProtocolAddr (bytes 0–1)
SourceProtocolAddr (bytes 2–3)		TargetHardwareAddr (bytes 0–1)
TargetHardwareAddr (bytes 2–5)		
TargetProtocolAddr (bytes 0–3)		

- **HardwareType: type of physical network (e.g., Ethernet)**
- **ProtocolType: type of higher layer protocol (e.g., IP)**
- **Hlen & Plen: length of link layer and protocol layer addresses**
- **Operation: request or response**

Figure 3.23 ARP packet format for mapping IP addresses into Ethernet addresses

IP Addresses: How to Get One?

Q: How does a host get IP address?

- hard-coded by system admin in a file
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config

- DHCP: Dynamic Host Configuration Protocol: A host dynamically gets address from a server.
  - A "plug-and-play" protocol

Host Configurations

- Ethernet addresses are configured into network by manufacturer and they are unique.
- IP addresses must be unique on a given internetwork but also must reflect the structure of the internetwork.
- Most host operating systems provide a mechanism to manually configure the host's IP information.
- Manual configuration disadvantages:
  - A lot of work to configure all the hosts in a large network.
  - Configuration process is error-prone.
- Automated configuration process is required.

DHCP

- Relies on the existence of a DHCP Server that provides configuration information.
- Uses UDP and ports 67 and 68.
- Has widespread use in wireless LANs.
- DHCP server maintains a pool of available addresses that are allocated on demand.

DHCP: Dynamic Host Configuration Protocol

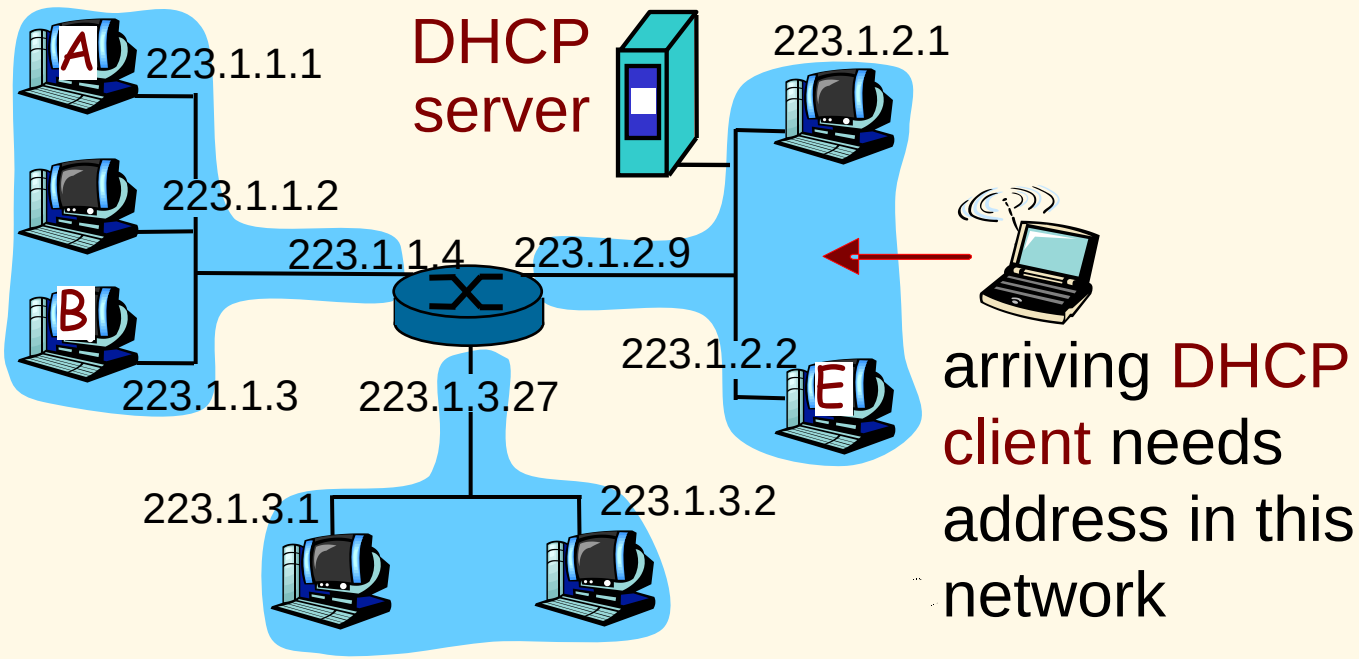
Goal: Allow a host to *dynamically* obtain its IP address from network server when it joins the network.

- Can renew its lease on address in use.
- Allows reuse of addresses (only hold address while connected and "on").
- Support for mobile users who want to join network (more shortly). {widely used in WLANs}

DHCP overview:

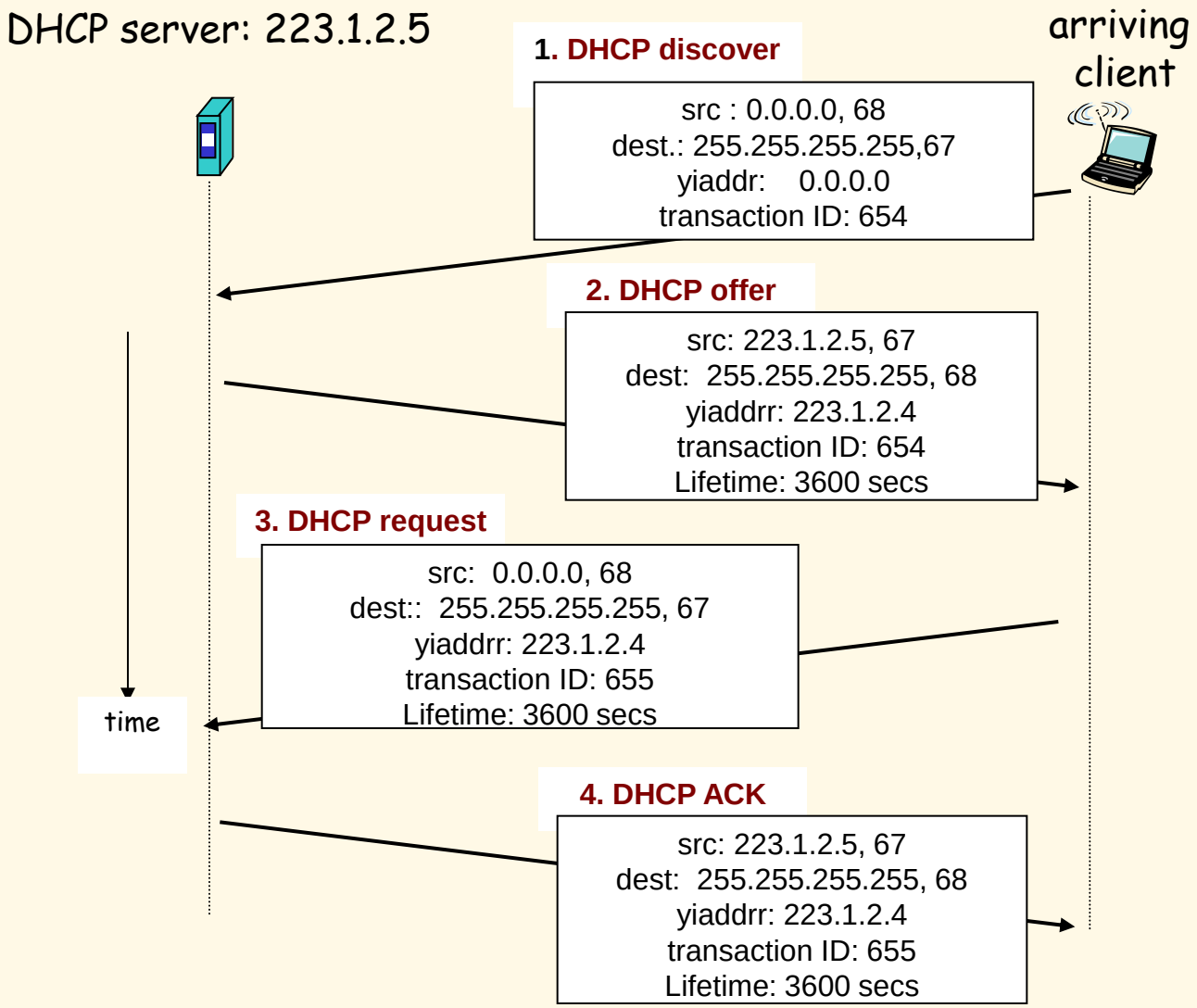
1. host broadcasts "DHCP discover" msg [optional]
2. DHCP server responds with "DHCP offer" msg [optional]
3. host requests IP address: "DHCP request" msg
4. DHCP server sends address: "DHCP ACK" msg

DHCP Client-Server Scenario



K & R

DHCP Client-Server Scenario



K & R

DHCP: More than IP address

DHCP can return more than just allocated IP address on subnet:

- address of first-hop router for client
- name and IP address of DNS sever
- network mask (indicating network versus host portion of address).

DHCP Relay Agent

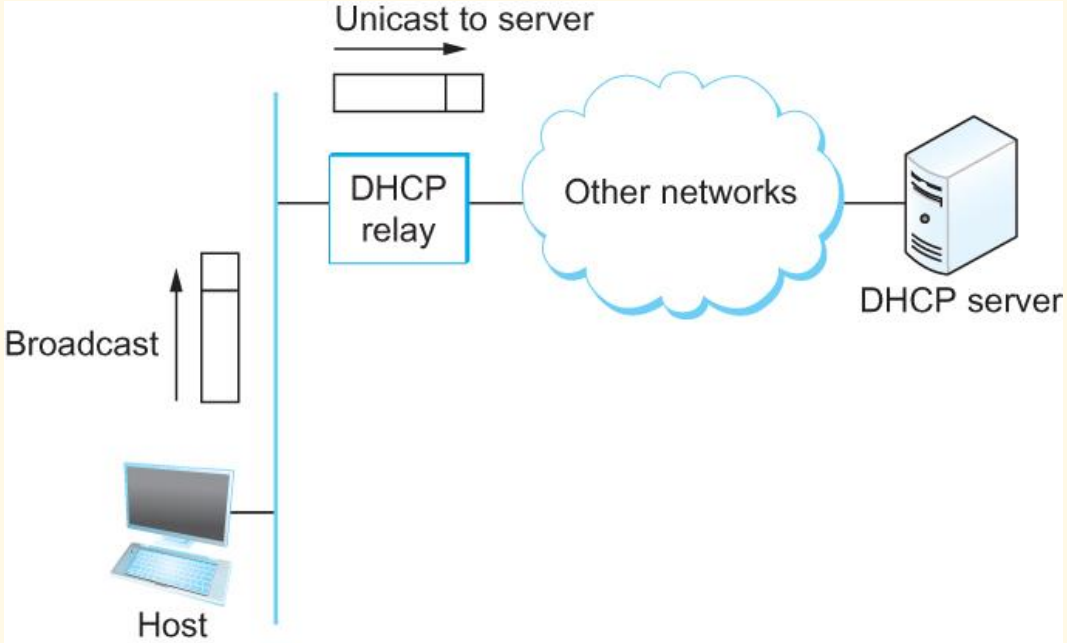
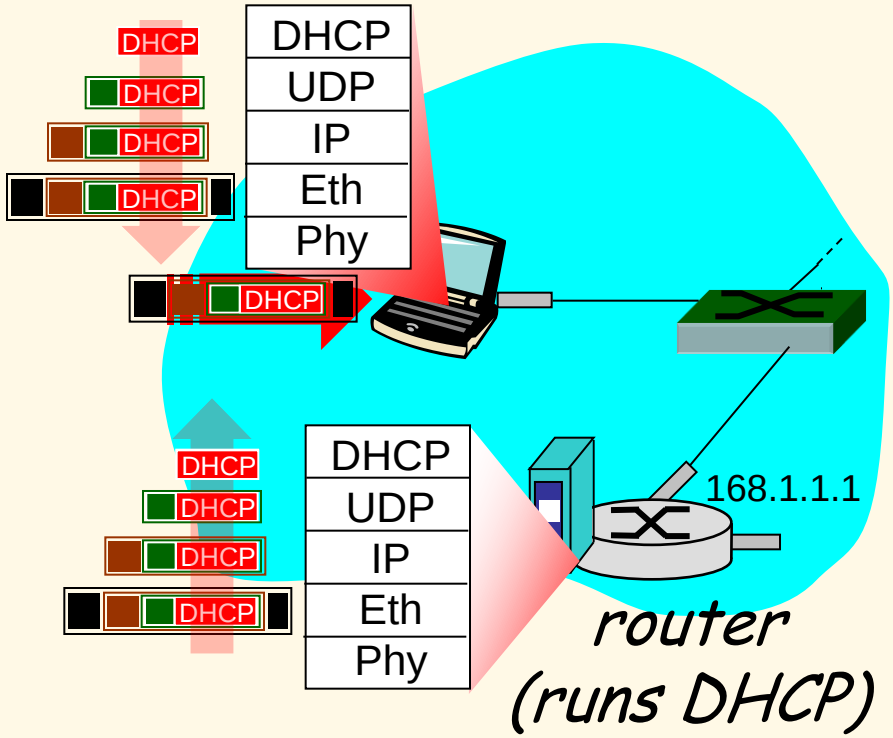


Figure 3.24 DHCP relay agent receives DISCOVER message

- To avoid a DHCP server on every network, DHCP relay agent unicasts the message to DHCP server and waits for the response.

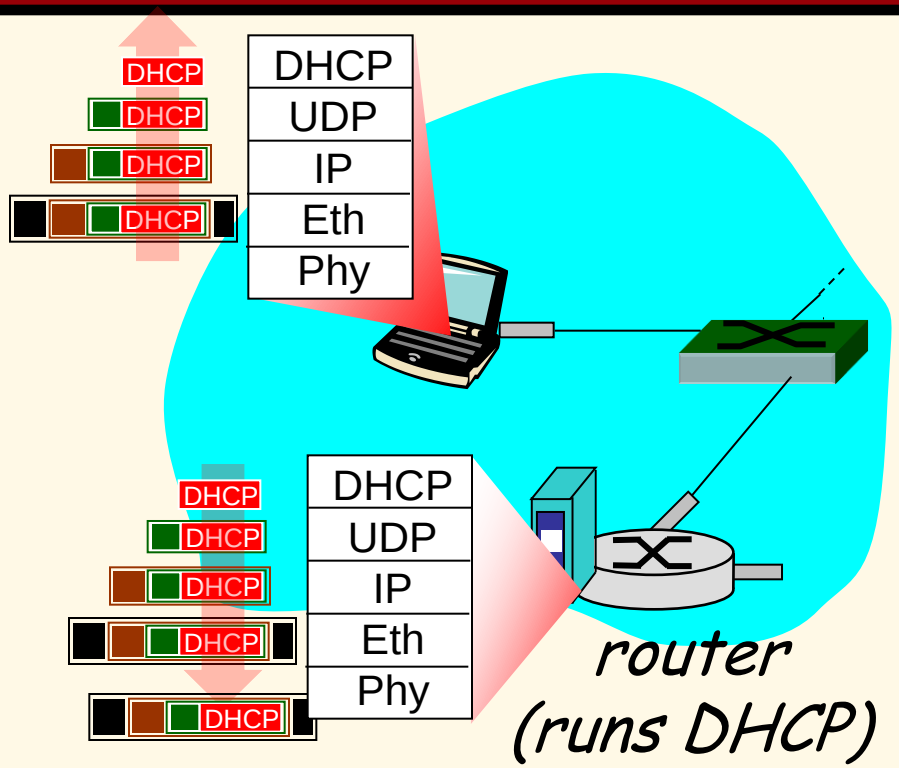
DHCP: Example



- **connecting laptop needs its IP address, addr of first-hop router, addr of DNS server: use DHCP**
  - DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.1 Ethernet.
  - Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server.
  - Ethernet demux'ed to IP demux'ed, UDP demux'ed to DHCP.

K & R

DHCP: Example



- DCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server

- encapsulation of DHCP server, frame forwarded to client, demux'ing up to DHCP at client.
- client now knows its IP address, name and IP address of DNS server, IP address of its first-hop router.

K & R

DHCP: Wireshark Output (home LAN)

reply

request

Message type: **Boot Request (1)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) **DHCP Message Type = DHCP Request**
Option: (61) Client identifier
    Length: 7; Value: 010016D323688A;
    Hardware type: Ethernet
    Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Option: (t=50,l=4) Requested IP Address = 192.168.1.101
Option: (t=12,l=5) Host Name = "nomad"
Option: (55) Parameter Request List
    Length: 11; Value: 010F03062C2E2F1F21F92B
    1 = Subnet Mask; 15 = Domain Name
    3 = Router; 6 = Domain Name Server
    44 = NetBIOS over TCP/IP Name Server
……

reply

Message type: **Boot Reply (2)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 192.168.1.101 (192.168.1.101)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 192.168.1.1 (192.168.1.1)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP ACK
Option: (t=54,l=4) Server Identifier = 192.168.1.1
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
Option: (t=3,l=4) Router = 192.168.1.1
Option: (6) Domain Name Server
    Length: 12; Value: 445747E2445749F244574092;
    IP Address: 68.87.71.226;
    IP Address: 68.87.73.242;
    IP Address: 68.87.64.146
Option: (t=15,l=20) Domain Name = "hsd1.ma.comcast.net."

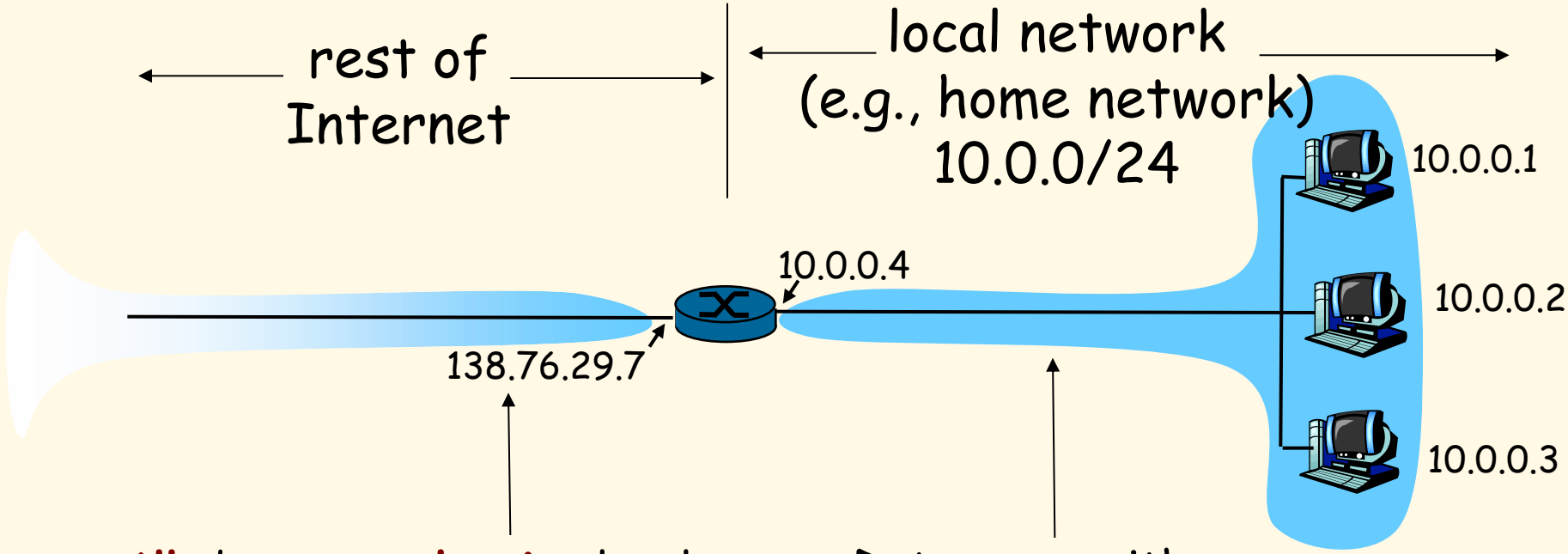
K & R

Network Layer Topics

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 ICMP

K & R

NAT: Network Address Translation



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

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

K & R

NAT: Network Address Translation

- **Motivation:** local network uses just one IP address as far as outside world is concerned:
  - range of addresses not needed from ISP: just one IP address for all devices.
  - can change addresses of devices in local network without notifying outside world.
  - can change ISP without changing addresses of devices in local network.
  - devices inside local net not explicitly addressable, visible by outside world (a security plus).

NAT: Network Address Translation

Implementation: NAT router must:

- **outgoing datagrams: replace** (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination address.

- **remember (in NAT translation table)** every (source IP address, port #) to (NAT IP address, new port #) translation pair.

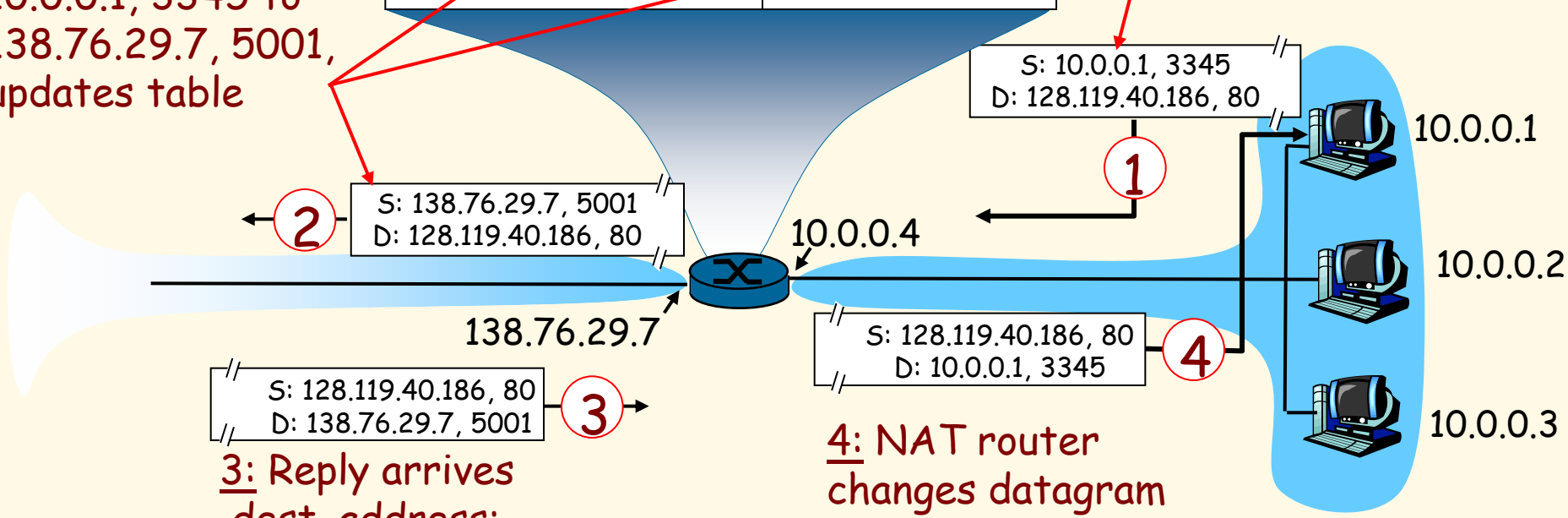
- **incoming datagrams: replace** (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table.

NAT: Network Address Translation

NAT translation table	
WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345

2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80



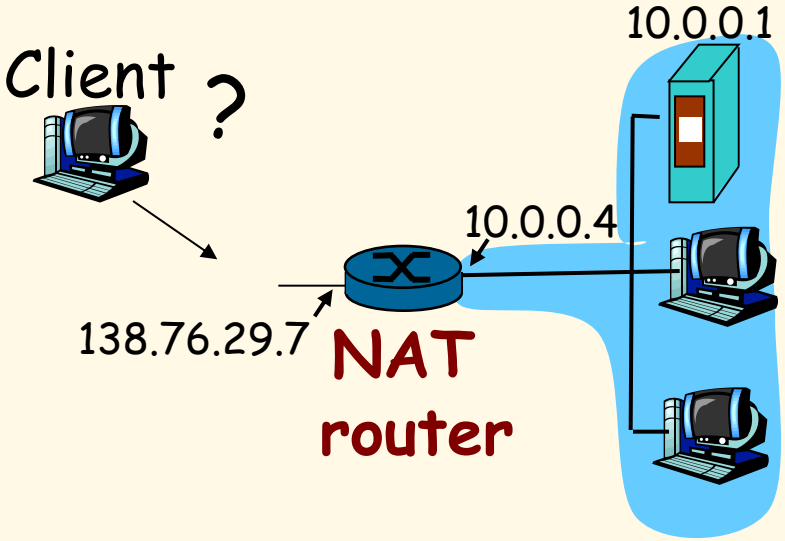
3: Reply arrives dest. address: 138.76.29.7, 5001

4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

K & R

NAT Traversal Problem

- client wants to connect to server with address 10.0.0.1
  - server address 10.0.0.1 local to LAN (client can't use it as destination addr)
  - only one externally visible NATted address: 138.76.29.7
- Solution 1: statically configure NAT to forward incoming connection requests at given port to server
  - e.g., (138.76.29.7, port 2500) always forwarded to 10.0.0.1 port 25000

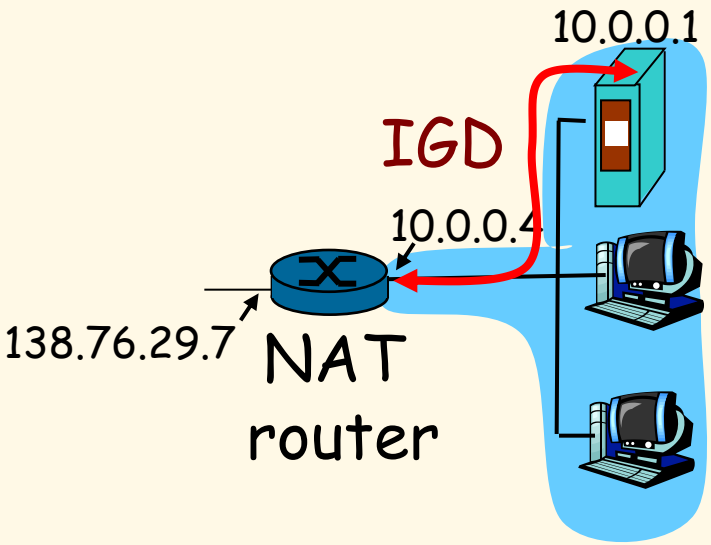


K & R

NAT Traversal Problem

- Solution 2: Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol allows NATted host to:
  - ❖ learn public IP address (138.76.29.7)
  - ❖ add/remove port mappings (with lease times)

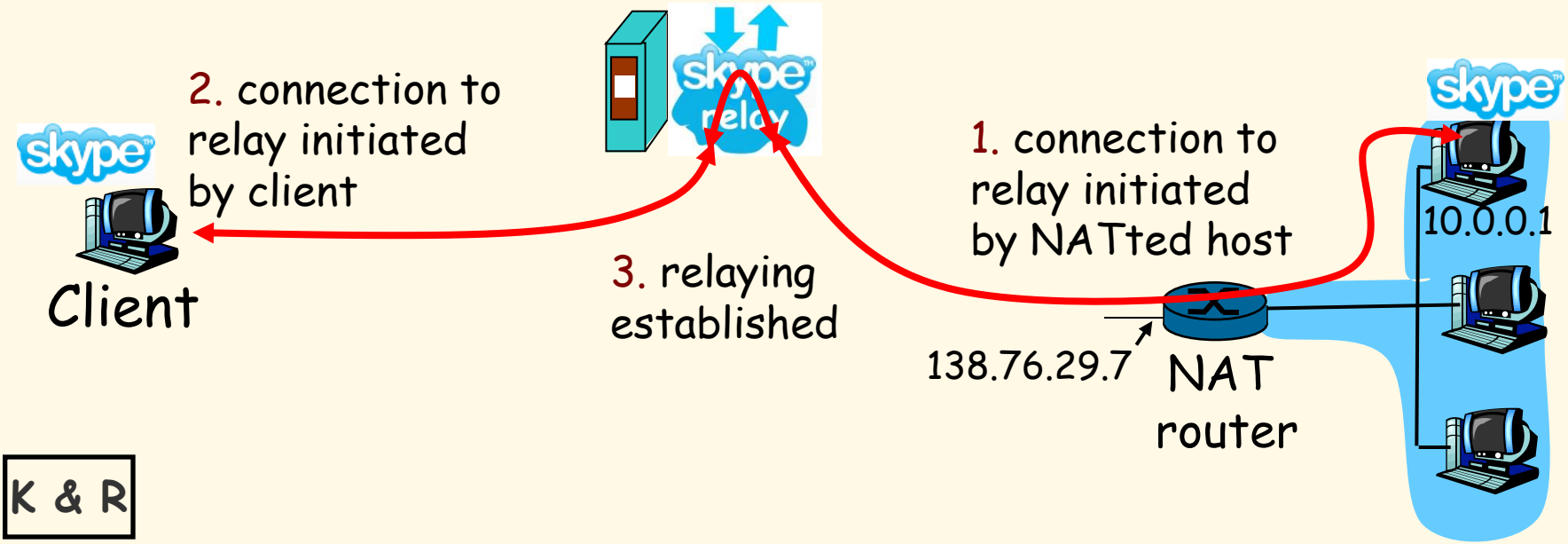
  i.e., automate static NAT port map configuration.



K & R

NAT Traversal Problem

- Solution 3: relaying (used in Skype)
  - NATed client establishes connection to relay node.
  - External client connects to relay.
  - Relay bridges packets between to connections.



K & R

Network Layer Topics

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What is inside a router?
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- ICMP

K & R

Internet Control Message Protocol (ICMP)

- IP is always configured with a companion protocol (ICMP) that defines a collection of error messages that are sent back to the source host whenever a router or host is unable to process an IP datagram successfully, e.g.,
  - Destination host unreachable due to link /node failure.
  - Reassembly process failed.
  - TTL had reached 0 (so datagrams don't cycle forever)
  - IP header checksum failed
- ICMP includes several control messages that a router can use to send back to a source host.
  - ICMP-Redirect
  - Sent from router R1 to a source host to inform host there exists a better path through R2 to a particular destination.
  - Contains better route information.

Other ICMP functions

- ping uses ICMP echo messages to determine if a node is reachable and alive. Can be used to estimate RTT.
- traceroute is used to determine the set of routes along the path to a desintination.

* Note - some routers send ICMP packets through a separate 'special' queue.

Network Layer Summary

- 4.1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What is inside a router?
- 4.4 IP: Internet Protocol
  - Datagram Format
  - IPv4 addressing
  - Subnets
  - CIDR
  - ARP & DHCP
  - NAT
- 4.5 Routing algorithms
  - Algorithm Classification
  - Link State
  - Distance Vector
  - Hierarchical Routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- ICMP

K & R