HOST-HOST Communication, 1970

- Paul Barran, On Distributed Communication Networks, IEEE Transactions on Communication Systems, Volume 12, No. 1, March 1964

- Introduced Concepts of
  - Distributed Networks
  - Routing (hot-potato-routing)
  - Packet-Switching (message-block)

- Name worth mentioning: Bob Kahn

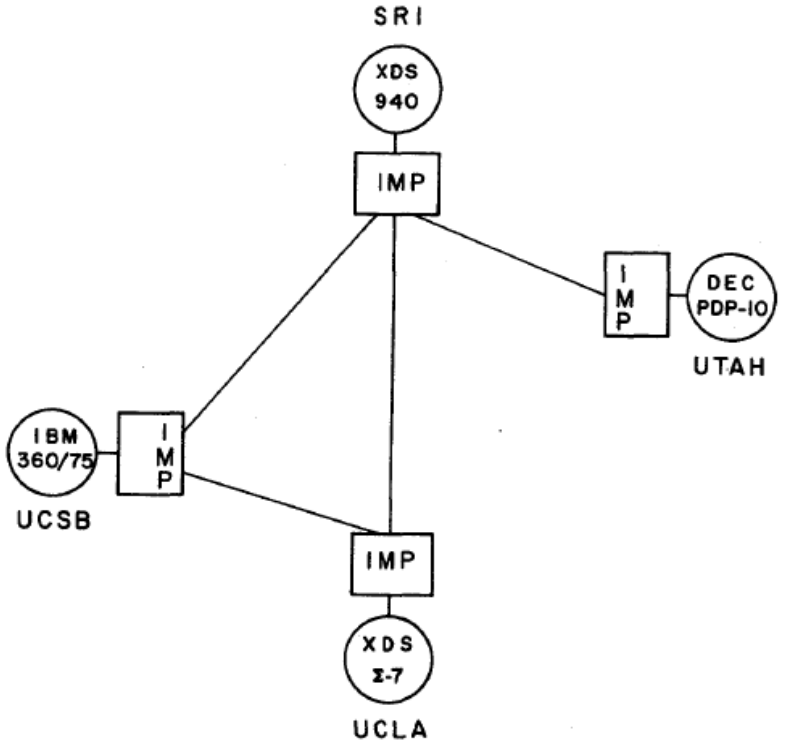
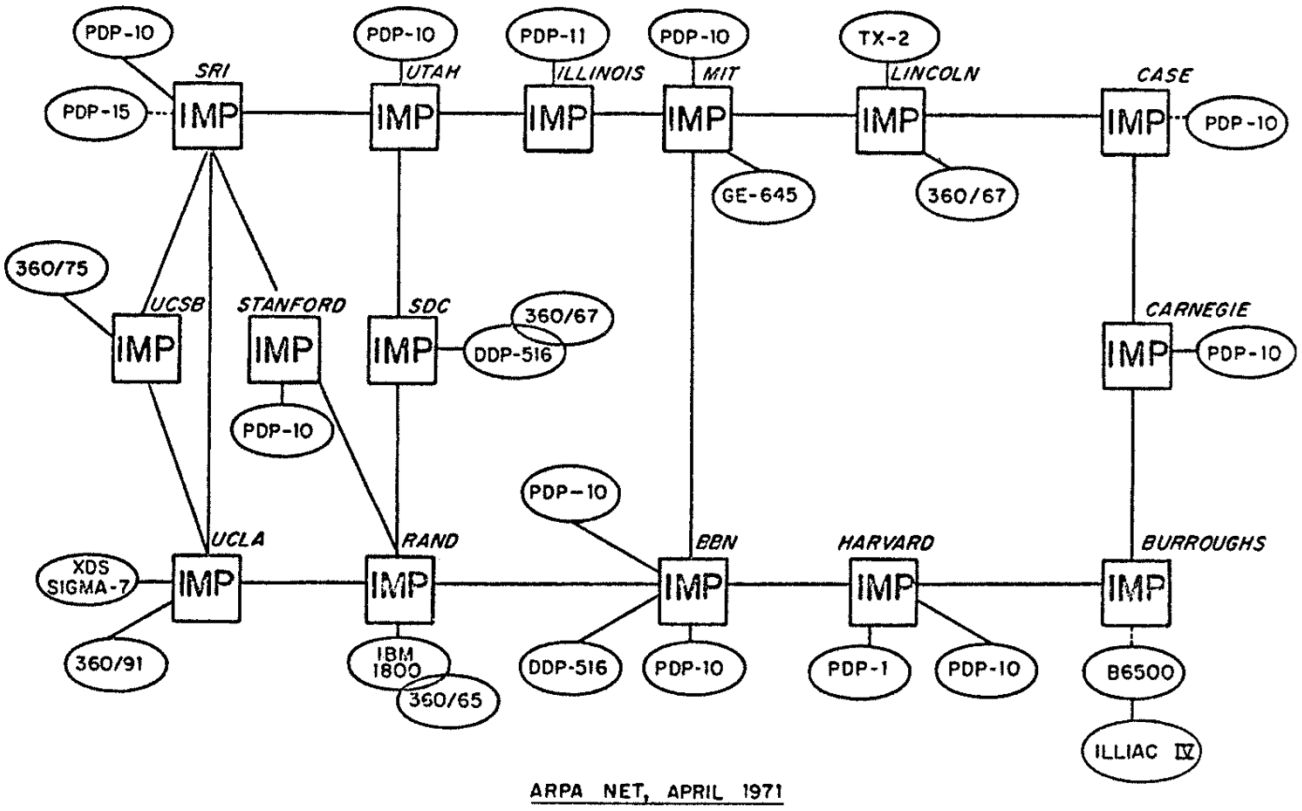


Figure 1—Initial network configuration

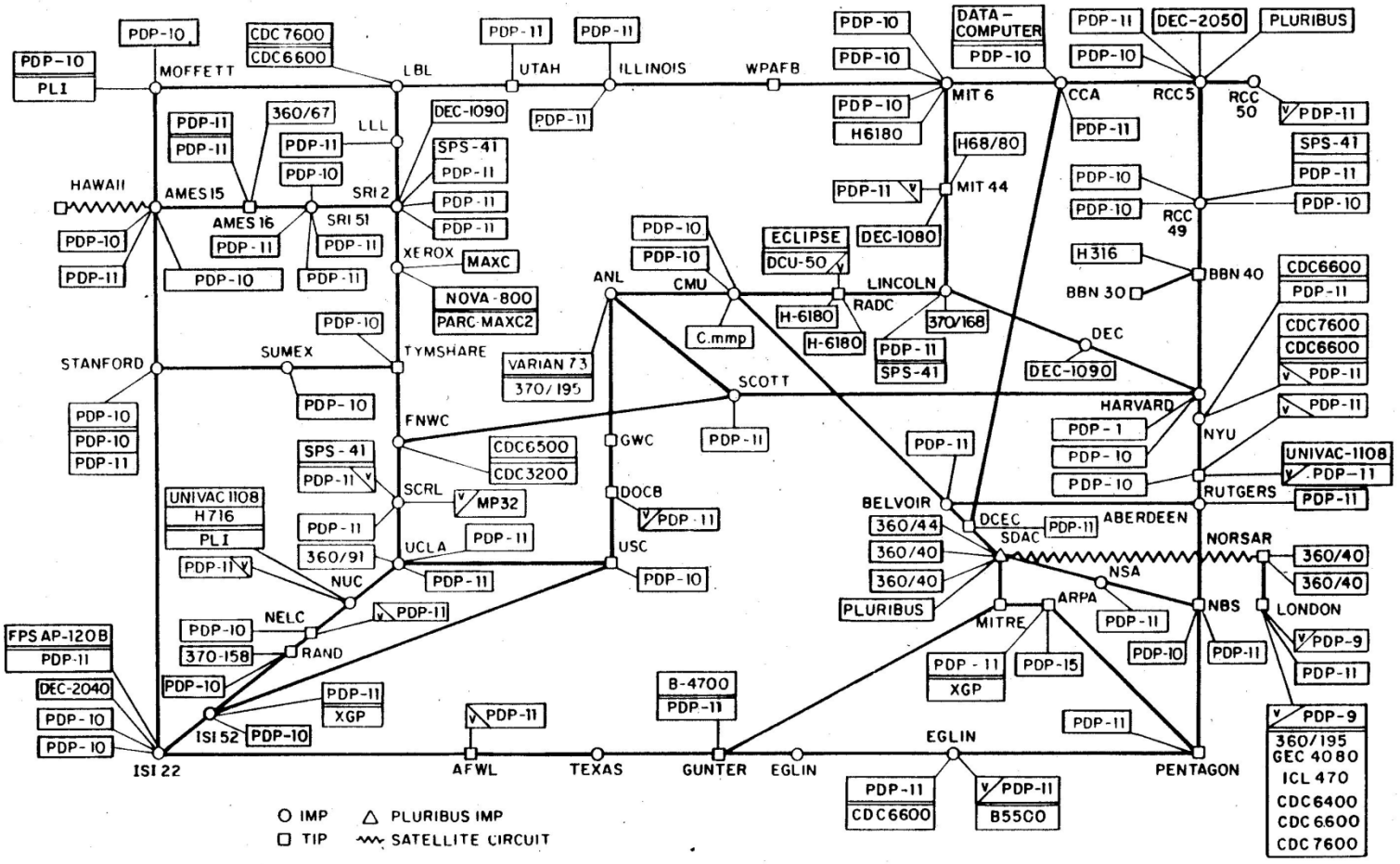
AT&T's comment was that packet-switching will never be useful.

* C. Stephen Carr, Stephen D. Crocker, and Vinton G. Cerf, HOST-HOST Communication Protocol in the ARPA Network, Spring Joint Computer Conference, pages 589-597, Atlantic City, NJ, USA, May 1970

ARPANET 1971



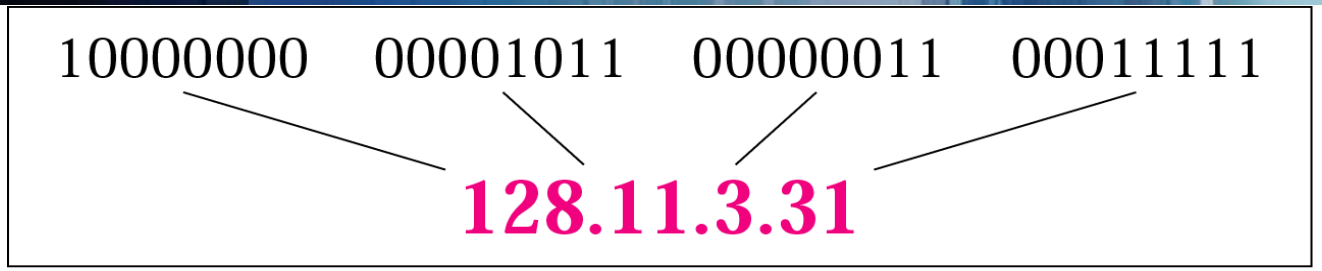
ARPANET 1977



(PLEASE NOTE THAT WHILE THIS MAP SHOWS THE HOST POPULATION OF THE NETWORK ACCORDING TO THE BEST INFORMATION OBTAINABLE, NO CLAIM CAN BE MADE FOR ITS ACCURACY)

NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES

IP Addresses



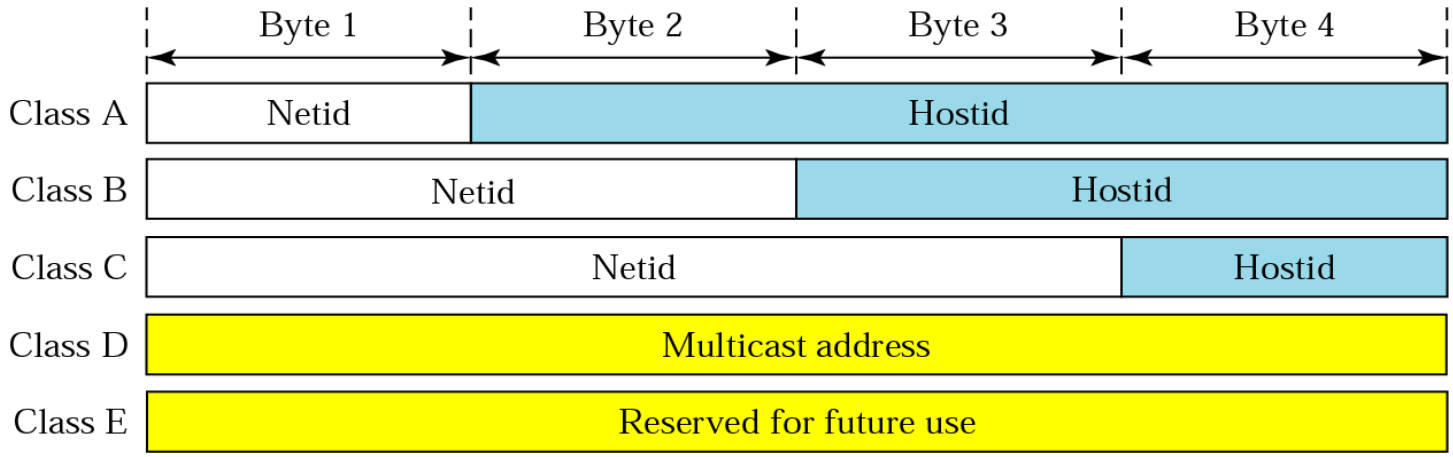
- 32-bit number
  - 4.294.967.296 addresses
- IP addresses are unique and universal
  - except private addresses

Range			Total
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

- Dotted decimal notation:
  - Bytes of binary notation represented as decimal separated by dot

* Figure is courtesy of B. Forouzan

Classful Addresses



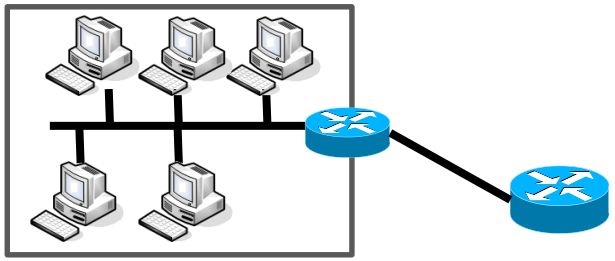
- Class A (international organisations)
  - 126 networks with 16,277,214 hosts each
- Class B (large companies)
  - 16,384 networks with 65,354 hosts each
- Class C (smaller companies)
  - 2,097,152 networks with 254 hosts each

- Inefficient use of hierarchical address space
  - Class C with 2 hosts (2/254 = 0.78% efficient)
  - Class B with 256 hosts (256/65534 = 0.39% efficient)

* Figure is courtesy of B. Forouzan

Network Layer

Trinity College

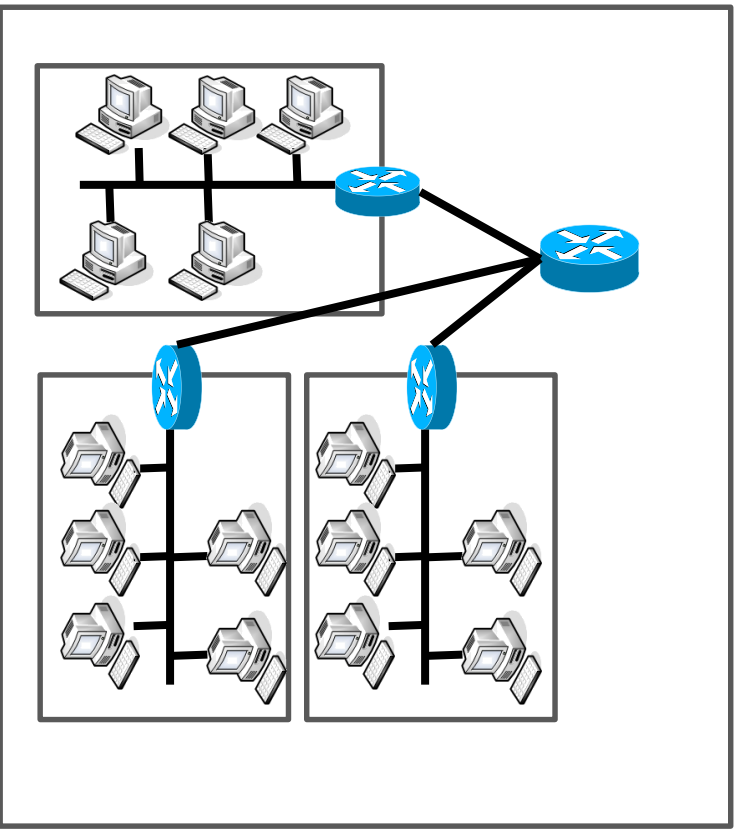


134.226.0.0

Class B

Network Layer

Trinity College

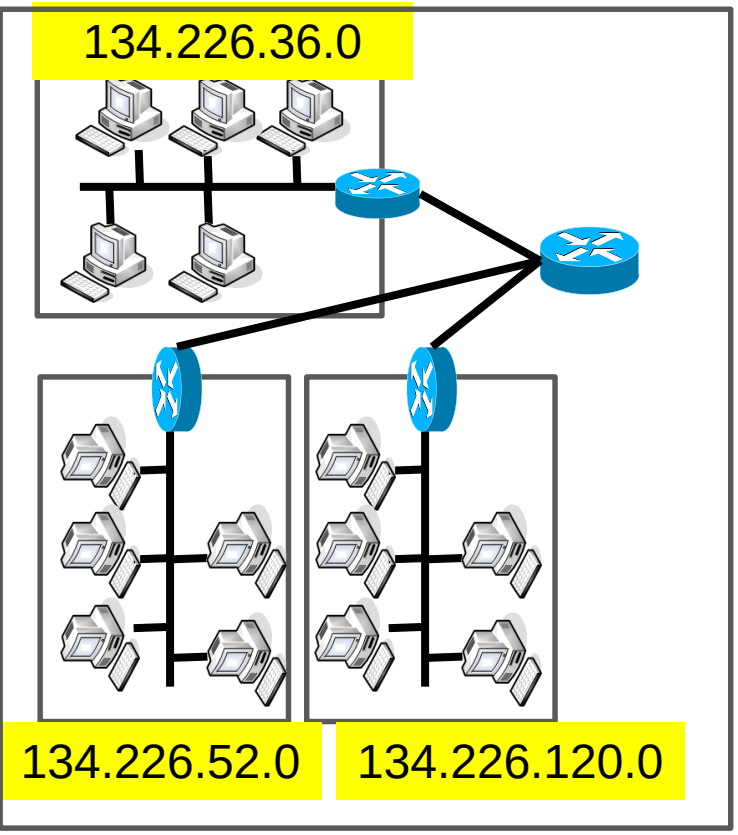


134.226.0.0

Class B

Network Layer

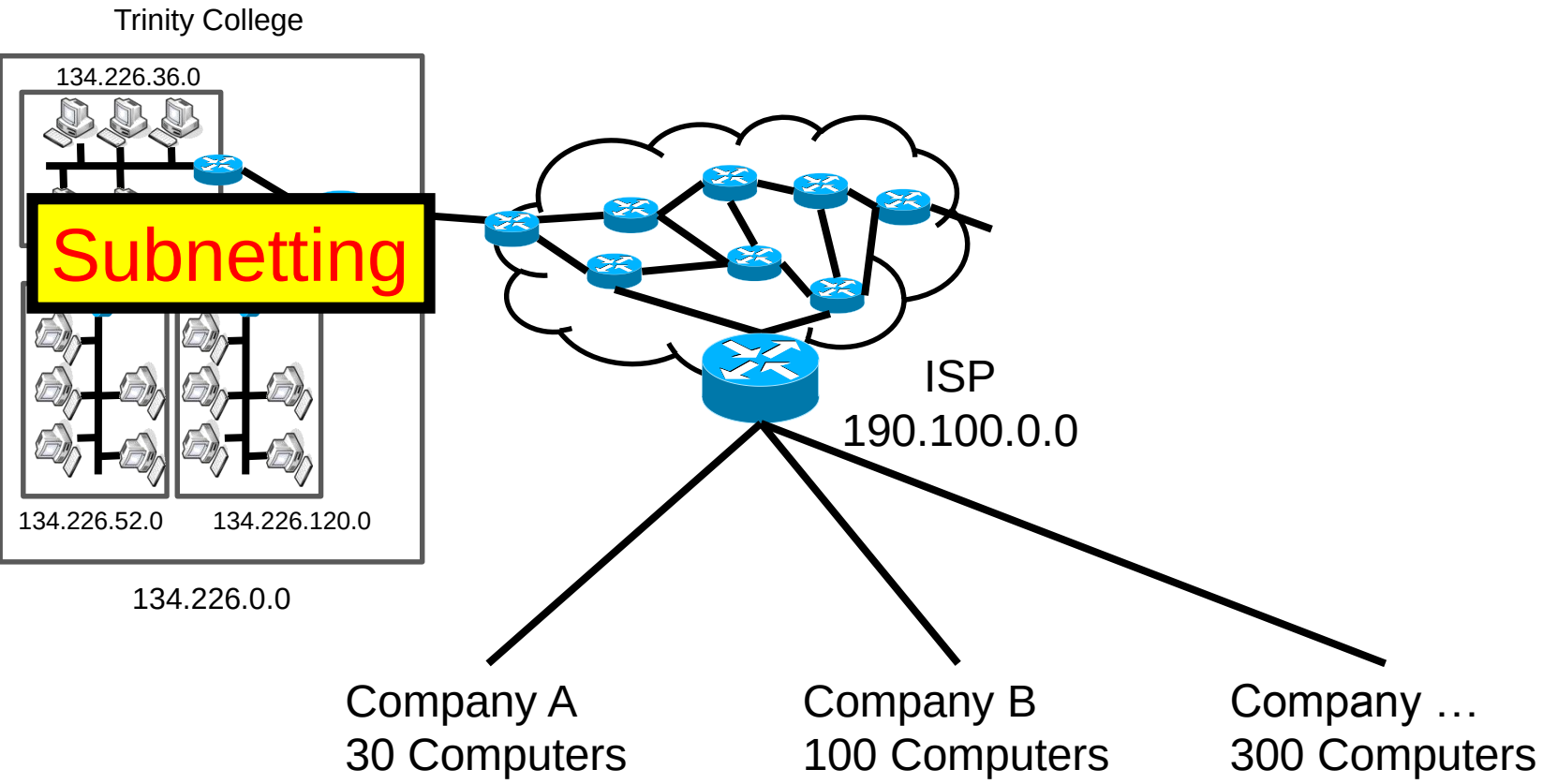
Trinity College



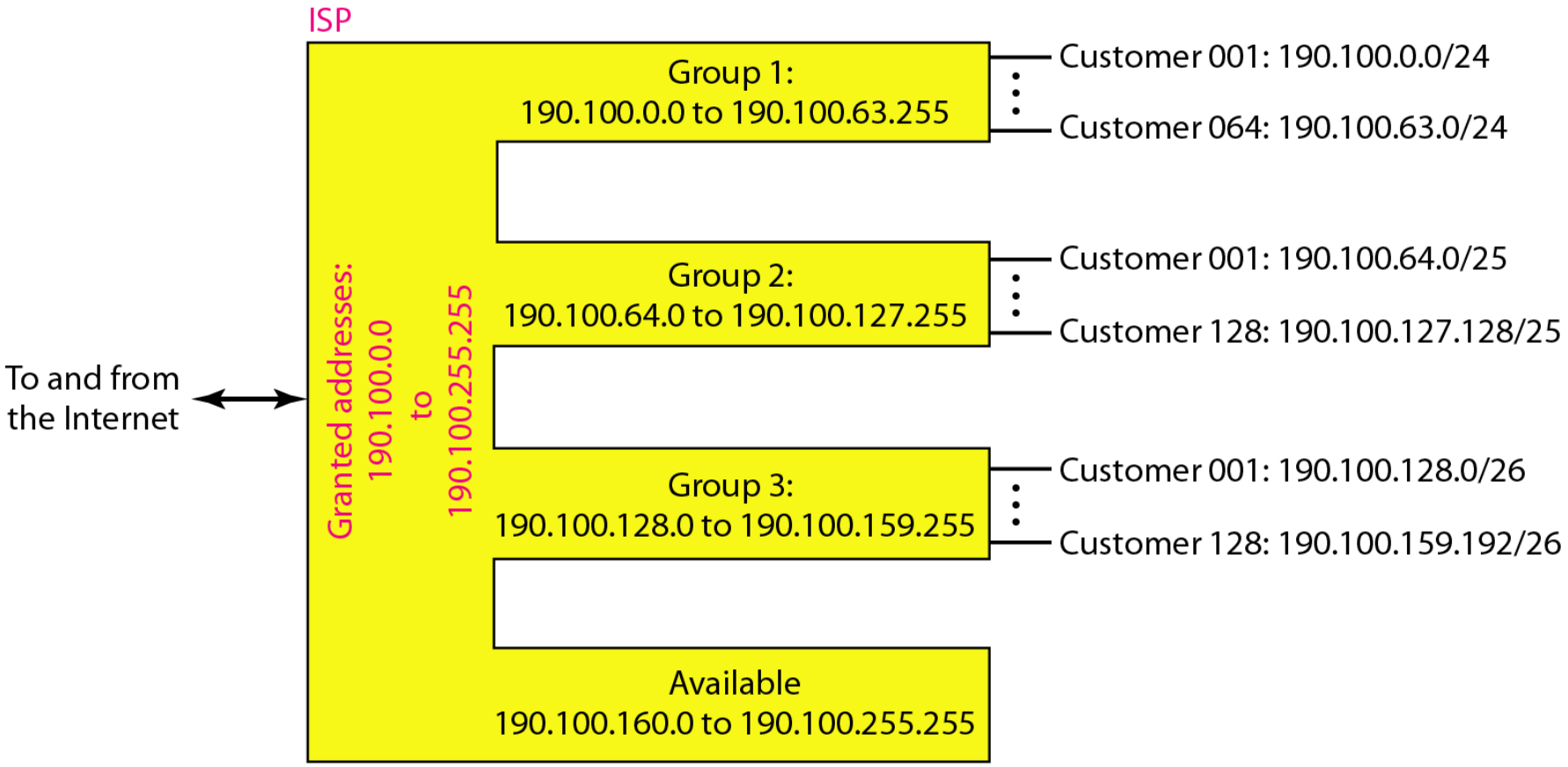
134.226.0.0

Class B ← Classful Addresses

Network Layer

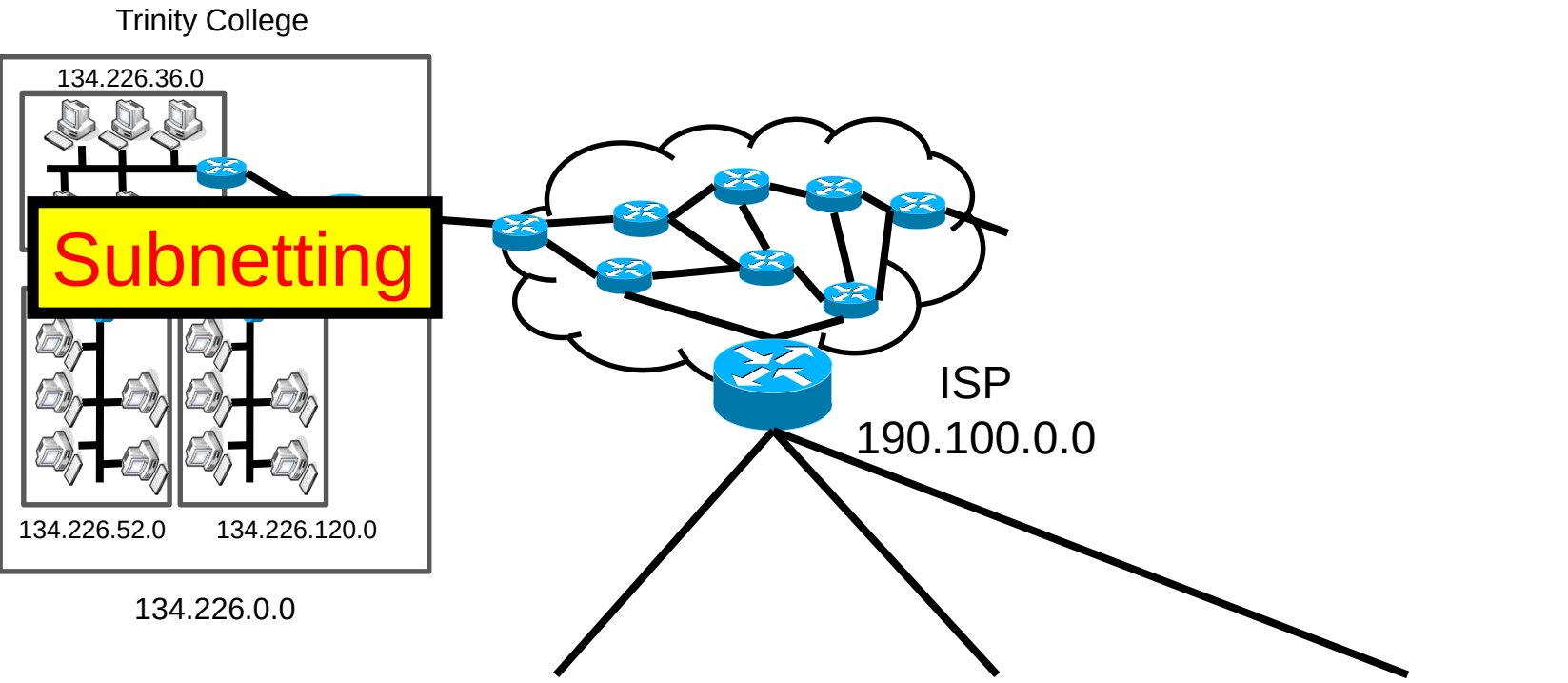


ISPs & Classless Addresses

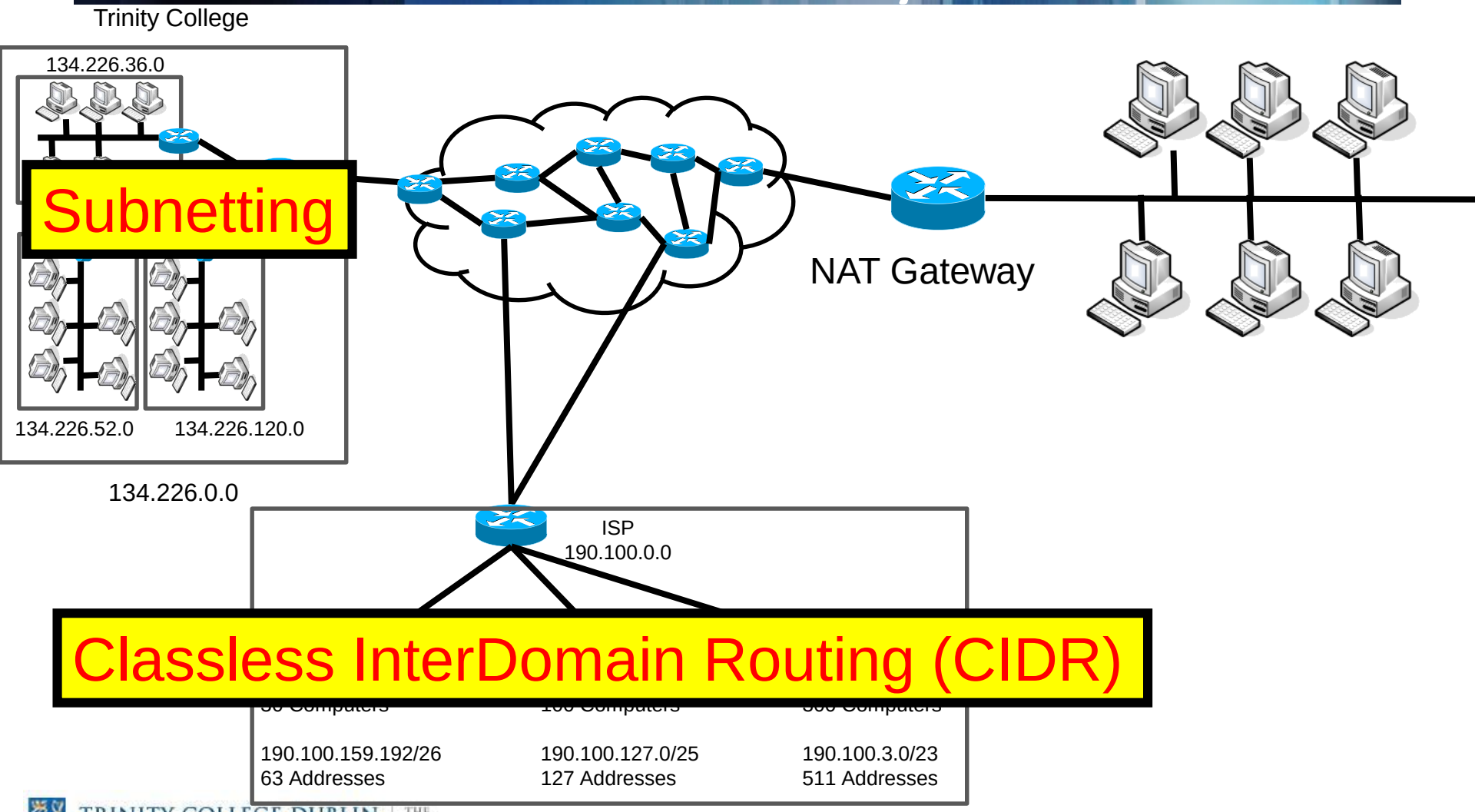


* Figure is courtesy of B. Forouzan

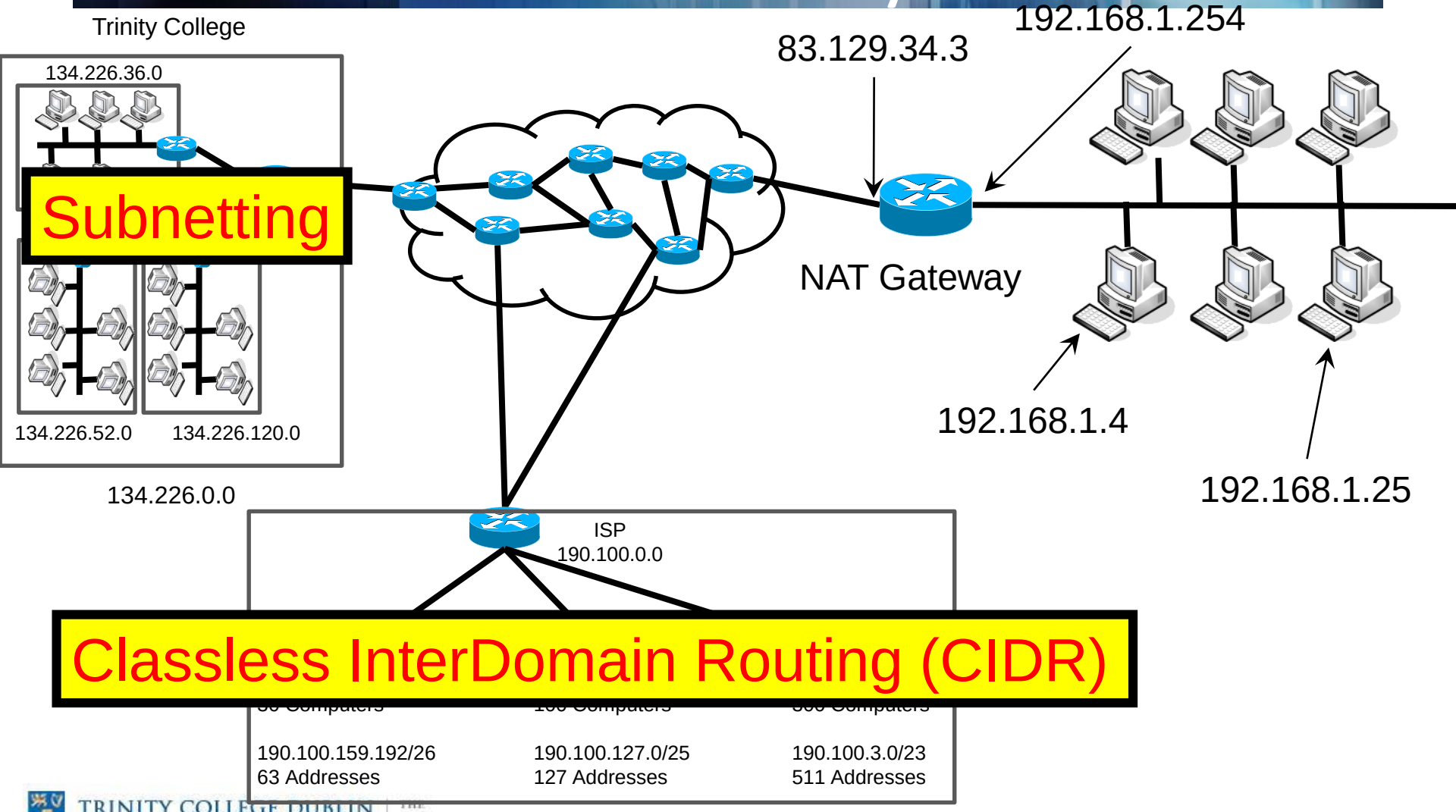
Network Layer



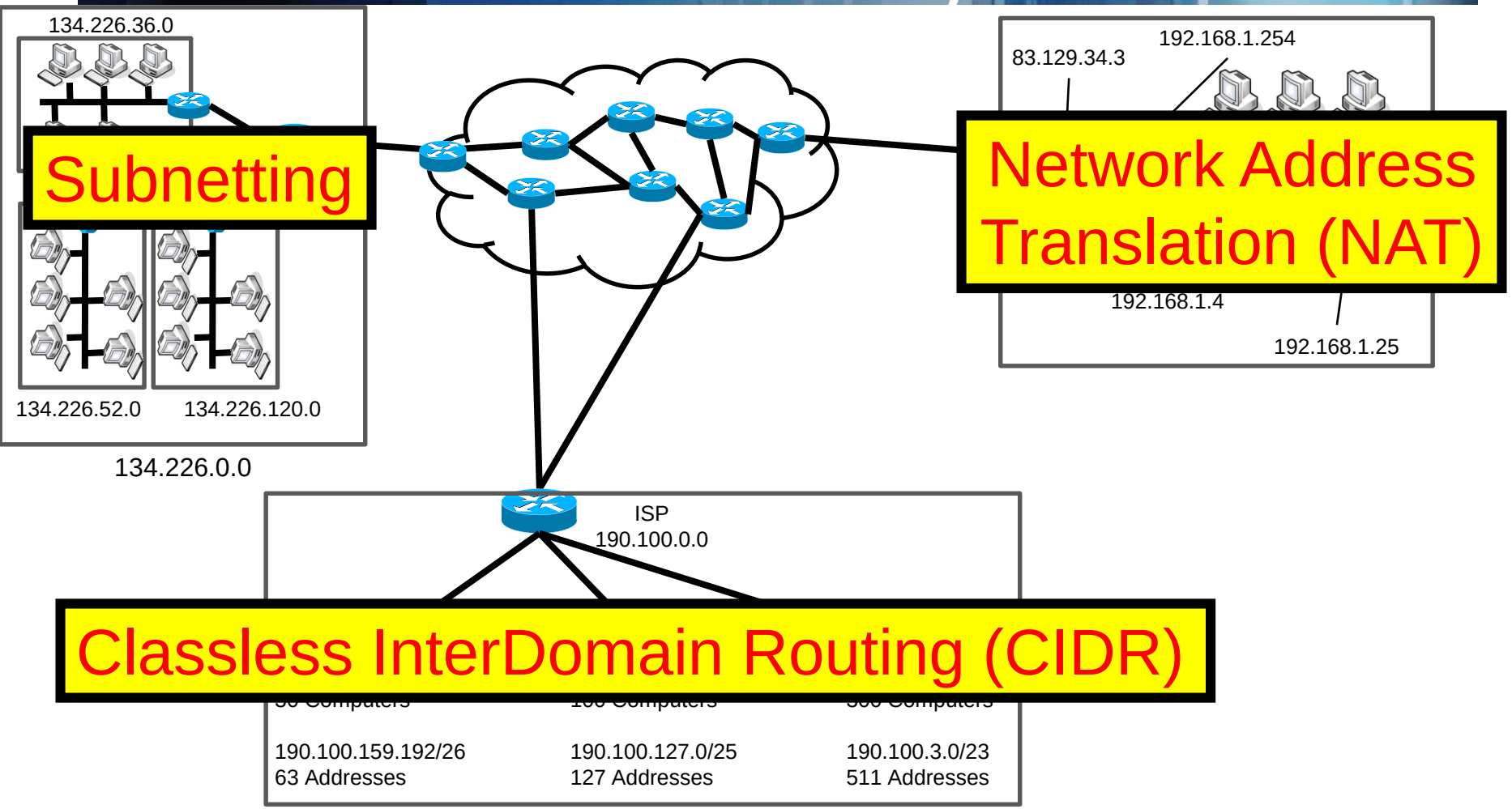
Network Layer



Network Layer



Network Layer



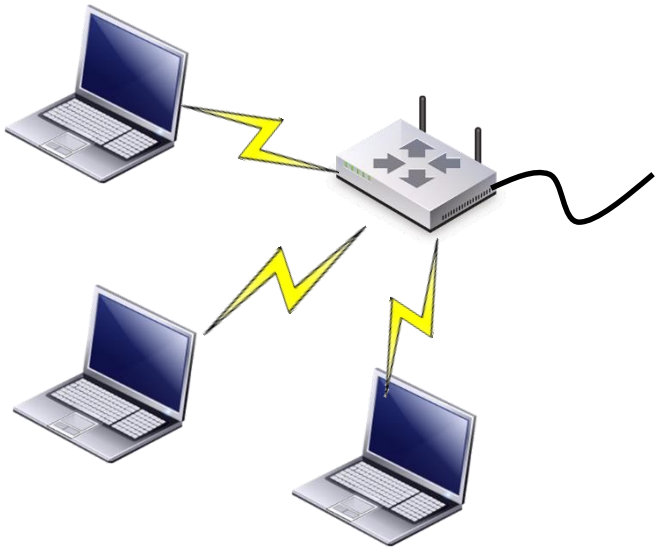
CS2031
Telecommunications II

Routing

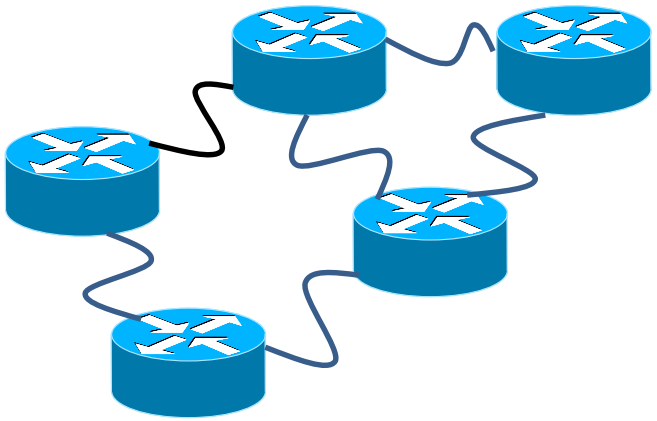
Outcome for Today

- Distance Vector Routing
- Link State Routing
- Multicast Routing
  - Dense Mode (DM)
  - Sparse Mode (SM)

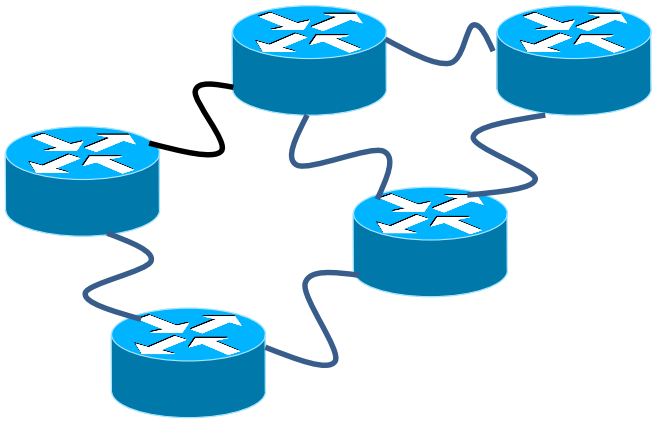
Home Network



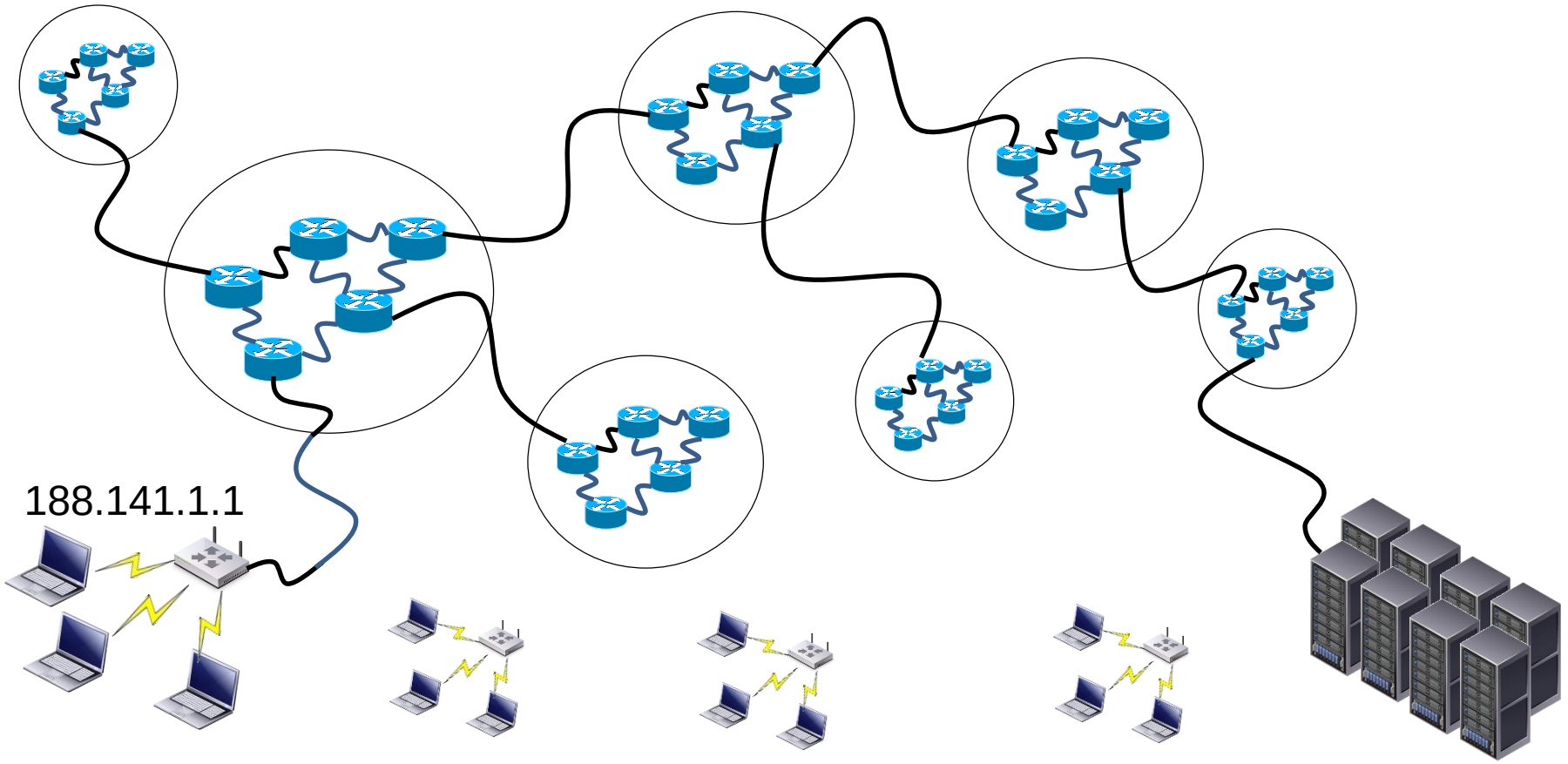
ISPs



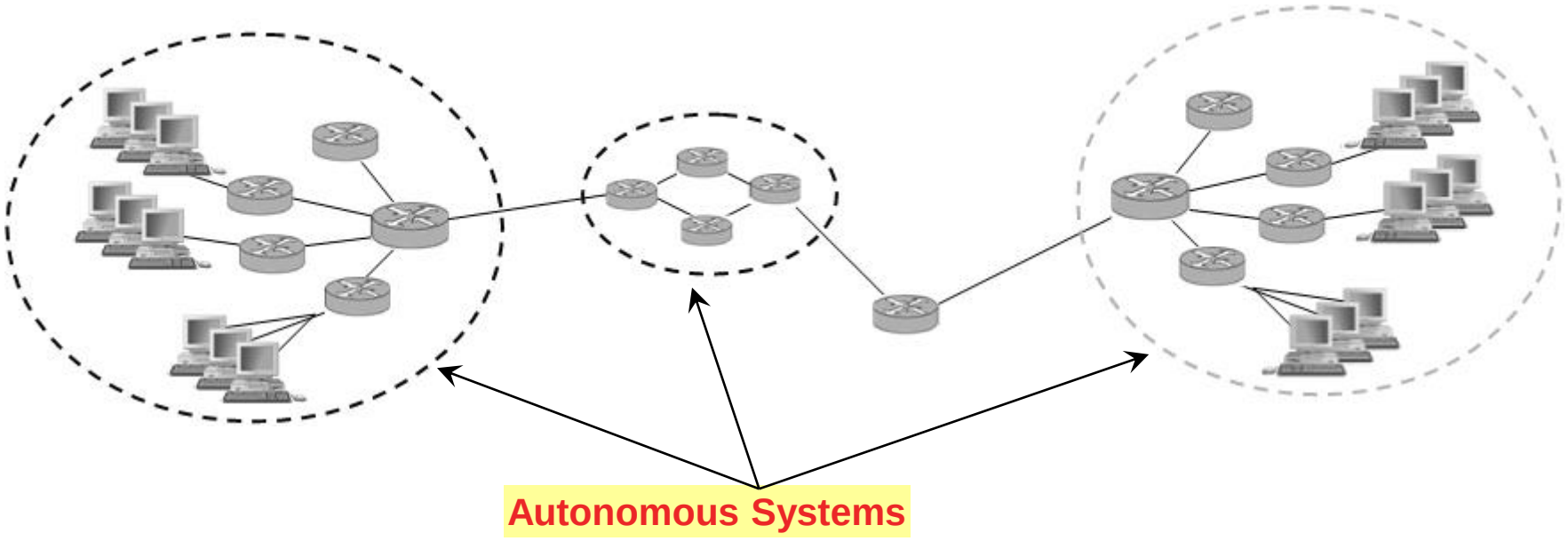
ISPs



Internet = Network of Networks

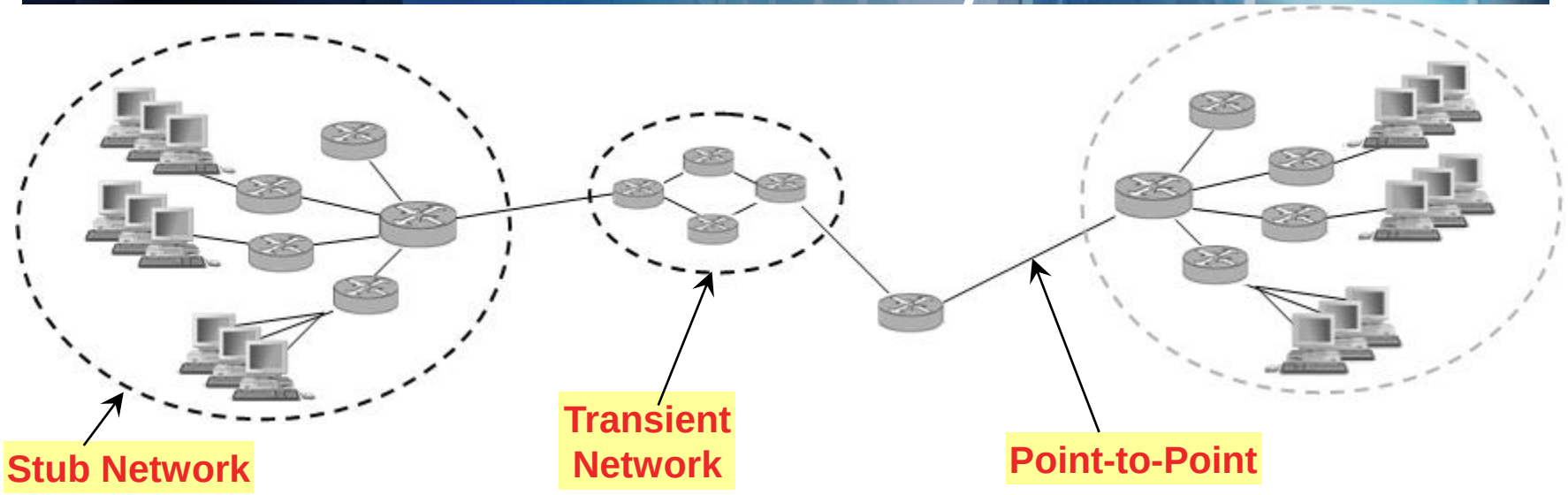


Structure of the Internet



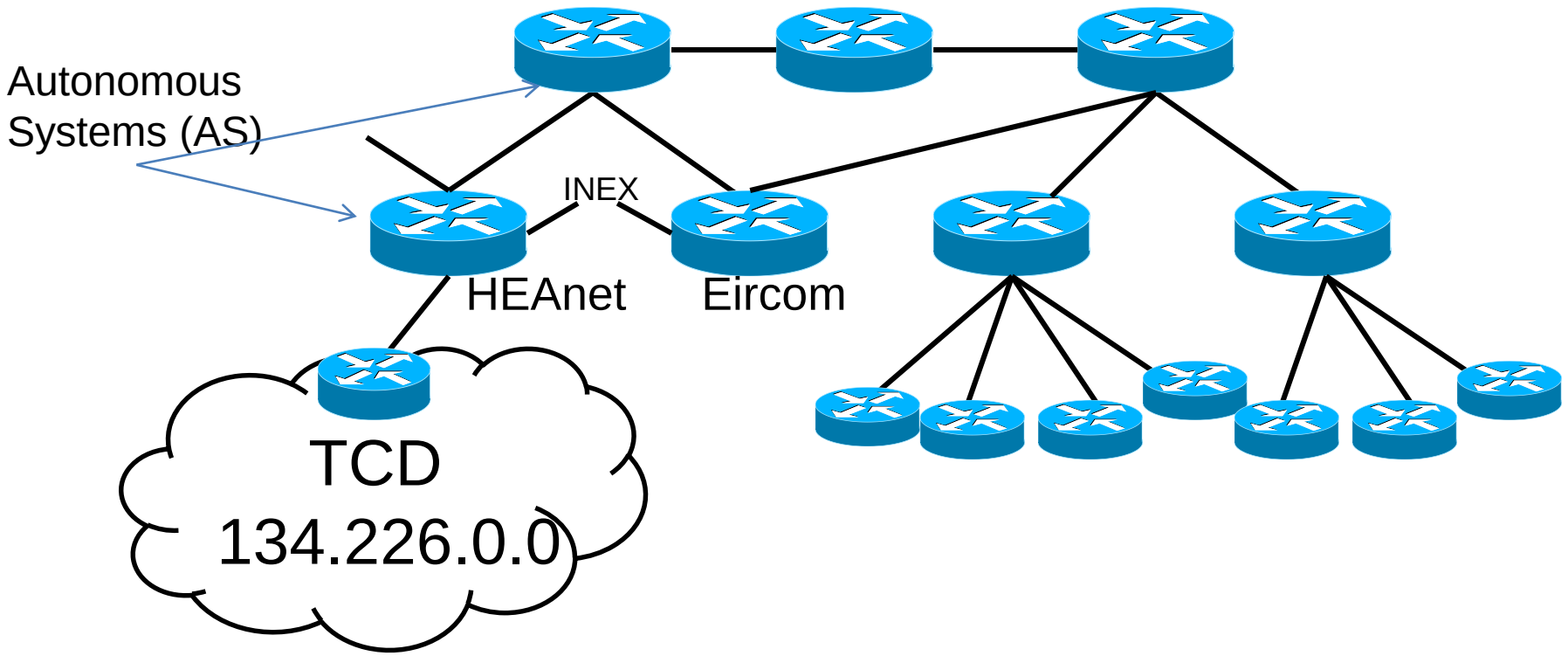
- Autonomous Systems
  - e.g. Companies, ISPs, 3rd-level Institutions

Autonomous Systems

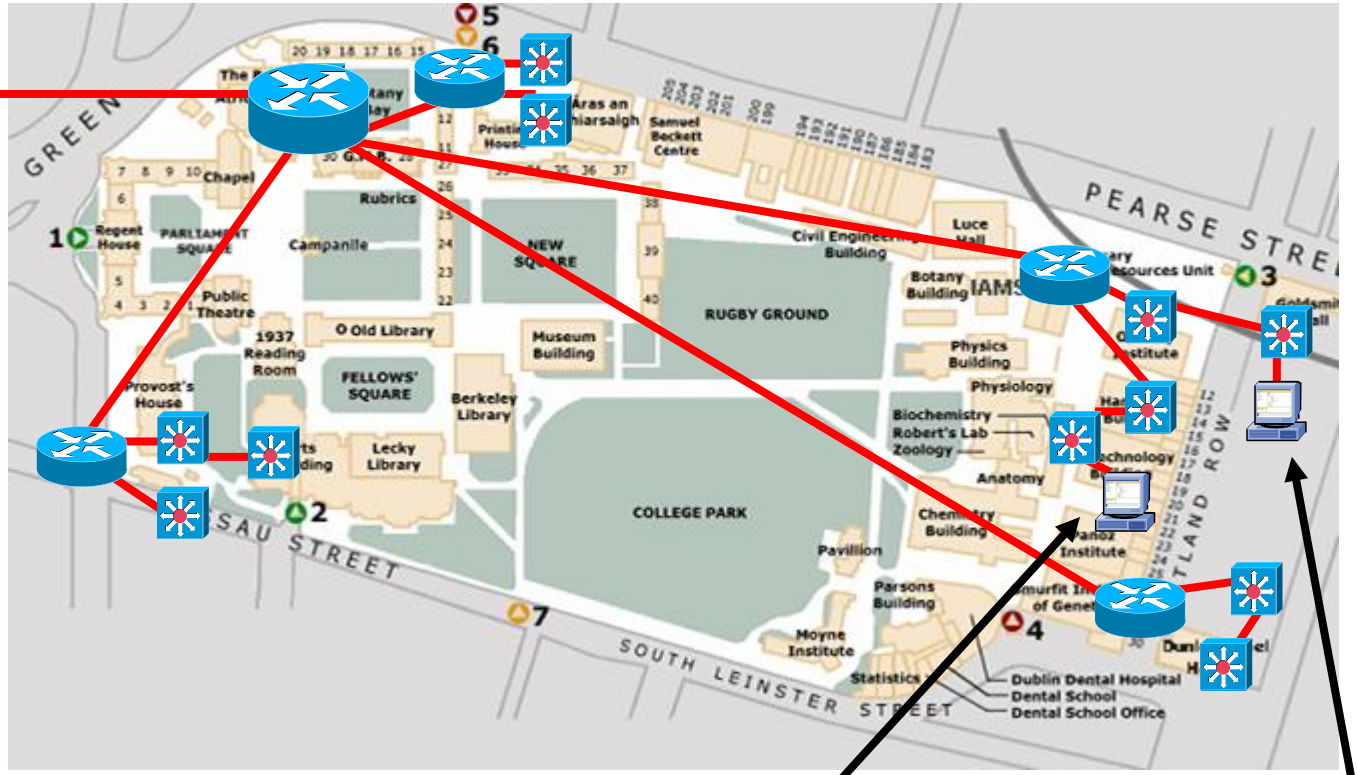


- Stub network
  - Network that does not forward to other network
- Transient network
  - Network that forwards traffic between other networks
- Point-to-point link

Trinity's Connection



Trinity Network with Routers



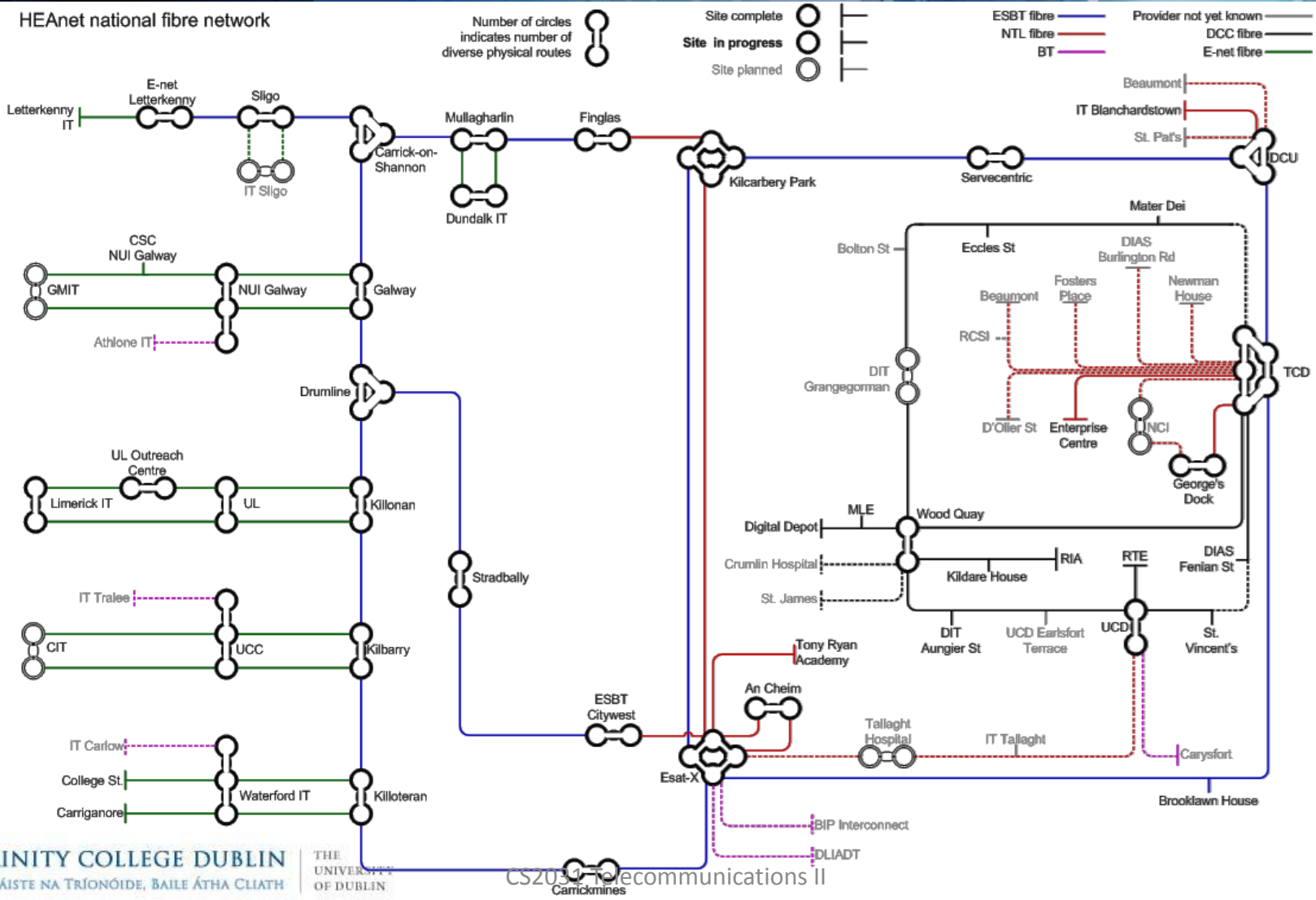
Switch

Router

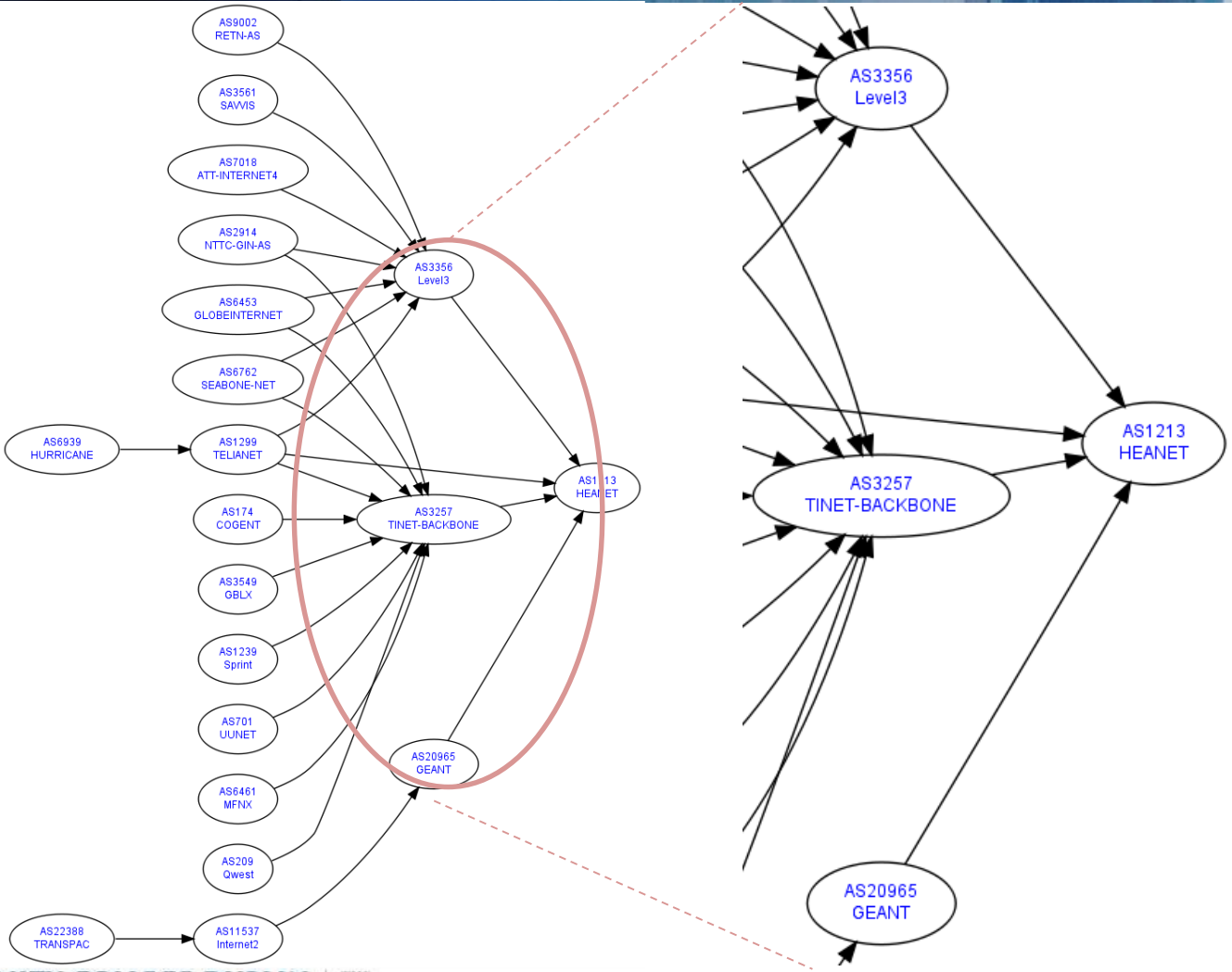
Address: 134.226.38.55
Subnet: 134.226.38.0

Address: 134.226.32.55
Subnet: 134.226.32.0

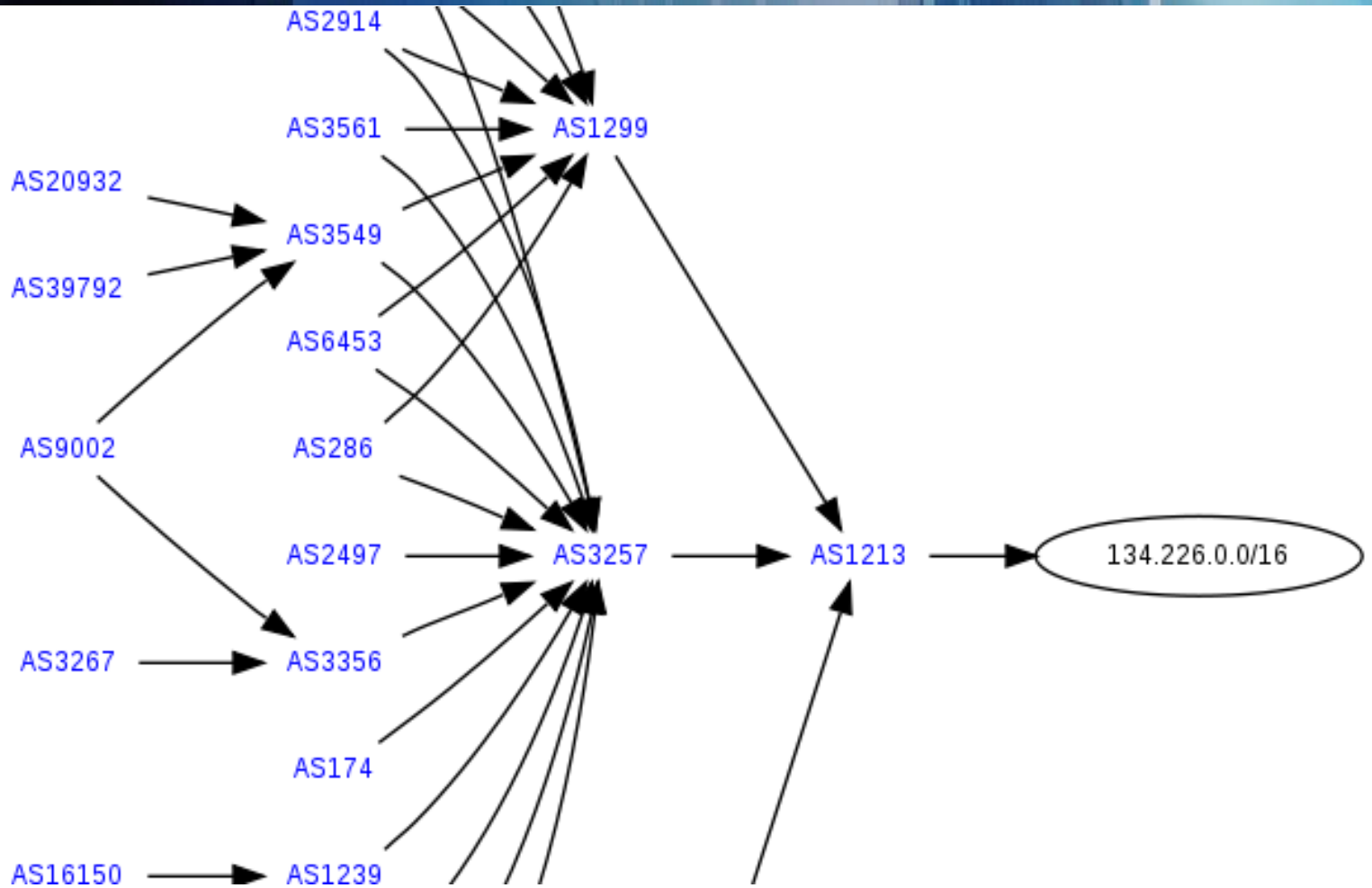
HEAnet Fibre Network



AS1213 - HEANET



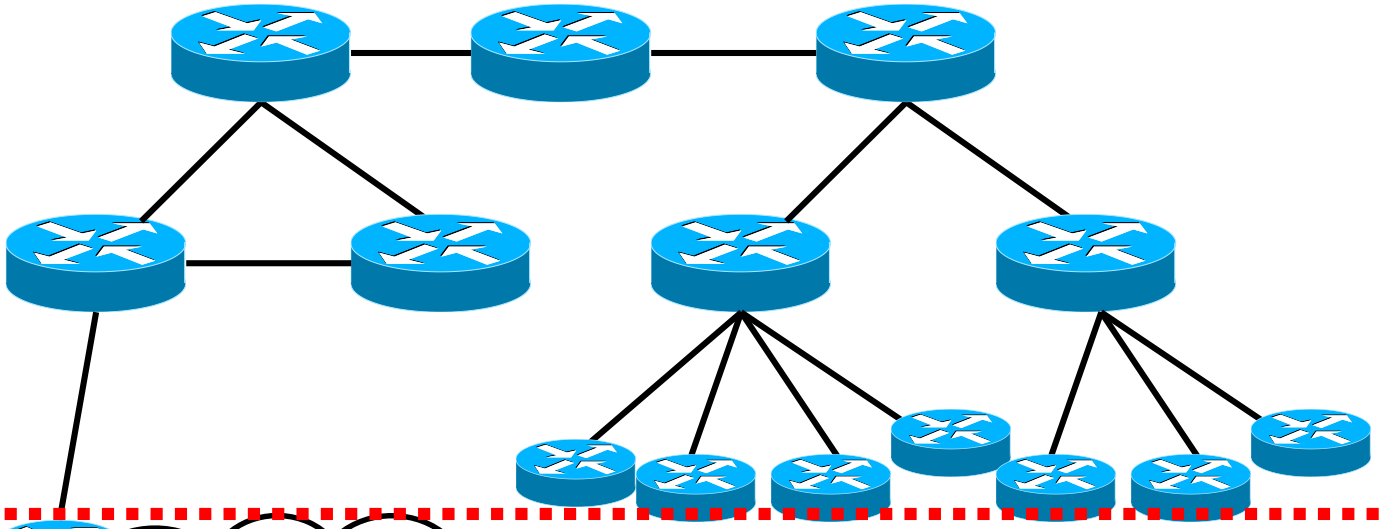
AS1213 - HEANET



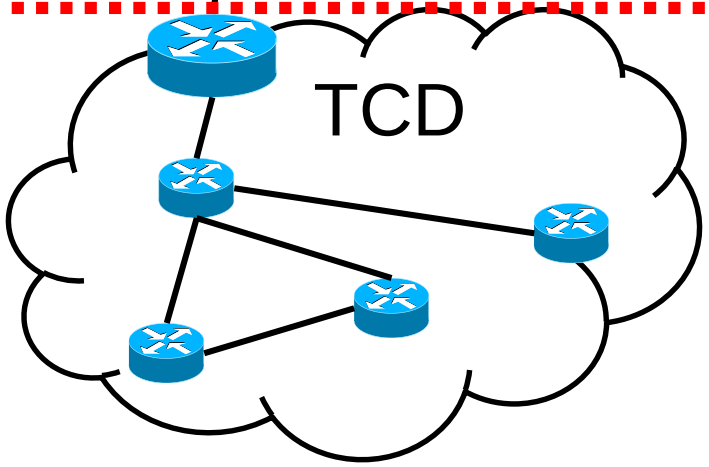
http://www.robtex.com/route/134.226.0.0-16.html

InterAS vs IntraAS Routing

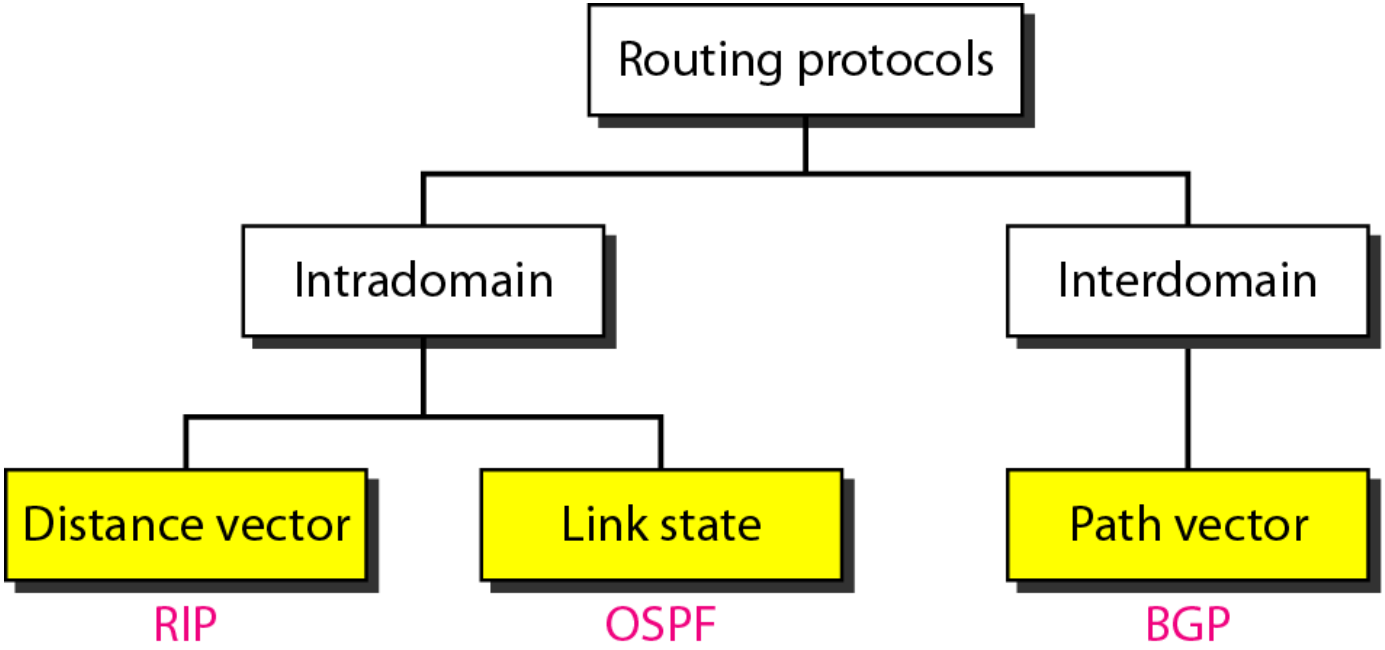
InterAS Routing



IntraAS Routing



Routing Protocols



- Interior Routing Protocols
  - Routing within ASs
- Exterior Routing Protocols
  - Routing between ASs

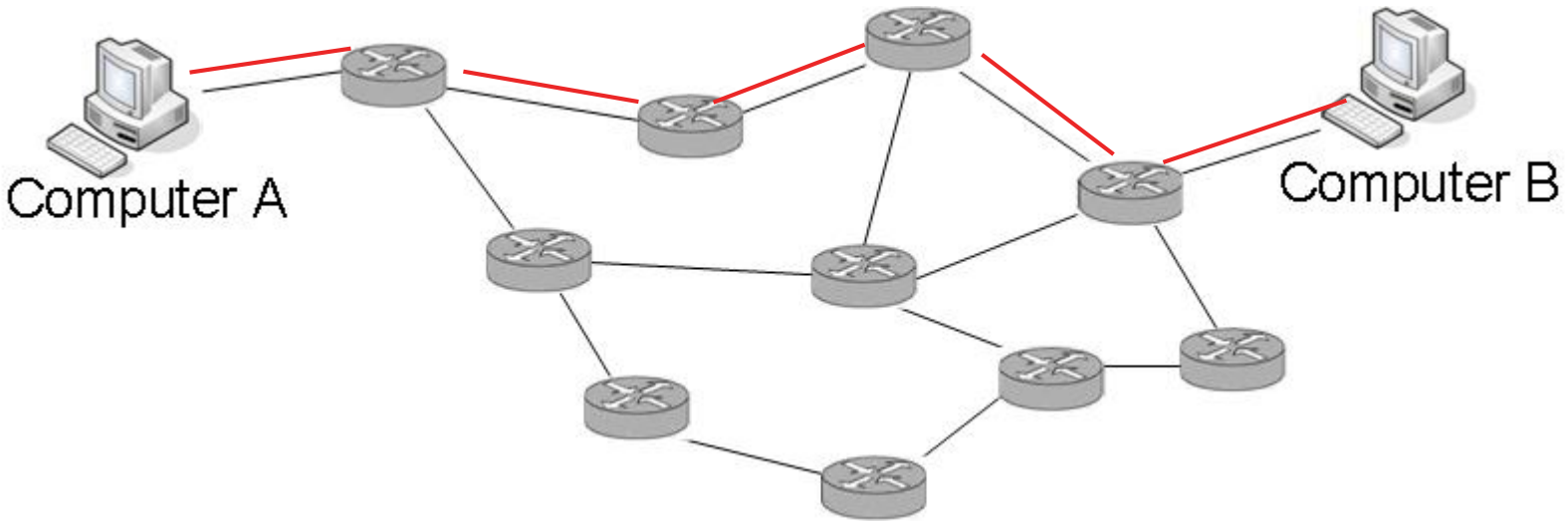
* Figure is courtesy of B. Forouzan

Routers

- One Main Interest
  Forwarding Packets
- Important Aspects
  Queue Length
  Routing Table

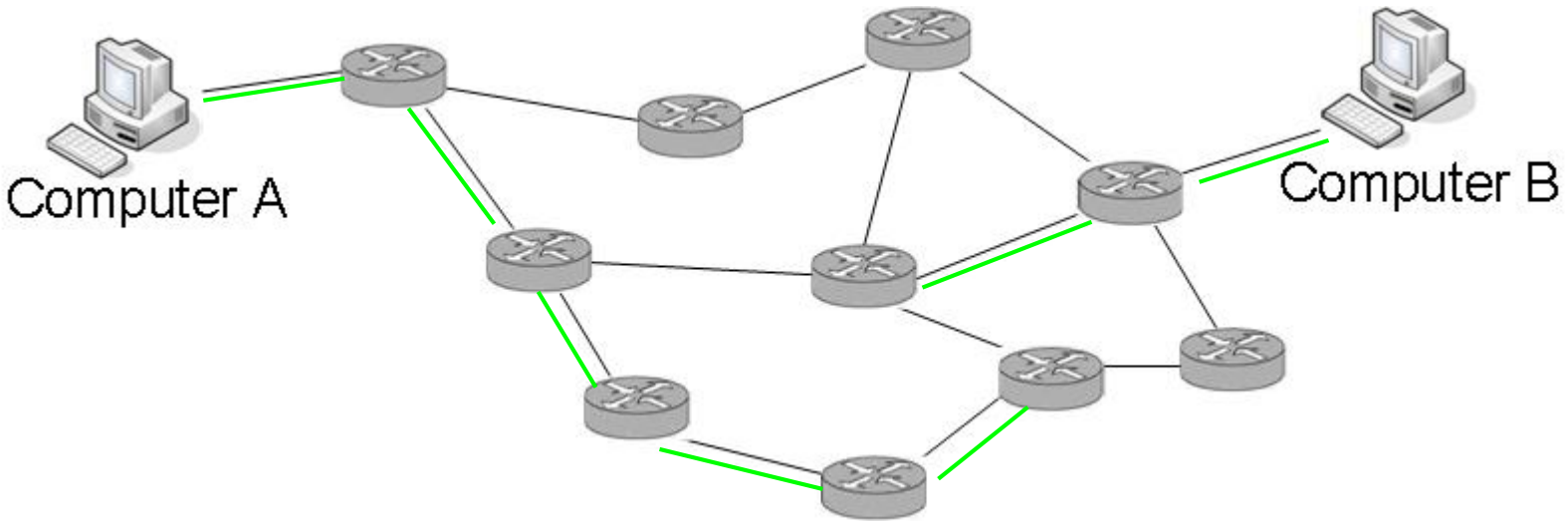
Destination	Gateway	Interface
IP Range$_1$ IP Range$_2$	G_1 G_2	IF_1 IF_2

The Scenario



- Best route: Smallest number of hops?

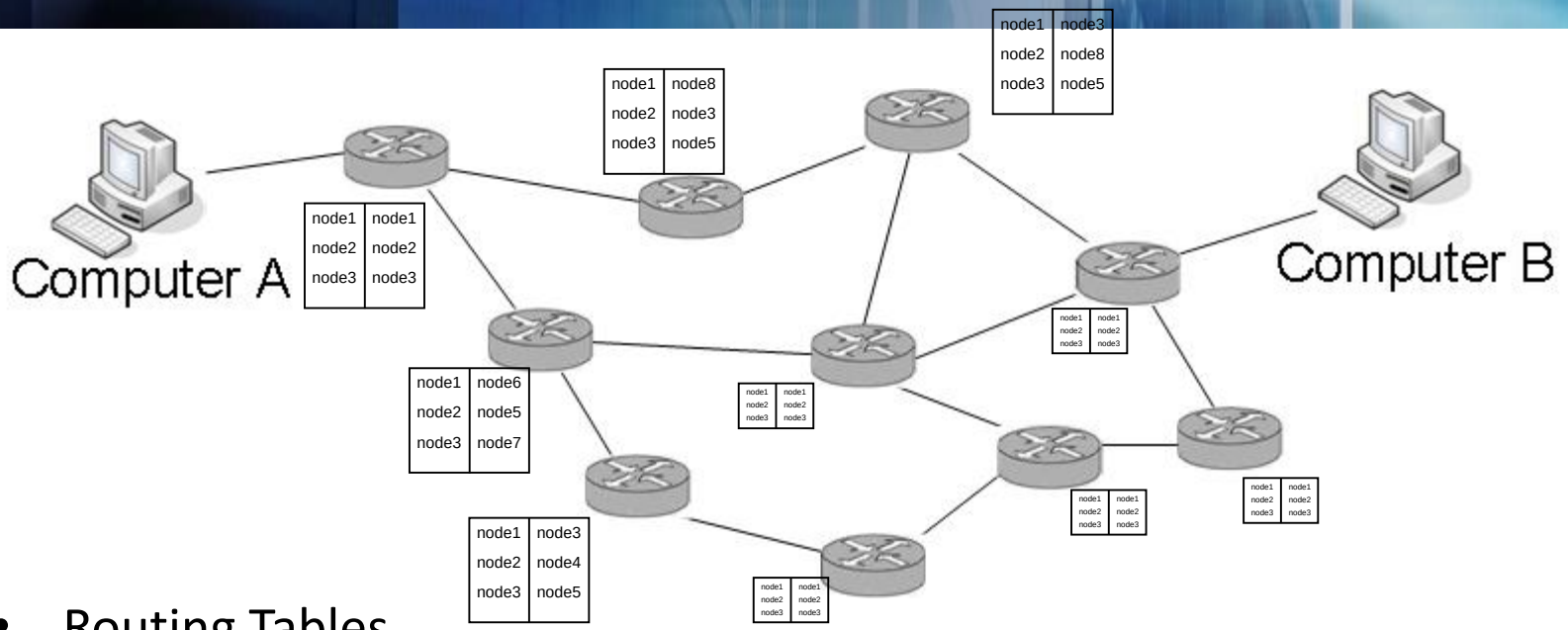
The Scenario



- Best route:

  Fastest round-trip time?

  Highest Bandwidth?



- Routing Tables
  - Creating tables
    - Dynamic vs. Static
  - Maintaining tables
    - Periodic vs. Aperiodic

Routing Approaches

- Distance Vector routing
  - Routes propagate through exchange of routing tables
  - Based on communication with neighbours

- Link State routing
  - Establishing view of complete topology
  - Makes use of Dijkstra's Shortest-Path Algorithm

Distance Vector Routing

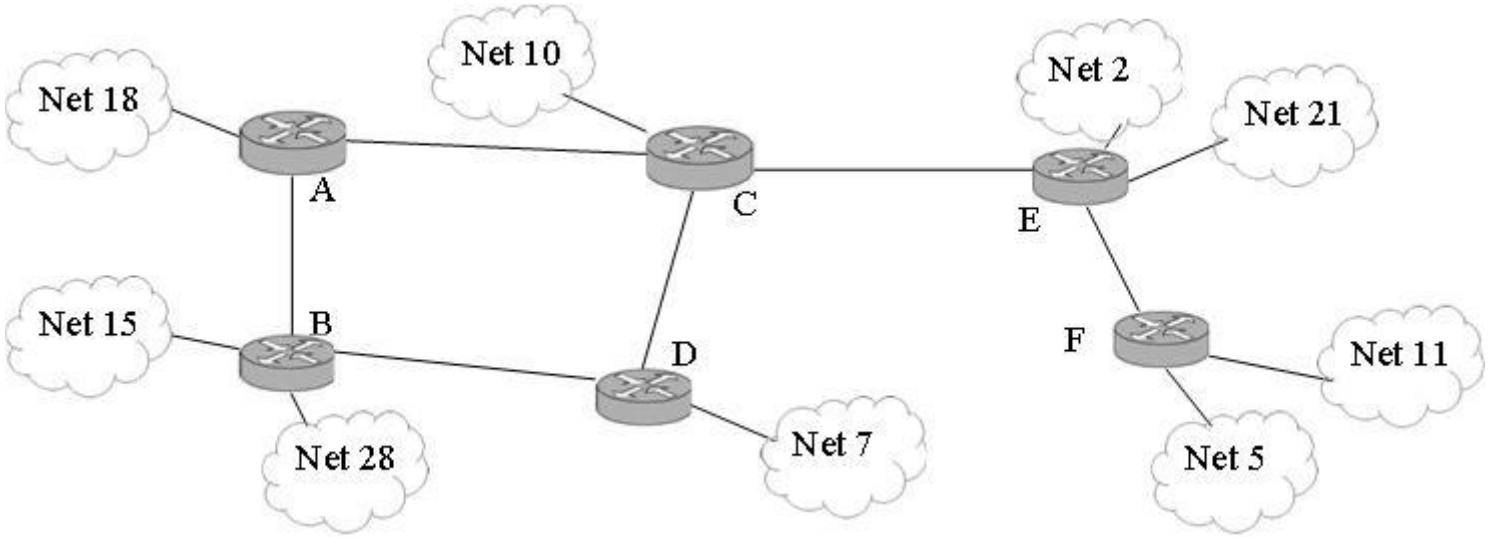
Destination	Hop Count	Next Router	Other info
163.5.0.0	7	172.6.23.4	
197.5.13.0	5	176.3.6.17	
189.45.0.0	4	200.5.1.6	
115.0.0.0	6	131.4.7.19	

- Each node maintains a set of triples
  - **`(Destination, Cost, NextHop)`**

- Exchange updates with directly connected neighbors
  - periodically (on the order of several seconds)
  - whenever table changes (called *triggered* update)

- Each update is a list of all pairs in the routing table:
  - **`(Destination, Cost)`**

Distance Vector: Example



A

Net18	1	-

B

Net15	1	-
Net28	1	-

C

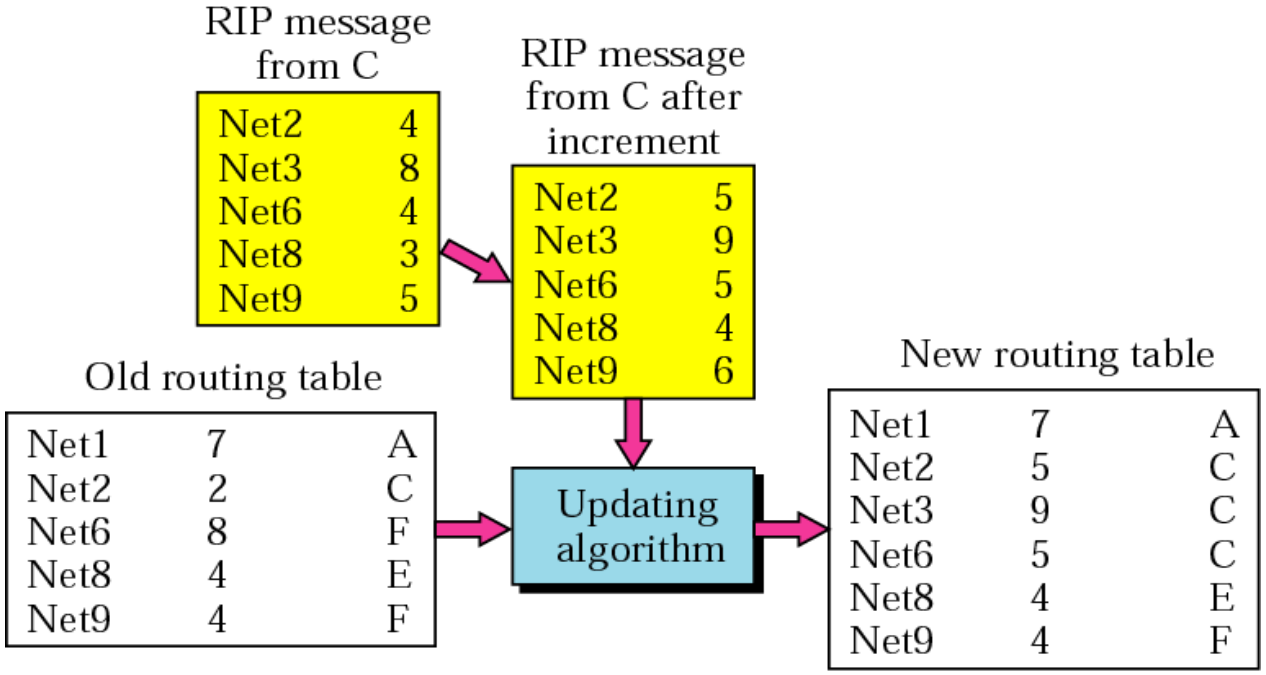
Net10	1	-

D

Net7	1	-

- Each router know the networks that are immediately connected to it

Receiving an Update



Net1: No news, do not change
Net2: Same next hop, replace
Net3: A new router, add
Net6: Different next hop, new hop count smaller, replace
Net8: Different next hop, new hop count the same, do not change
Net9: Different next hop, new hop count larger, do not change

* Figure is courtesy of B. Forouzan

RIP Updating Algorithm

Receive: A response RIP message

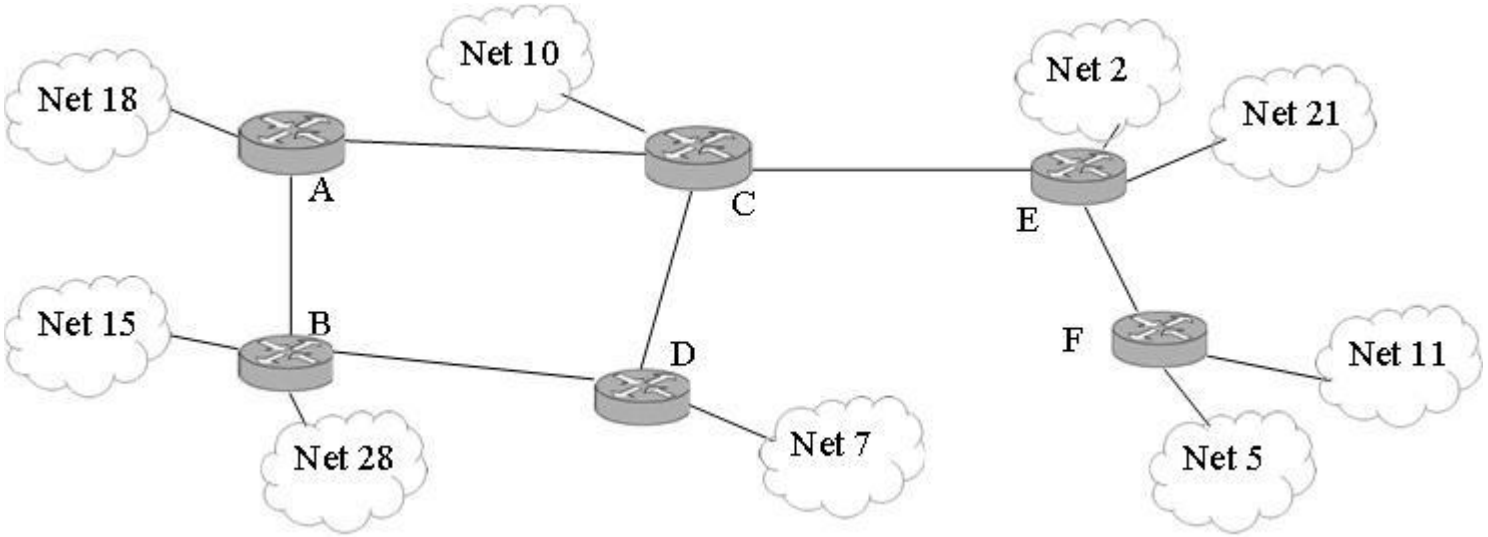
- Add one hop to the hop count for each advertised destination.

- Repeat the following steps for each advertised destination:

a) If (destination not in the routing table)
   Add the advertised information to the table.

b) Else
   If (next-hop field is the same)
      Replace entry in the table with the advertised one.
   Else
      If (advertised hop count smaller than one in the table)
         Replace entry in the routing table.

Distance Vector: Example II



A

Net18	1	-
Net15	2	B
Net28	2	B
Net10	2	C

B

Net15	1	-
Net28	1	-
Net18	2	A
Net7	2	D

C

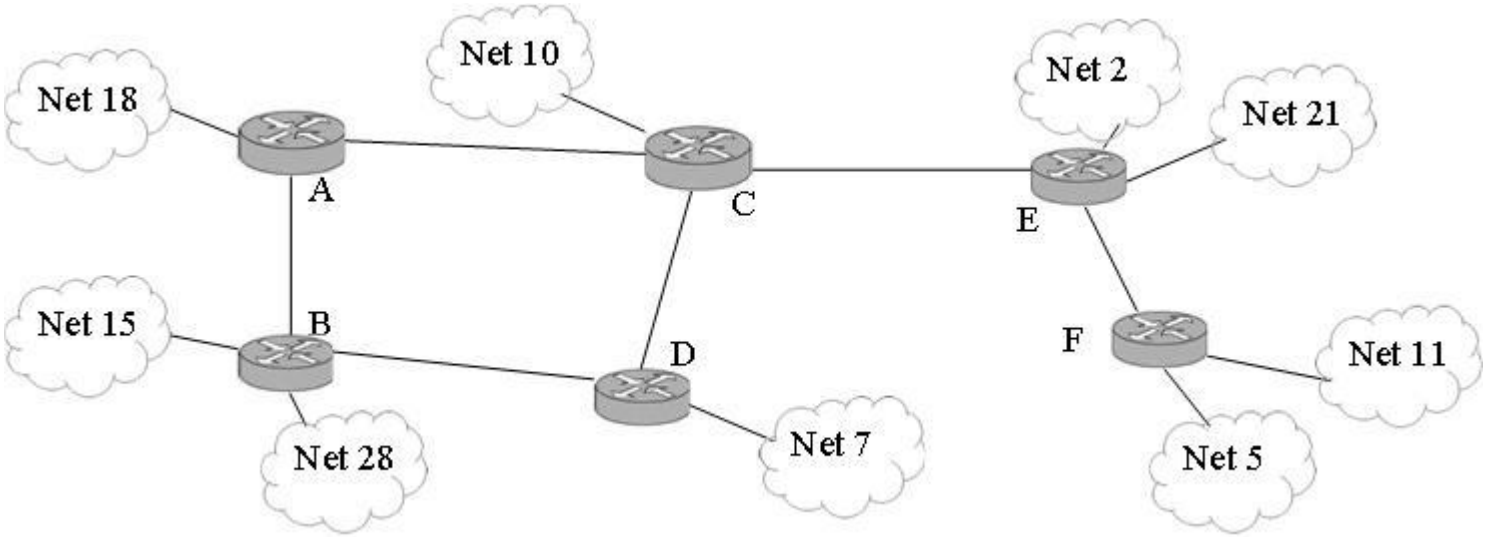
Net10	1	-
Net7	2	D
Net2	2	E
Net21	2	E

D

Net7	1	-
Net15	2	B
Net28	2	B
Net10	2	C

- e.g. Node A incorporates information from Node B

Distance Vector: Example III



A

Net18	1	-
Net15	2	B
Net28	2	B
Net10	2	C
Net7	3	B

B

Net15	1	-
Net28	1	-
Net18	2	A
Net7	2	D
Net10	3	D

C

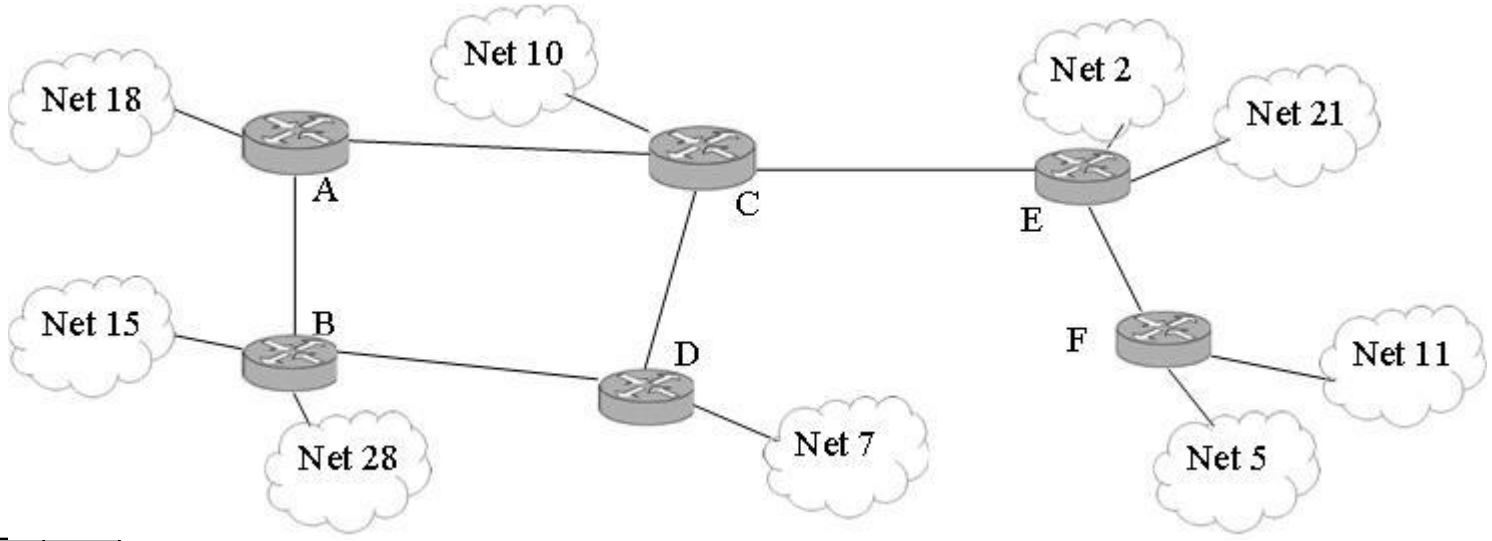
Net10	1	-
Net7	2	D
Net2	2	E
Net21	2	E
Net11	3	E
Net5	3	E

D

Net7	1	-
Net15	2	B
Net28	2	B
Net10	2	C
Net2	3	C
Net21	3	C

- e.g. Node A receives information Node B received during the last round

Distance Vector: Example IV

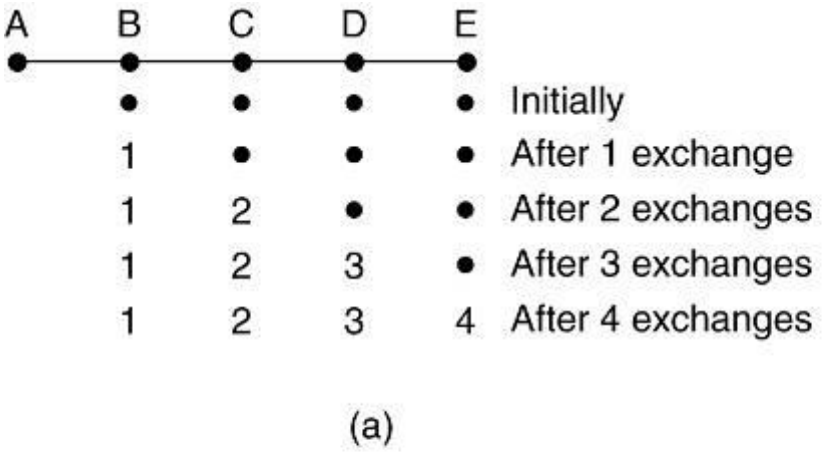


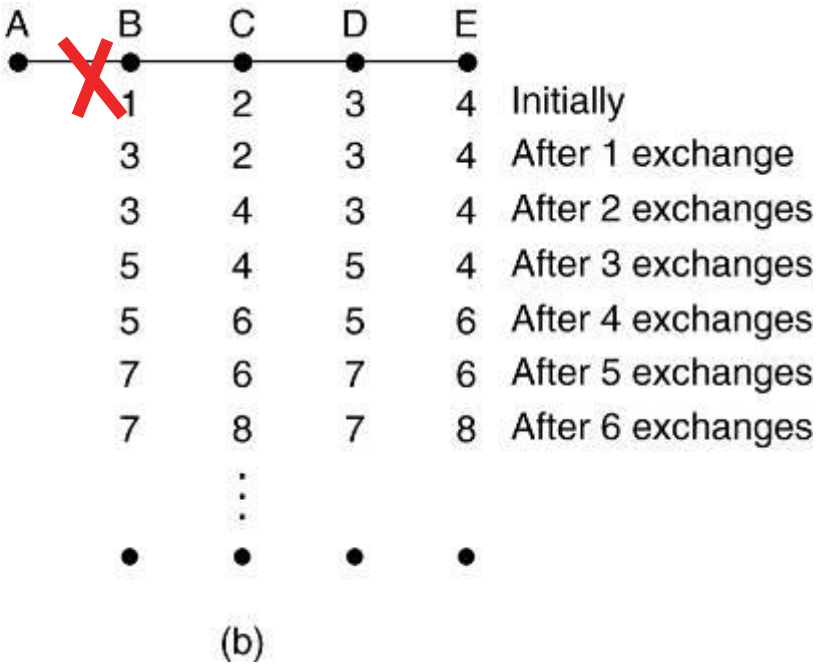
A

Net18	1	-
Net15	2	B
Net28	2	B
Net10	2	C
Net7	3	B
…	…	…
Net11	5	C
Net5	5	C

- Time require to build complete routing tables for all nodes
  - this is known as convergence

Count-to-Infinity Problem





- Routers exchange updates
- After 4 steps the network converges
- Every router knows how to get to router A

- Link between A and B breaks
- B receives update from C advertising route to A

* Figure is courtesy of A. Tanenbaum

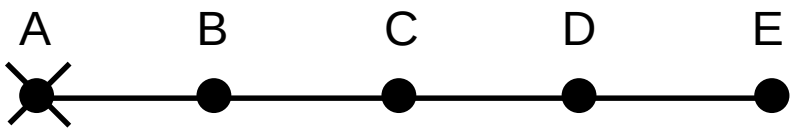
Solutions for Count-to-Infinity

- Possible solutions:
  1. Impose upper bound on maximum distance
  2. *Split horizon*
     - C should not send to node B its new distance to node A, if node B is C's next-hop towards A
  3. *Split horizon with poisoned reverse*
     - C should tell node B that its distance to node A is ∞, when node B is C's next-hop towards A

- Unfortunately, none of these solutions can deal with arbitrary topology cycles

Split Horizon with Poisoned Reverse

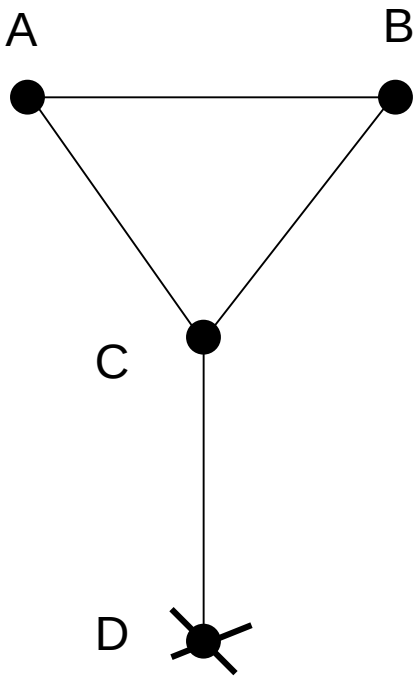
Report "split-horizon" routes as infinity to break loops on the first routing exchange.



inf.	2	3	4	B learns A is dead
inf.	inf. → 2	3	4	B reports to C that A's metric is inf.
inf.	inf.	3	4	After 1 exchange
inf.	inf.	inf.	4	After 2 exchanges
inf.	inf.	inf.	inf.	After 3 exchanges

* Taken from X. Li, Rutgers

Split Horizon Failure

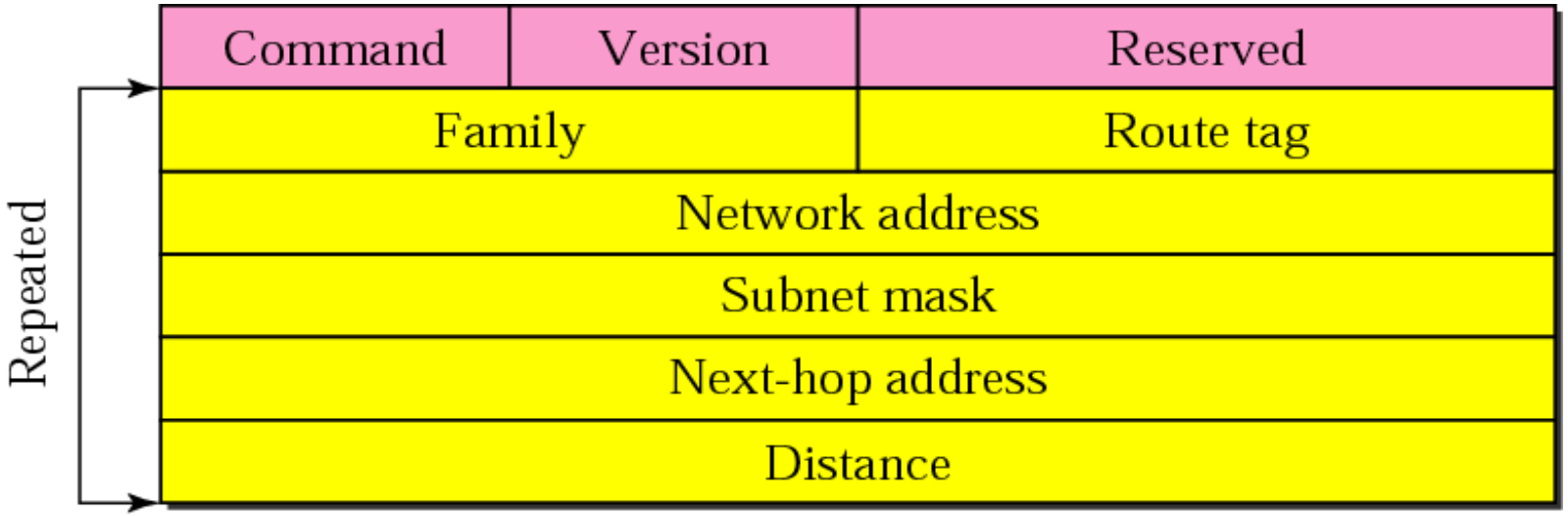


If D goes down, A and B will still count to infinity.

Split-Horizon infinity messages are sent from A->C and B->C, not A<->B

* Taken from X. Li, Rutgers

RIPv2 Message Format



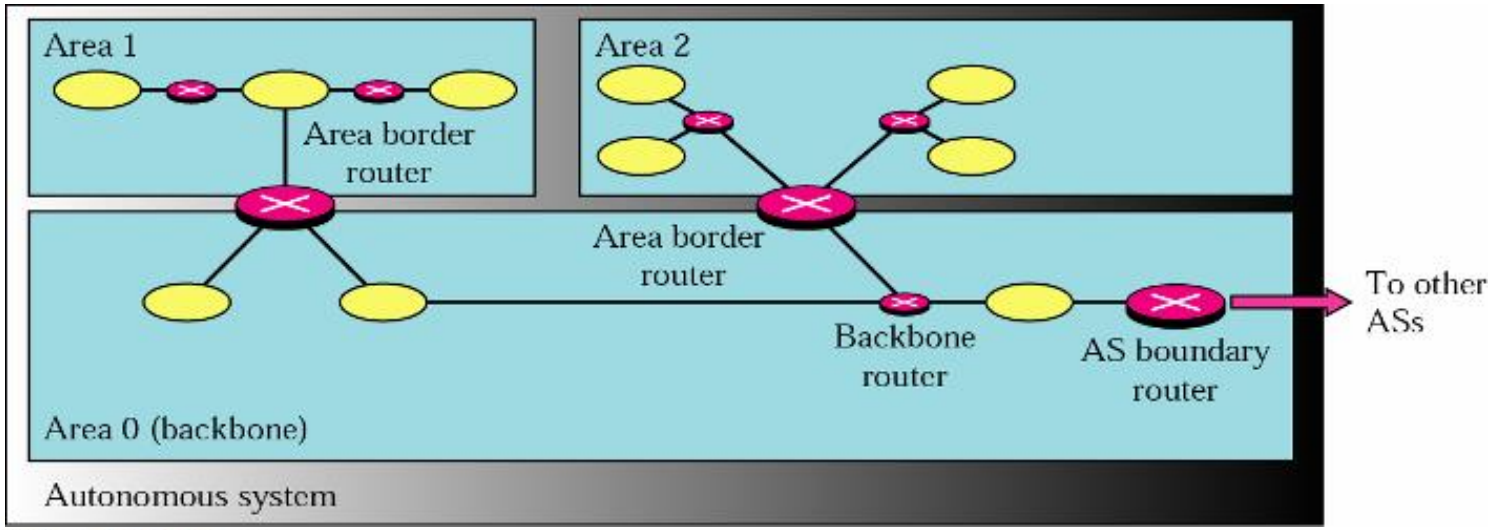
* Figure is courtesy of B. Forouzan

Link State Routing

- Sharing knowledge about neighbourhood
- Sharing with every other router

- Uses Dijkstra's Shortest-Path Algorithm to calculate the routing table

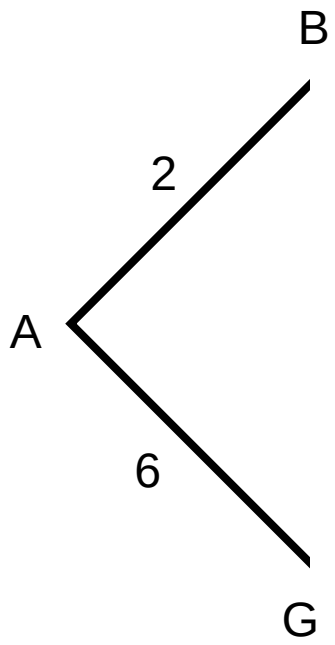
- Open Shortest Path First (OSPF)
  - Divides autonomous system into areas
  - Border routers summarize information for an area



* Figure is courtesy of B. Forouzan

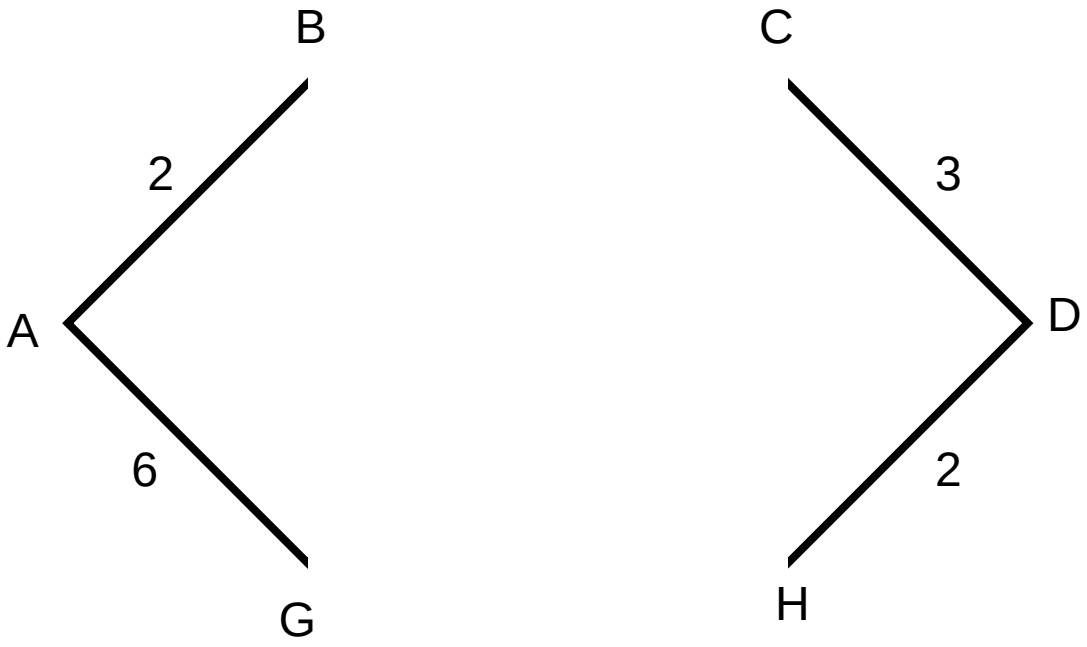
Example: Building a network graph

A discovers 2 neighbours B and G in distance 2 and 6 respectively



Example: Building a network graph

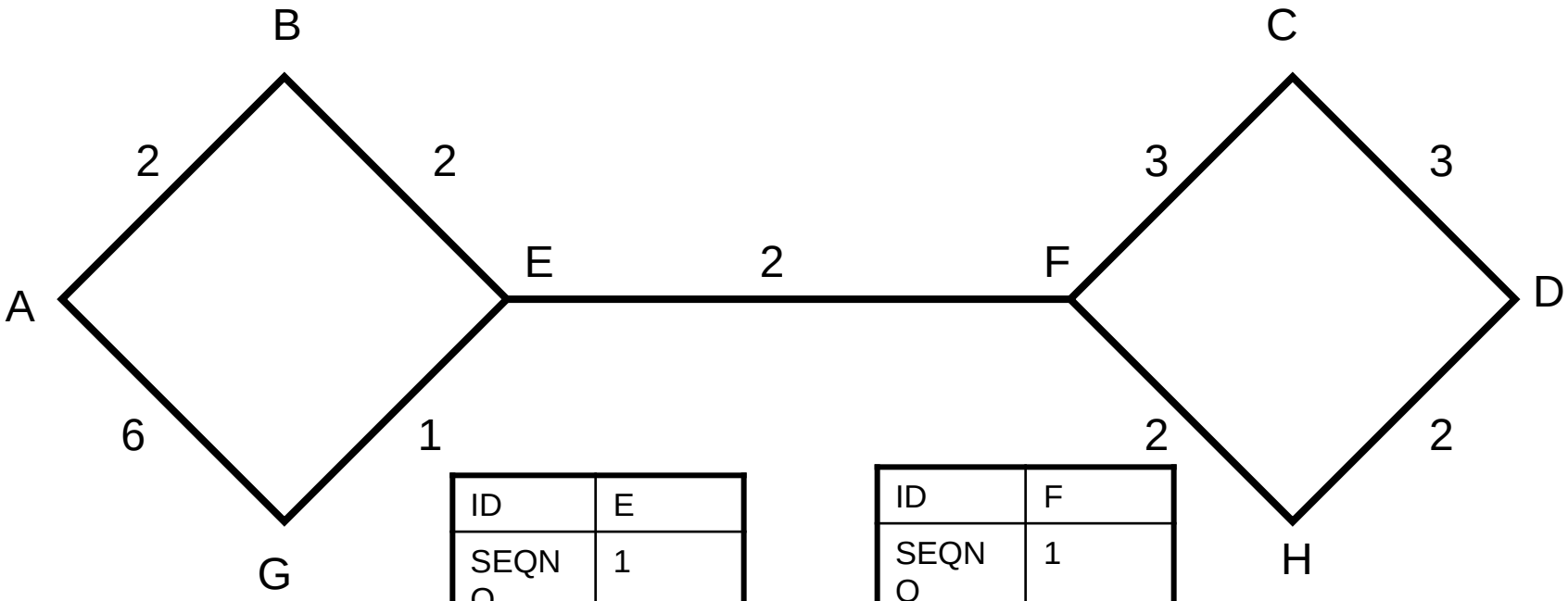
A receives an update from D



ID	D
SEQNO	0
TTL	21
C	3
H	2

Example: Building a network graph

A receives updates from E and F



ID	E
SEQNO	1
TTL	22
B	2
G	1
F	2

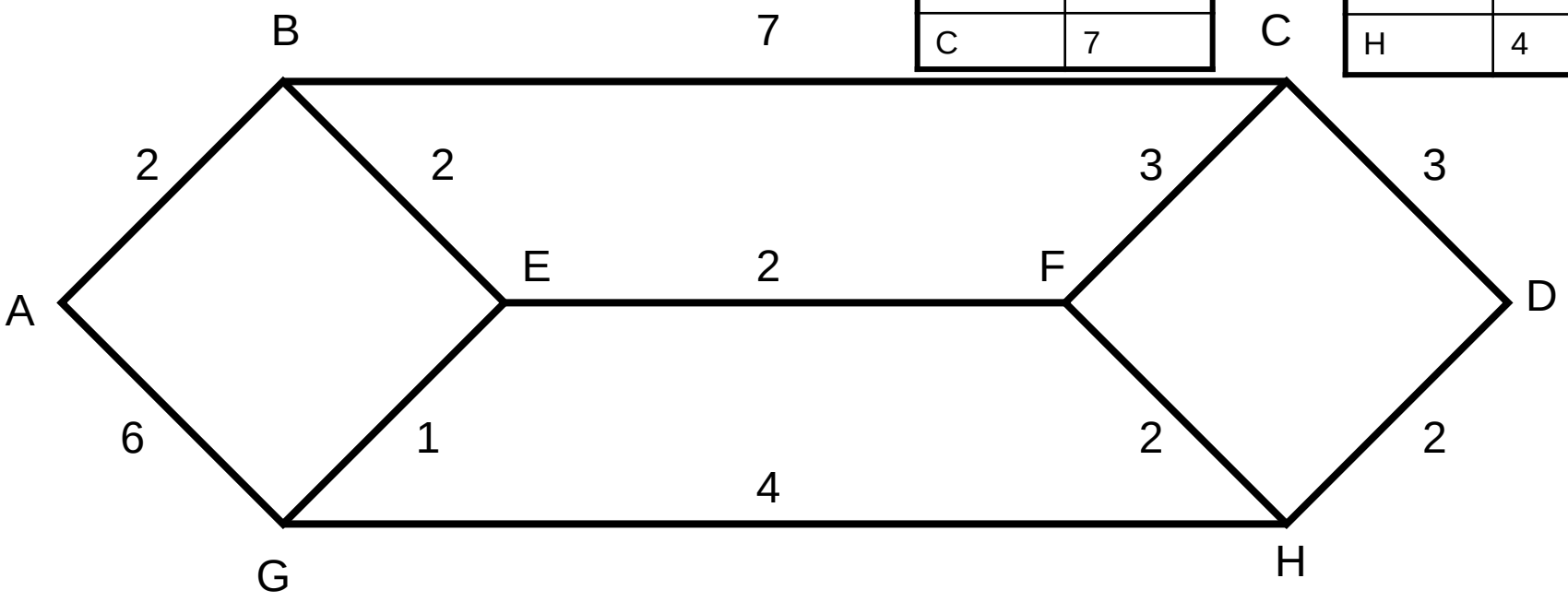
ID	F
SEQNO	1
TTL	21
C	3
H	2
E	2

Example: Building a network graph

A receives updates from B and G

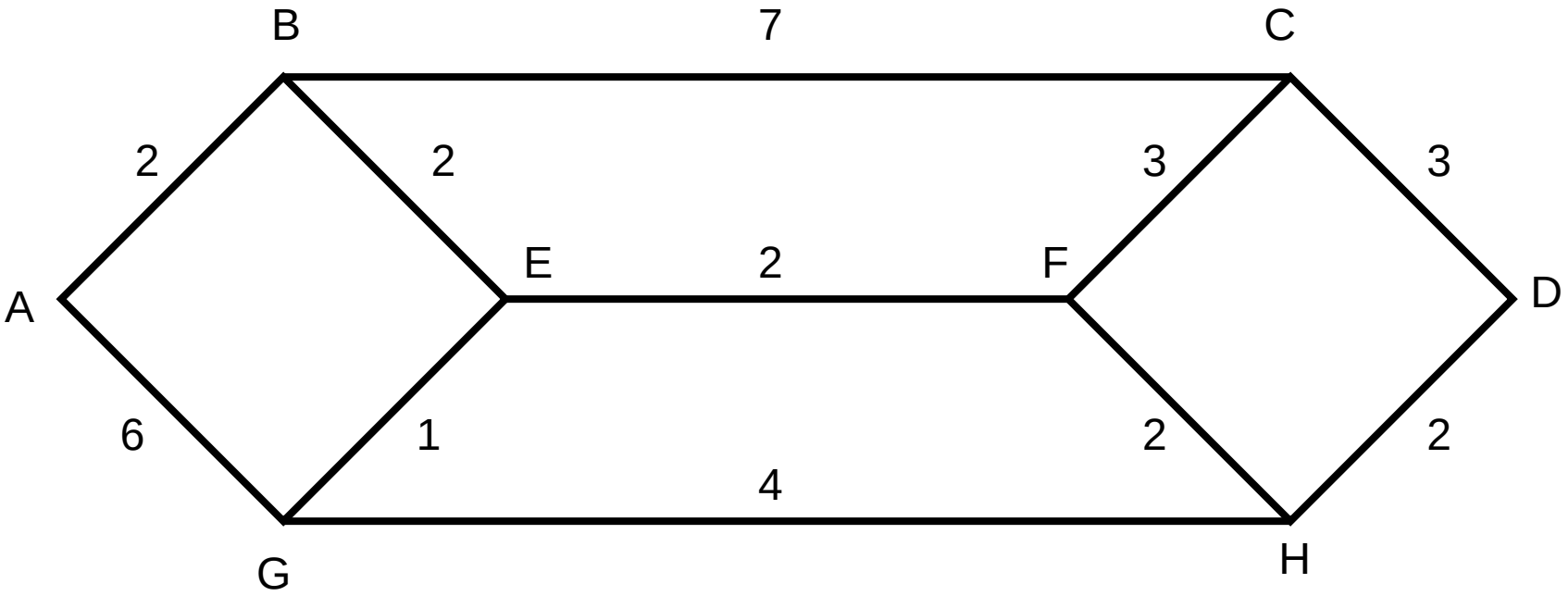
ID	B
SEQNO	2
TTL	23
A	2
E	2
C	7

ID	G
SEQNO	3
TTL	21
A	6
E	1
H	4



Example: Building a network graph

Final graph



Dijkstra's Algorithm

Let

- *N* denotes set of nodes in the graph
- *l* (*i*, *j*) denotes non-negative cost (weight) for edge (*i*, *j*)
- *s* denotes initial node (source)
- *M* denotes the set of nodes incorporated so far
- *C*(*n*) denotes cost of the path from *s* to node *n*

```
M = {s}
for each n in N - {s}
   C(n) = l(s, n)
while (N != M)
   M = M ∪ {w} such that C(w) is the minimum for
        all w in (N - M)
   for each n in (N - M)
      C(n) = MIN(C(n), C (w) + l(w, n ))
```

Dijkstra's Algorithm

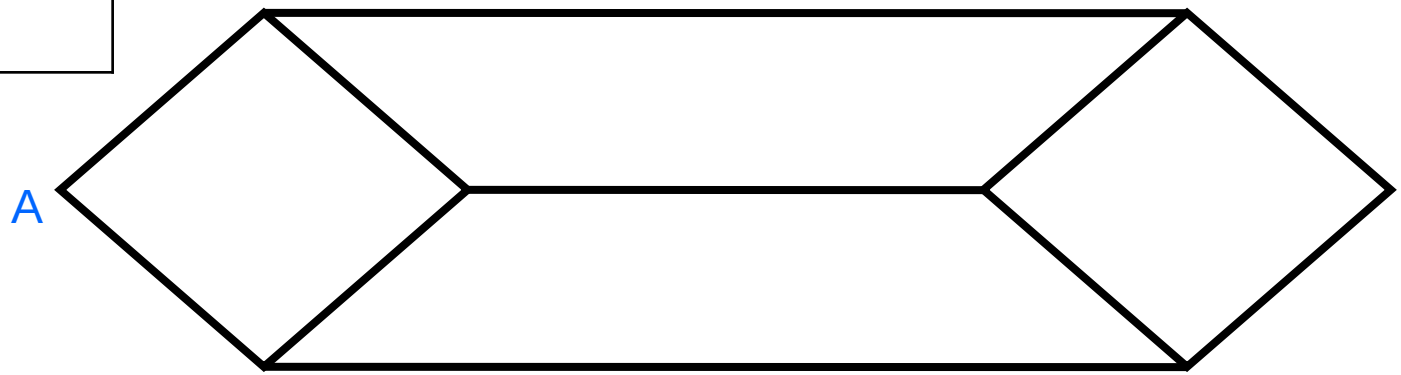
1. Start with the local node (router): the root of the tree.
2. Assign a cost of 0 to this node and make it the first permanent node.
3. Examine each neighbour node of the node that was the last permanent node.
4. Assign a cumulative cost to each node and make it tentative.
5. Among the list of tentative nodes
   a) Find the node with the smallest cumulative cost and make it permanent.
   b) If a node can be reached from more than one direction
      I. Select the direction with the shortest cumulative cost.
6. Repeat steps 3 to 5 until every node becomes permanent.

Example: Dijkstra's Algorithm

Routing Table

Tentative Nodes

A	0	-



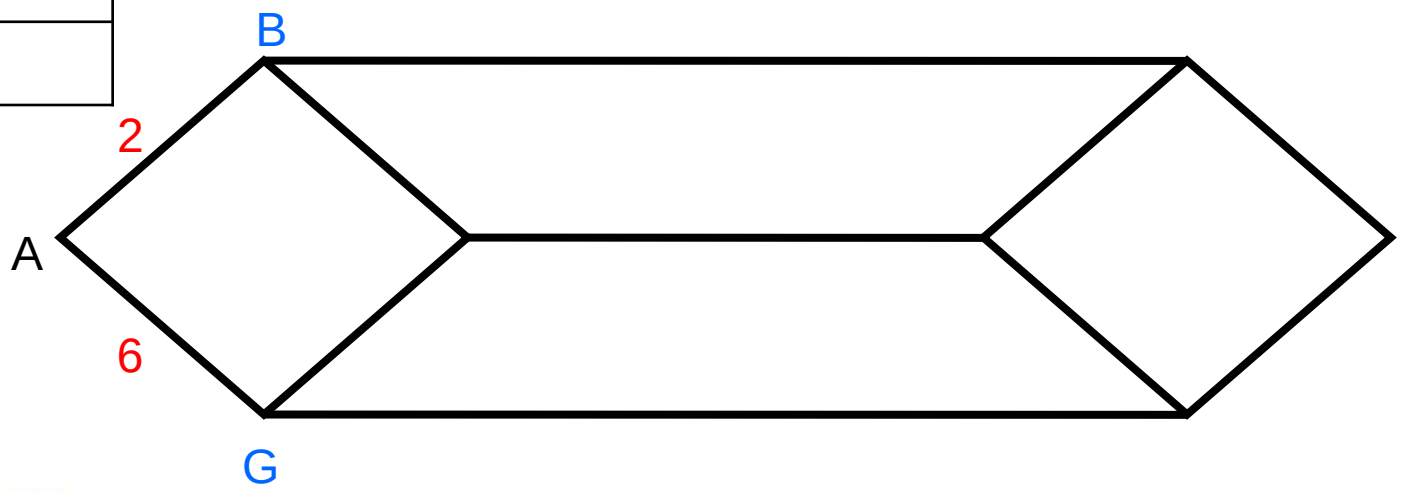
Example: Dijkstra's Algorithm

Routing Table

A	0	-

Tentative Nodes

B	2	A
G	6	A



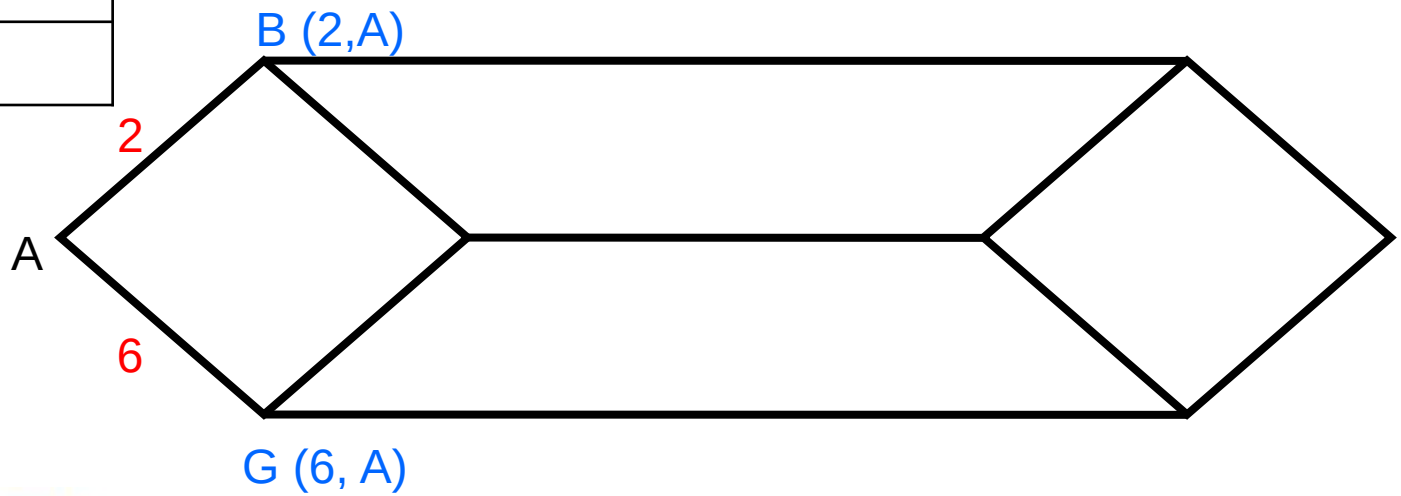
Example: Dijkstra's Algorithm

Routing Table

A	0	-
B	2	A

Tentative Nodes

G	6	A



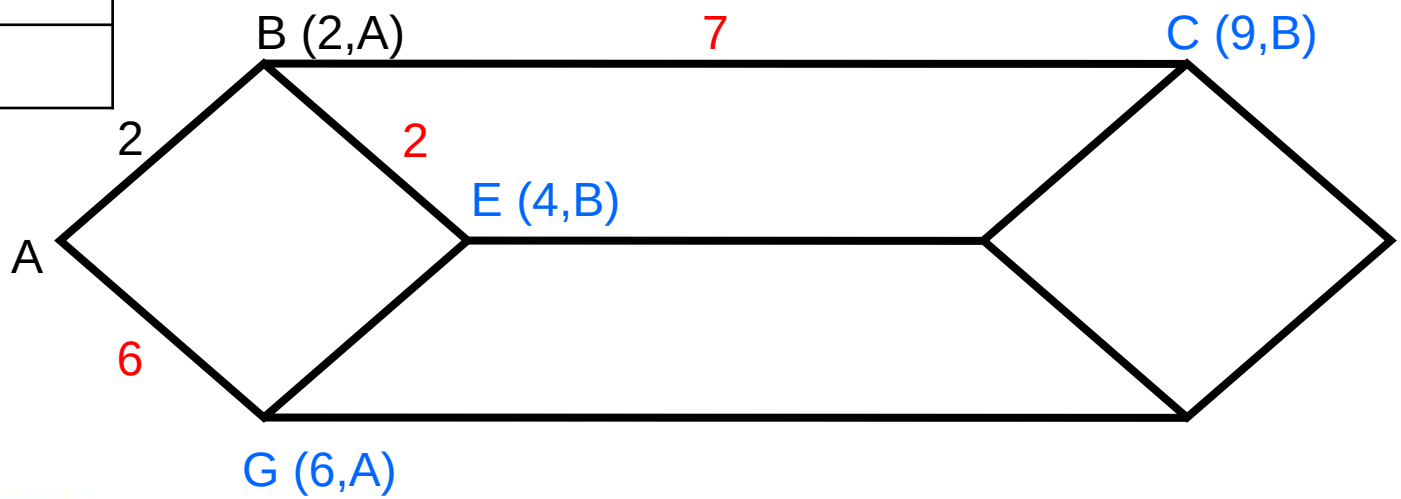
Example: Dijkstra's Algorithm

Routing Table

A	0	-
B	2	A

Tentative Nodes

G	6	A
E	4	B
C	9	B



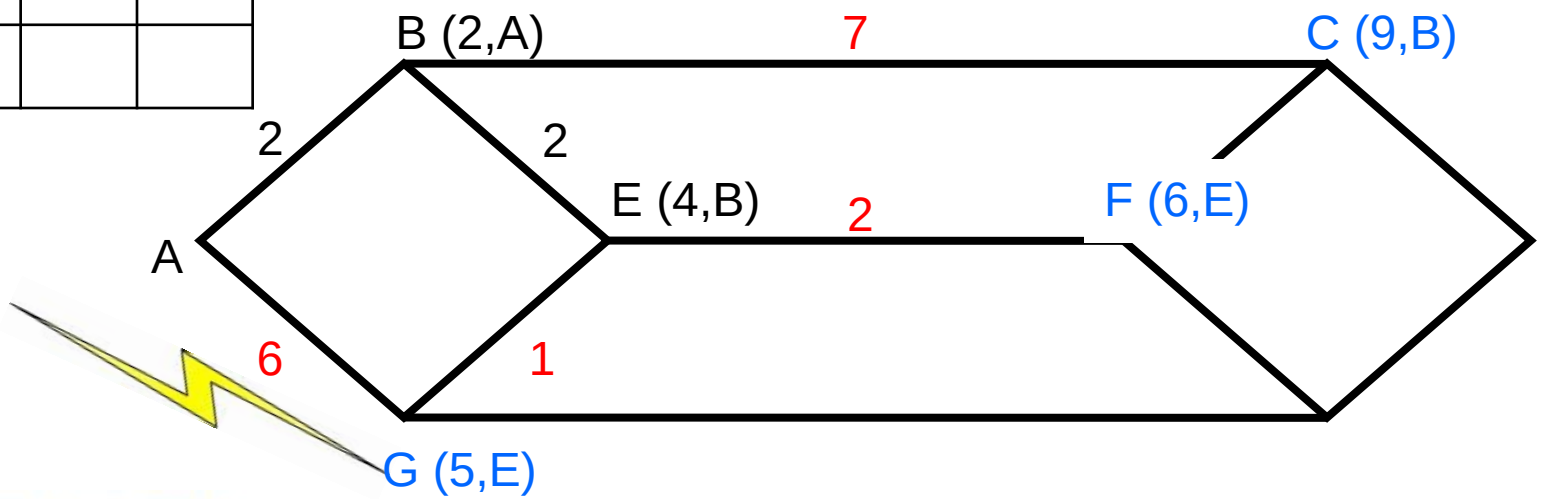
Example: Dijkstra's Algorithm

Routing Table

A	0	-
B	2	A
E	4	B

Tentative Nodes

~~G~~	~~6~~	~~A~~
C	9	B
F	6	E
G	5	E



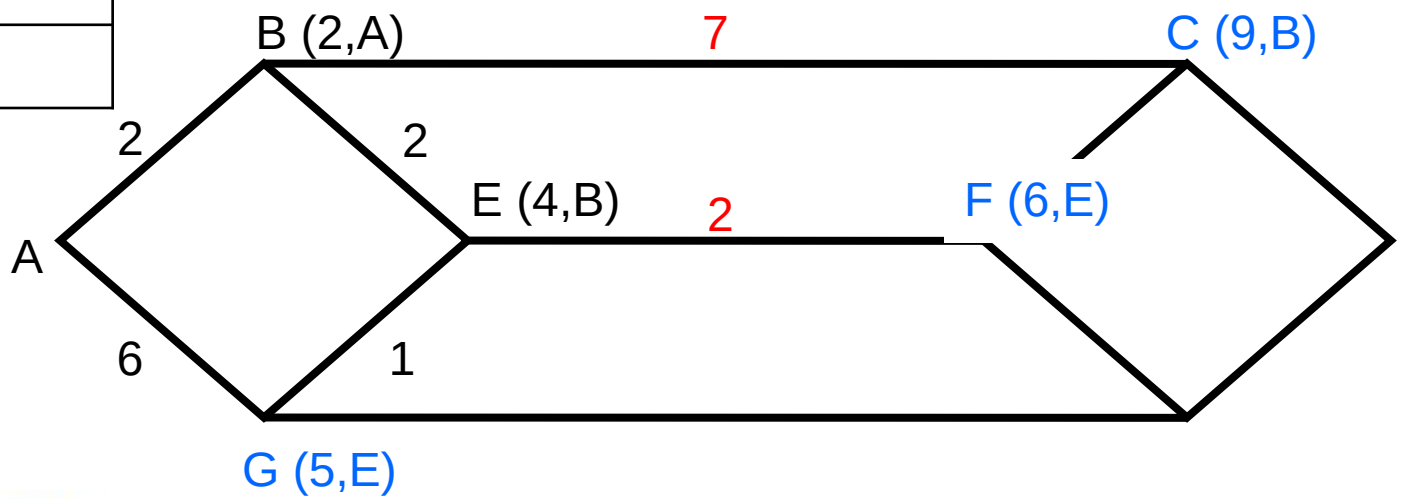
Example: Dijkstra's Algorithm

Routing Table

A	0	-
B	2	A
E	4	B
G	5	E

Tentative Nodes

C	9	B
F	6	E



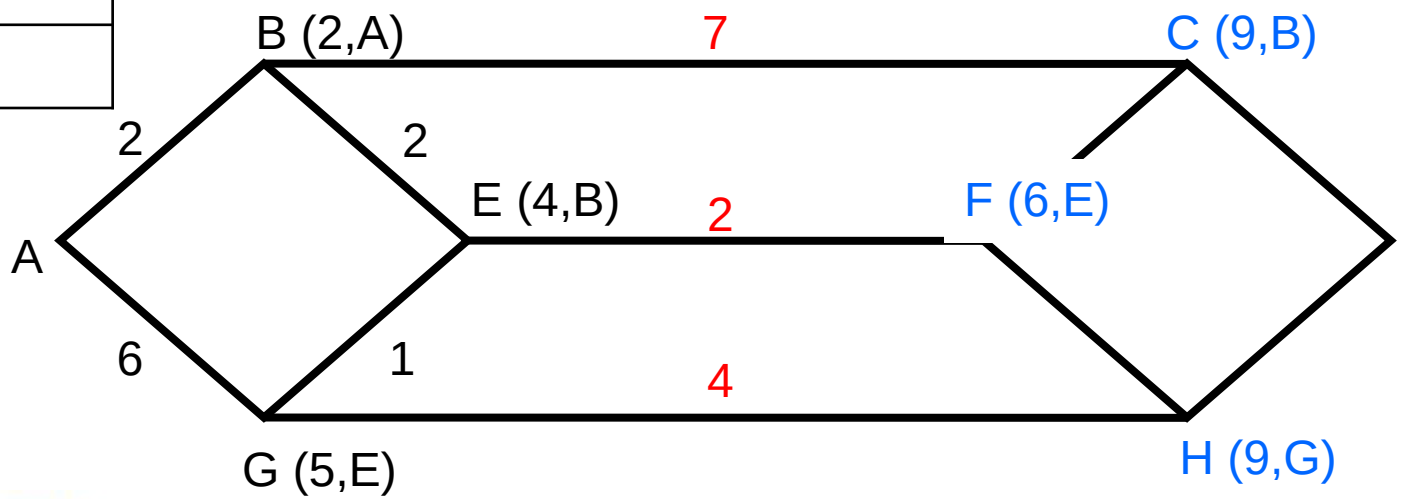
Example: Dijkstra's Algorithm

Routing Table

A	0	-
B	2	A
E	4	B
G	5	E

Tentative Nodes

C	9	B
H	9	G
F	6	E



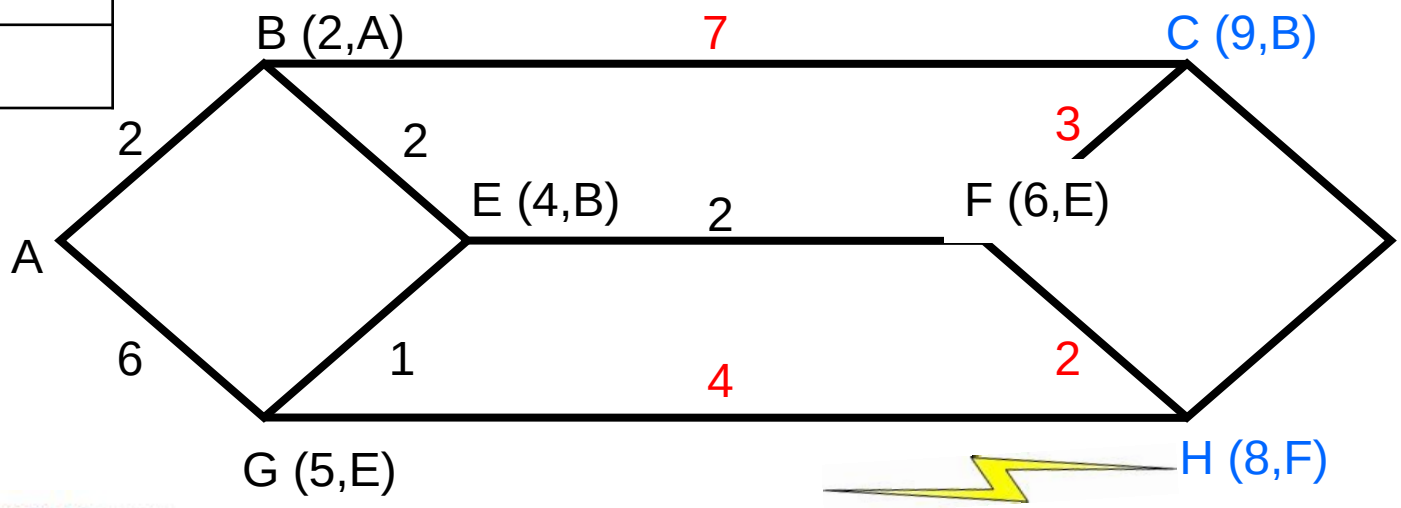
Example: Dijkstra's Algorithm

Routing Table

A	0	-
B	2	A
E	4	B
G	5	E
F	6	E

Tentative Nodes

C	9	B
~~H~~	~~9~~	~~G~~
H	8	F



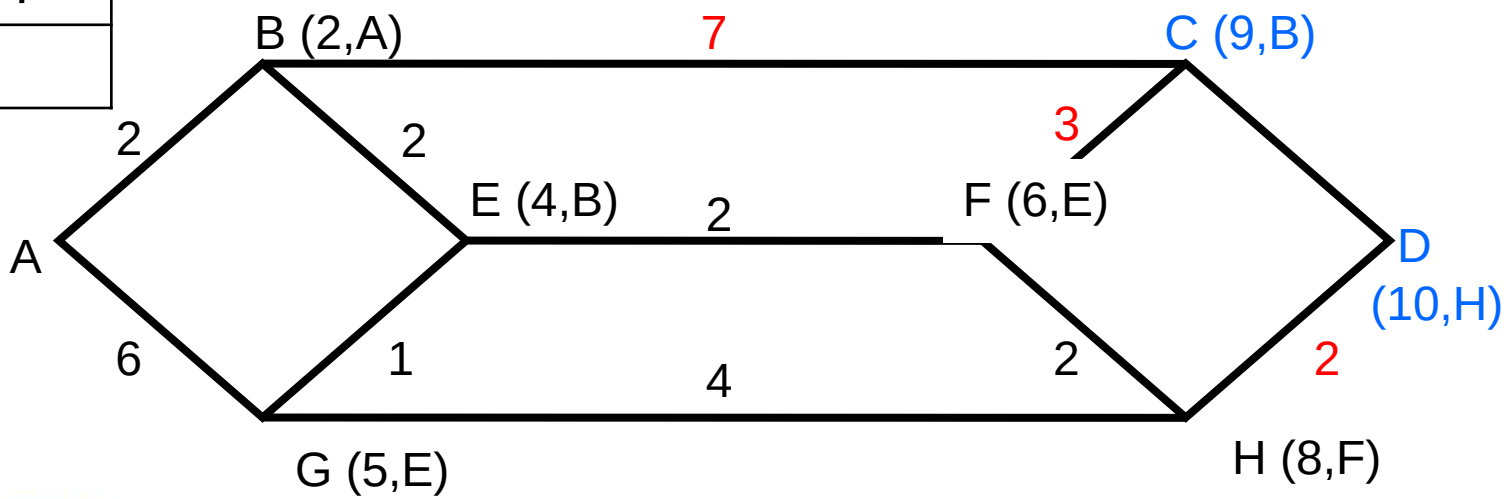
Example: Dijkstra's Algorithm

Routing Table

A	0	-
B	2	A
E	4	B
G	5	E
F	6	E
H	8	F

Tentative Nodes

C	9	B
D	10	H



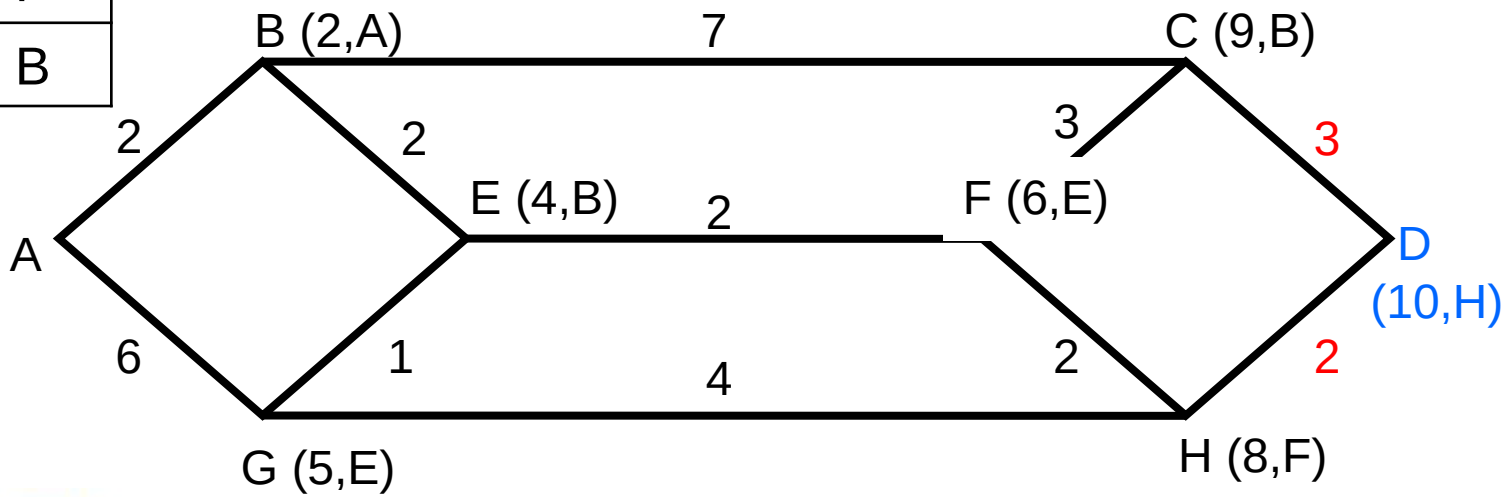
Example: Dijkstra's Algorithm

Routing Table

A	0	-
B	2	A
E	4	B
G	5	E
F	6	E
H	8	F
C	9	B

Tentative Nodes

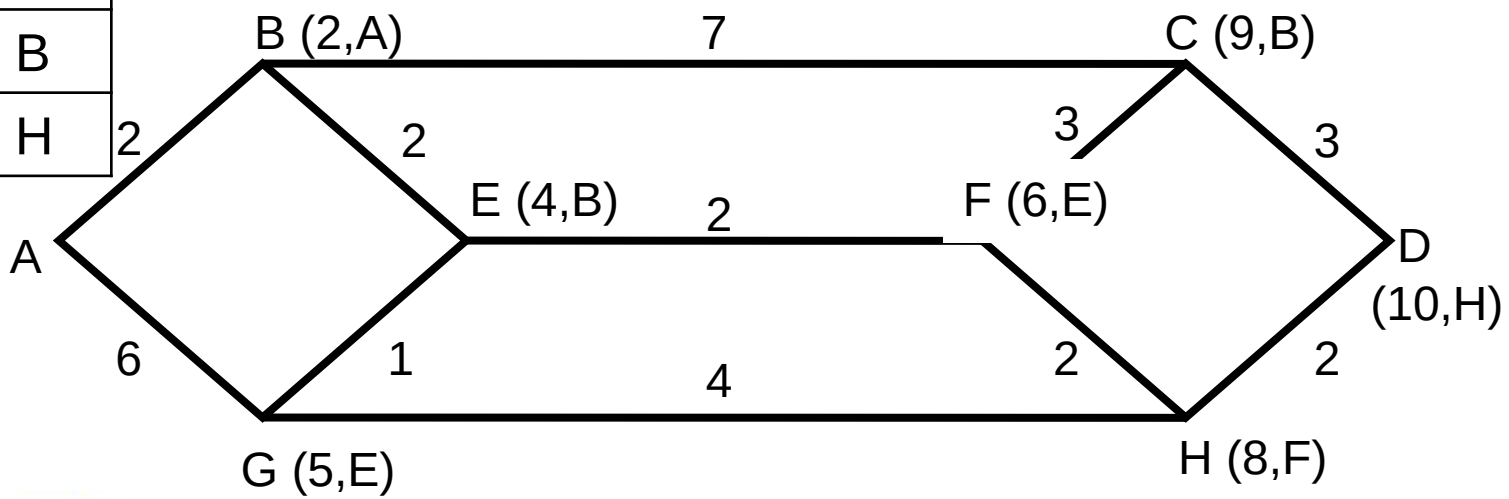
~~D~~	~~12~~	C
D	10	H



Example: Dijkstra's Algorithm

Routing Table

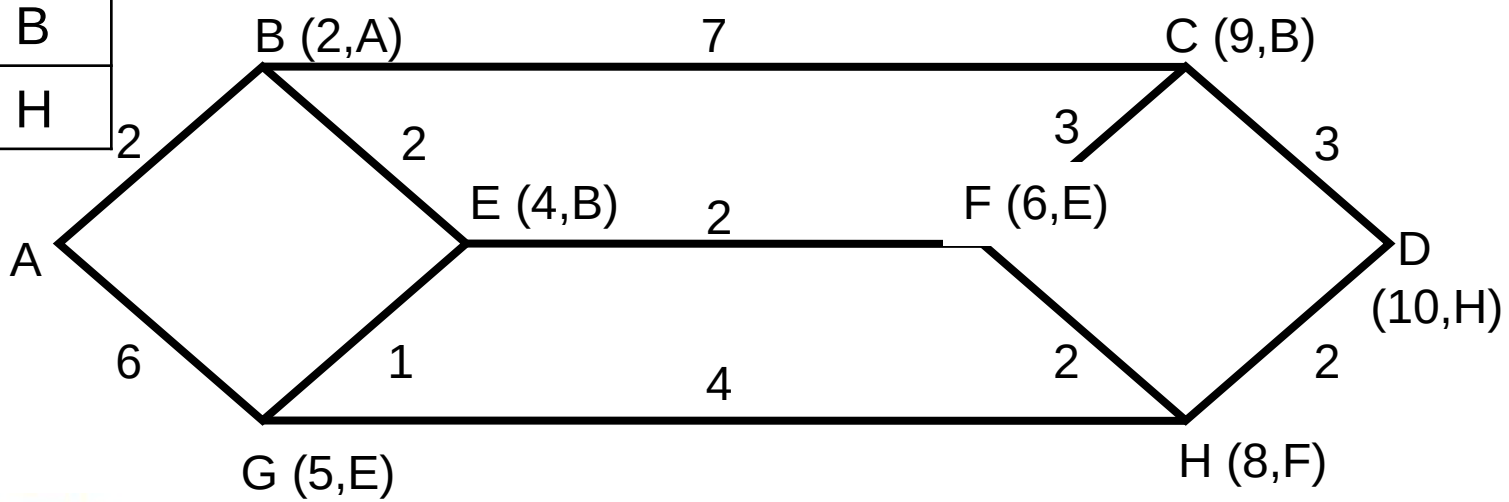
A	0	-
B	2	A
E	4	B
G	5	E
F	6	E
H	8	F
C	9	B
D	10	H



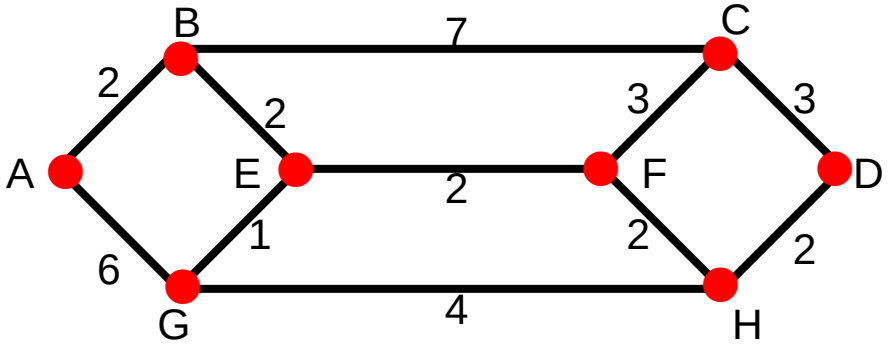
Example: Dijkstra's Algorithm

A	0	-
B	2	A
E	4	B
G	5	E
F	6	E
H	8	F
C	9	B
D	10	H

- Shortest connection from A to any other node
- Greedy algorithm! Follows minima!



Informal Description of Shortest Path Algo.



origin = starting point (e.g. point A)

destination = point that has a distance from the origin associated with itself (e.g. C, distance 9)

neighbour = destination that has a direct connection to another destination (e.g. B's neighbours A, E and C)

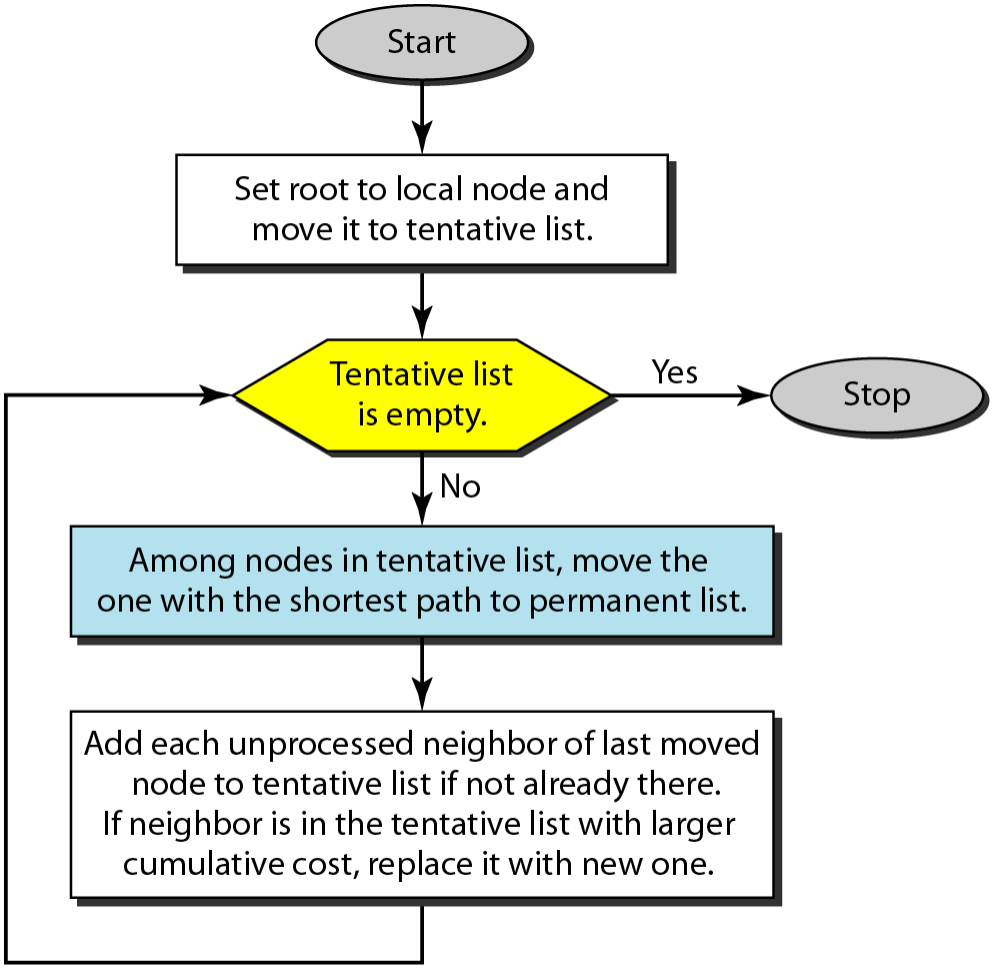
table = table of destinations with a known distance from the origin

bag = bag of destinations to be resolved

Informal Description of Shortest Path Algo.

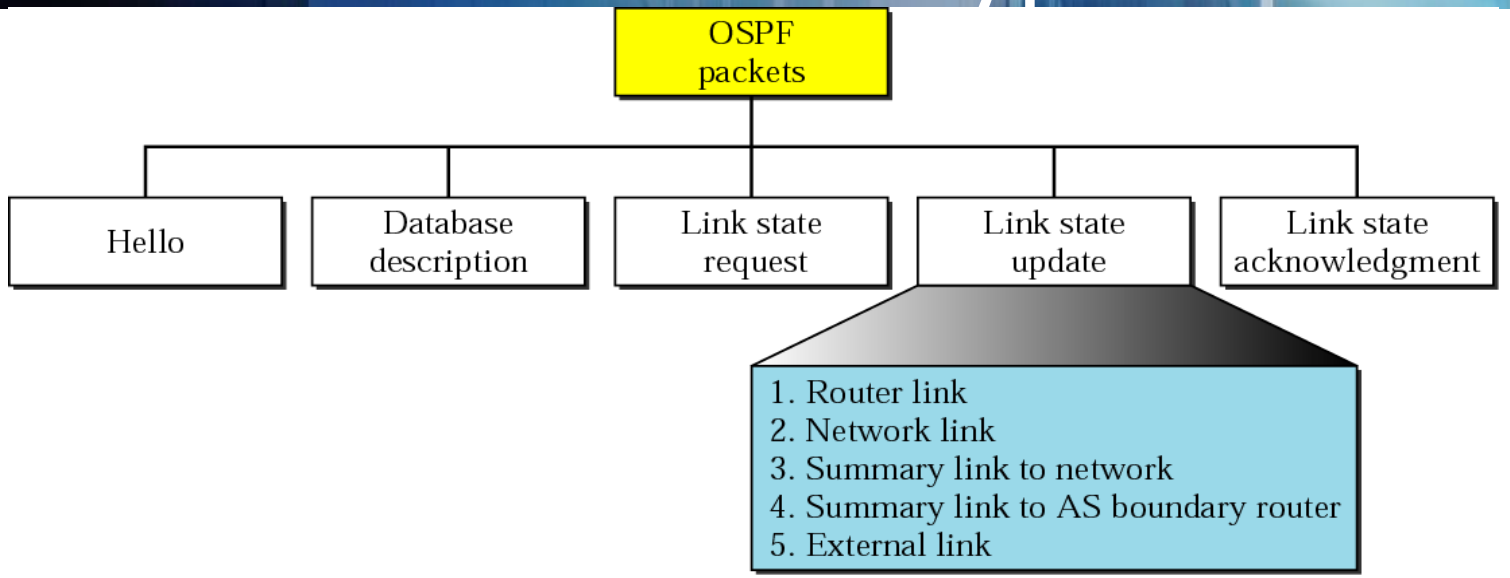
- Table and bag are empty at the beginning
- Put the origin as a destination with the distance 0 into the bag

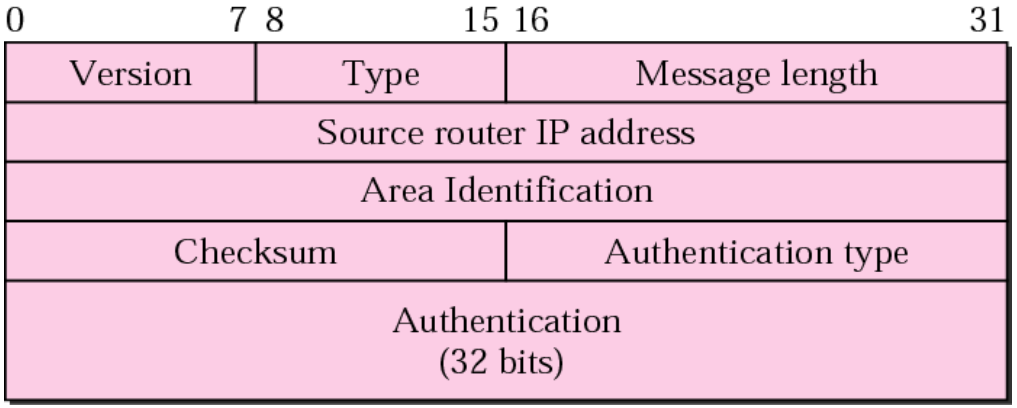
- As long as there are destinations in the bag
  - Take the destination with the smallest distance out of the bag and put it into the table (if the distance is smaller than an already existing entry for the same destination)
  - Put all the neighbours of that destination with their distance to the origin (=distance to the destination + distance from the destination to the neighbour) into the bag



* Figure is courtesy of B. Forouzan

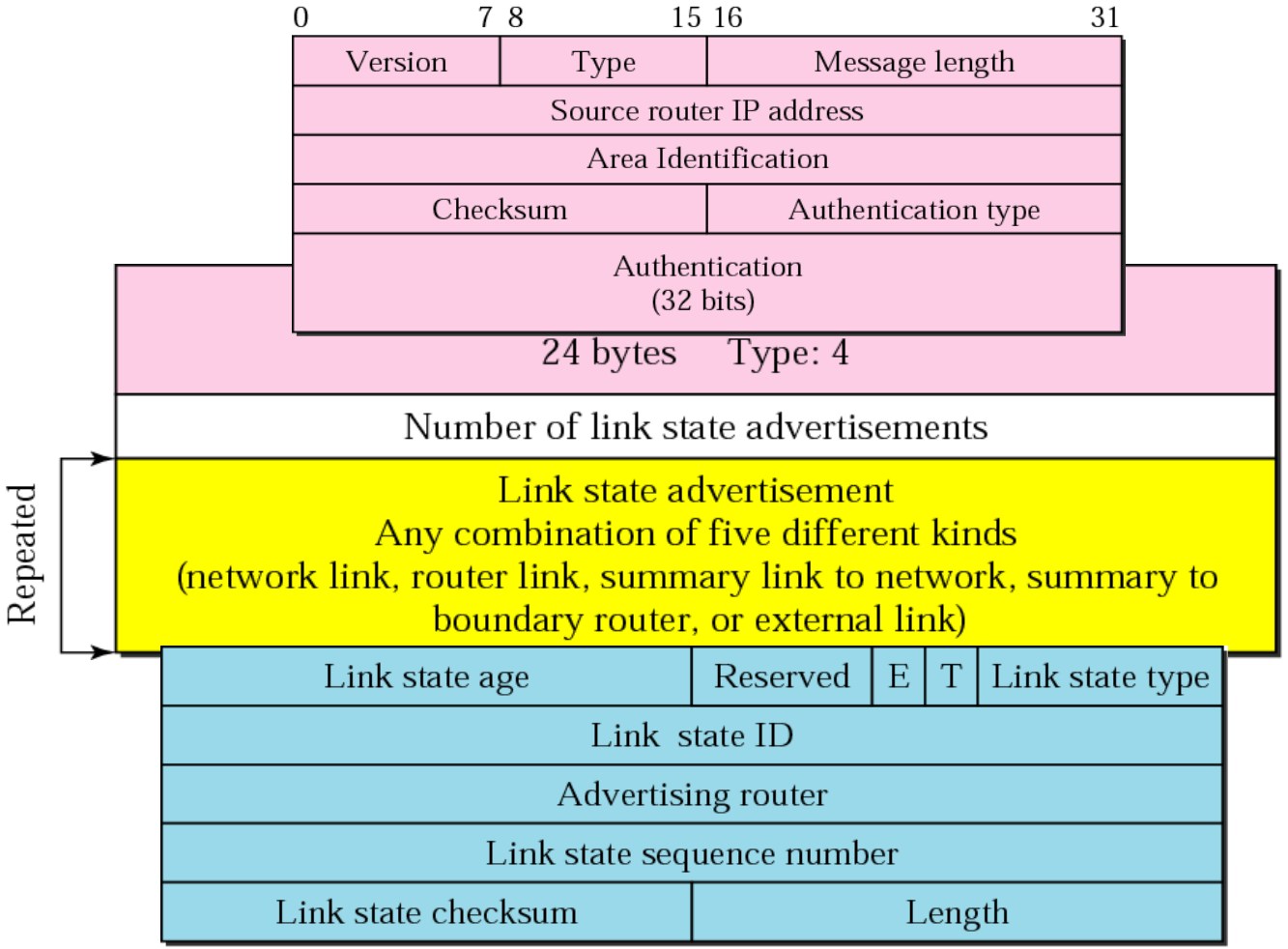
OSPF Packet Types





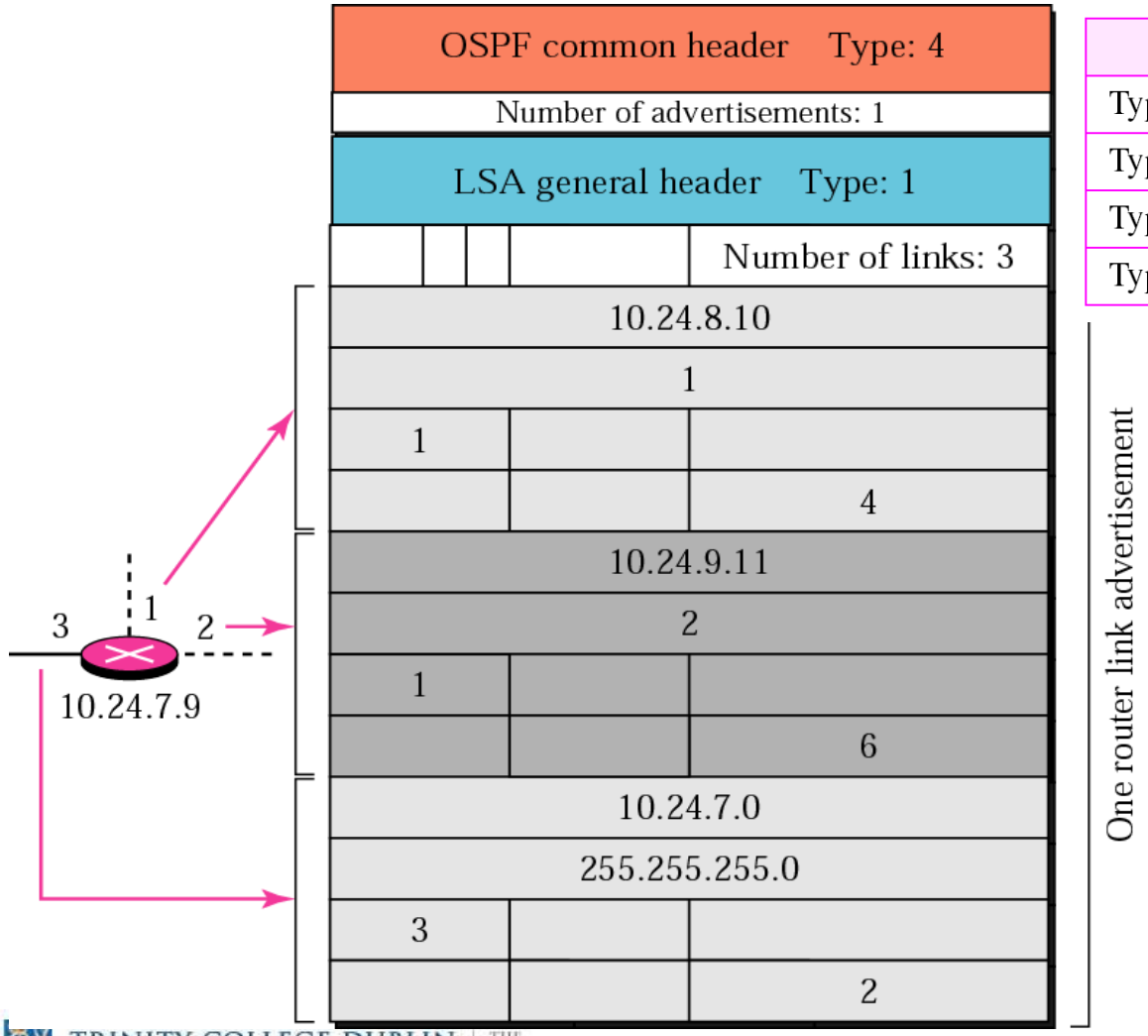
* Figure is courtesy of B. Forouzan

OSPF Link State Advertisement



* Figure is courtesy of B. Forouzan

OSPF LSA Example

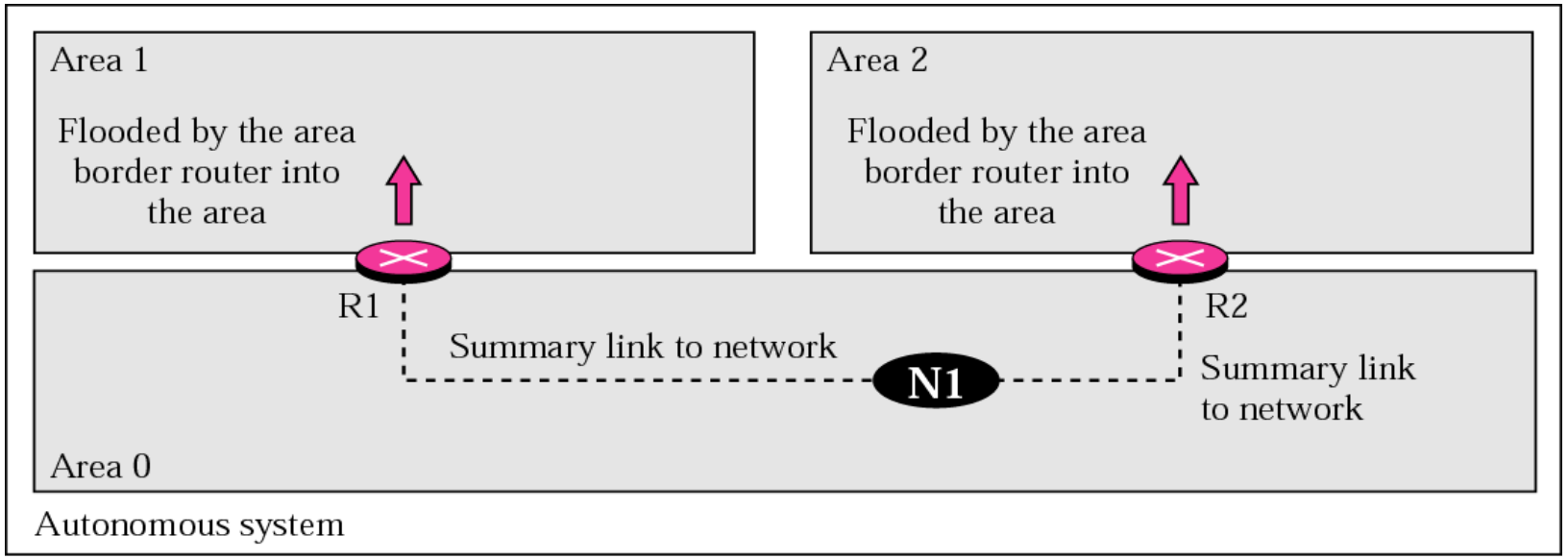


Link Type	*Link Identification*
Type 1: Point-to-point	Address of neighbor router
Type 2: Transient	Address of designated router
Type 3: Stub	Network address
Type 4: Virtual	Address of neighbor router

Link Type	*Link Data*
Type 1: Point-to-point	Interface number
Type 2: Transient	Router address
Type 3: Stub	Network mask
Type 4: Virtual	Router address

* Figure is courtesy of B. Forouzan

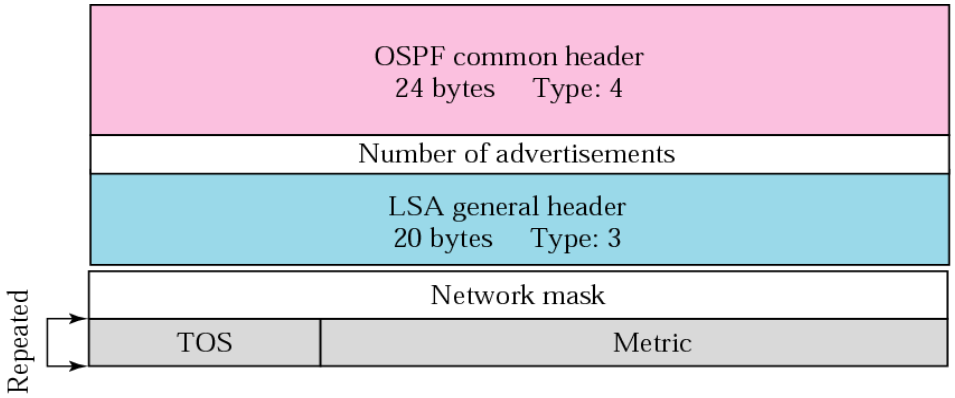
Areas in OSPF

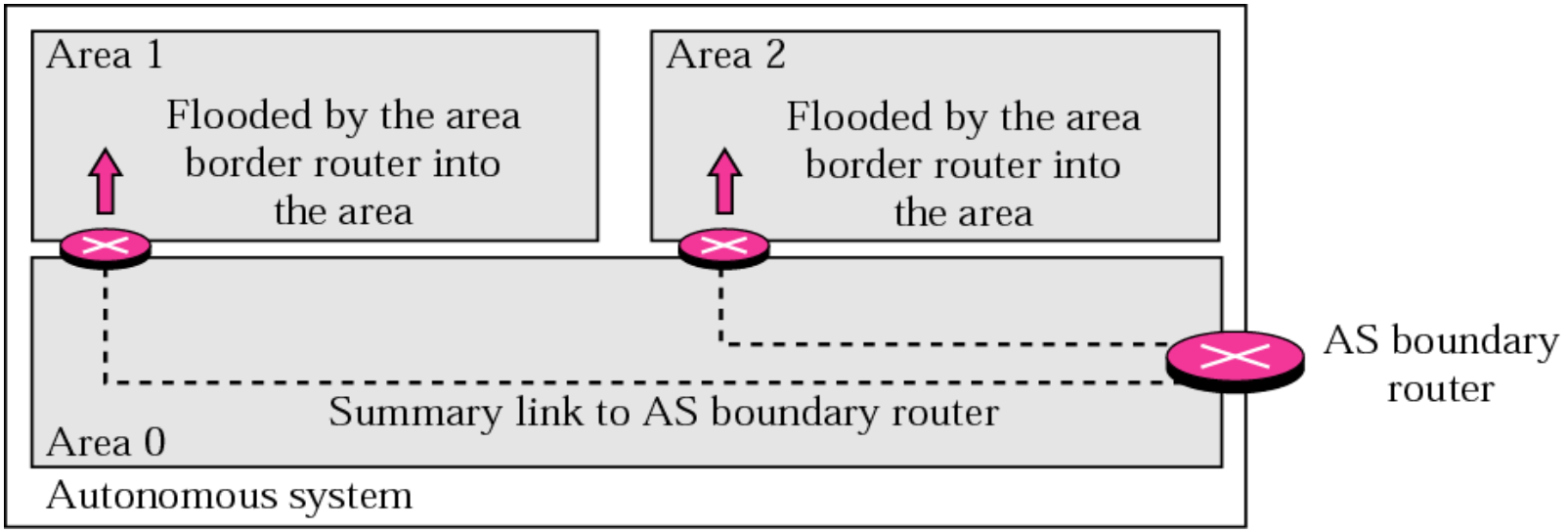


- Limits flooding of advertisements to areas

* Figure is courtesy of B. Forouzan

OSPF Summary Link



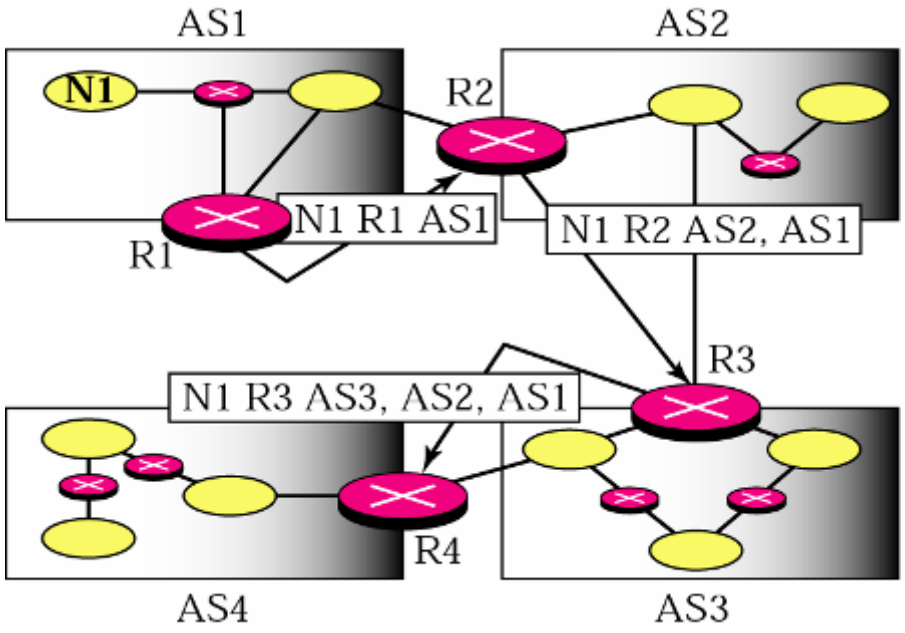


* Figure is courtesy of B. Forouzan

Border Gateway Protocol (BGP)

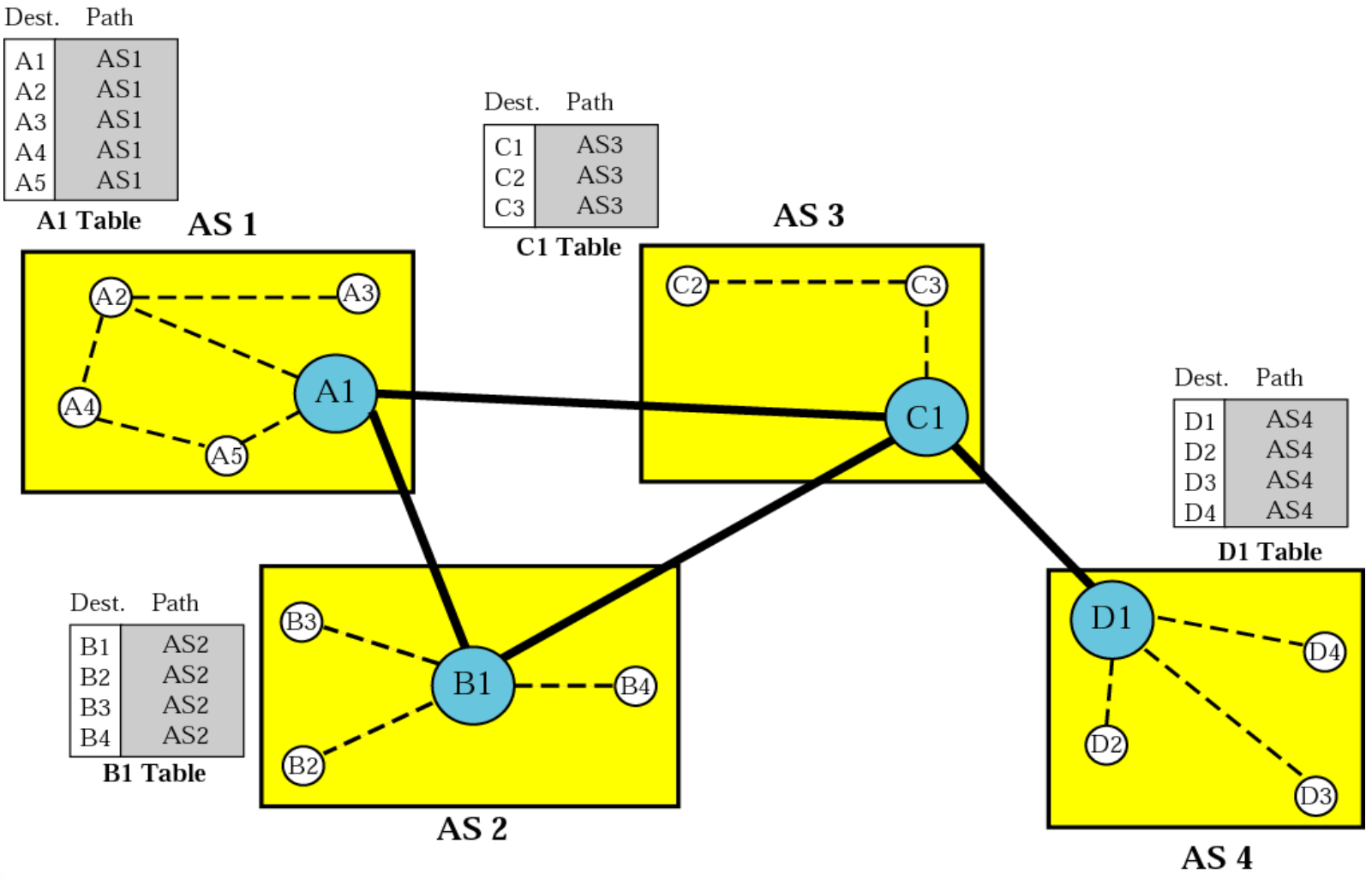
- Uses Path Vector Routing
- Advertisements include complete path to destination
- Router that forwards advertisement adds itself to the list
- Path can be checked for loops
- Policies are applied when incorporating new routes

Network	Next Router	Path
N01	R01	AS14, AS23, AS67
N02	R05	AS22, AS67, AS05, AS89
N03	R06	AS67, AS89, AS09, AS34
N04	R12	AS62, AS02, AS09



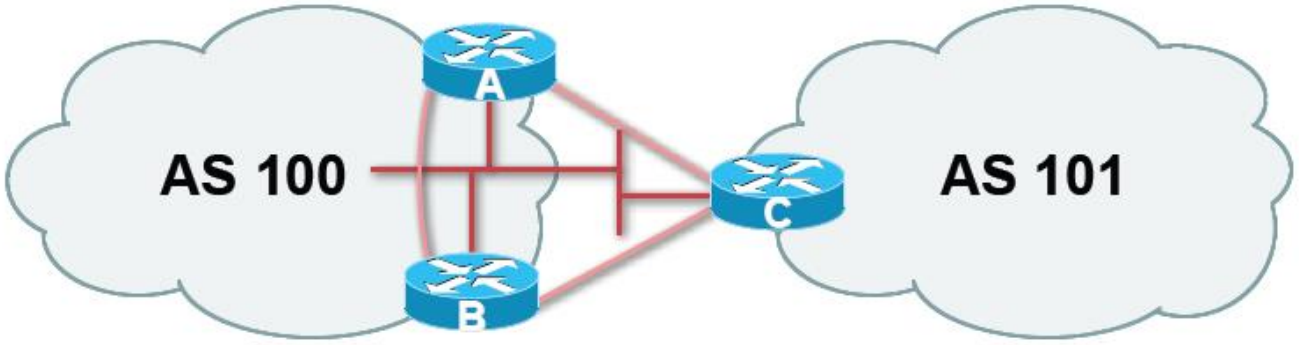
* Figure is courtesy of B. Forouzan

Tables at Autonomous Systems



* Figure is courtesy of B. Forouzan

BGP Example



Router A in AS100

```
interface ethernet 5/0
 ip address 102.102.10.2 255.255.255.240
!
router bgp 100
 network 100.100.8.0 mask 255.255.252.0
 neighbor 102.102.10.1 remote-as 101
 neighbor 102.102.10.1 prefix-list RouterC in
 neighbor 102.102.10.1 prefix-list RouterC out
!
```

ip address on ethernet interface

Local ASN

Remote ASN

ip address of Router C ethernet interface

Inbound and outbound filters

Router C in AS101

```
interface ethernet 1/0/0
 ip address 102.102.10.1 255.255.255.240
!
router bgp 101
 network 100.100.8.0 mask 255.255.252.0
 neighbor 102.102.10.2 remote-as 100
 neighbor 102.102.10.2 prefix-list RouterA in
 neighbor 102.102.10.2 prefix-list RouterA out
!
```

ip address on ethernet interface

Local ASN

Remote ASN

ip address of Router A ethernet interface

Inbound and outbound filters

* Figure is courtesy of Cisco

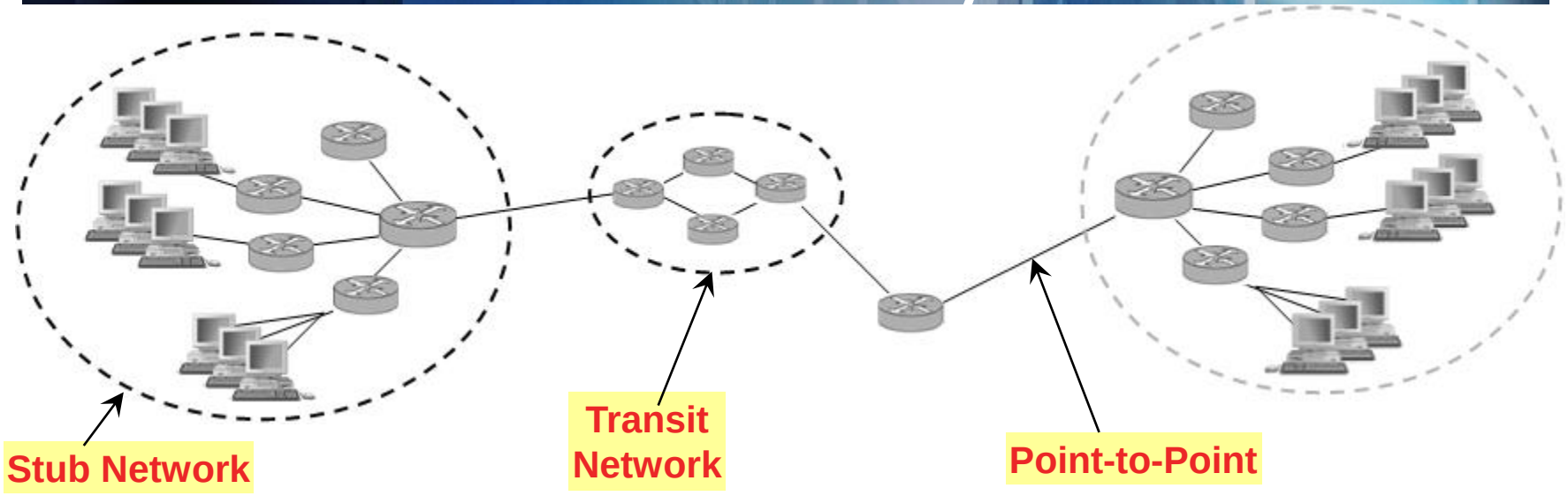
BGP-4: Border Gateway Protocol

- AS Types
  - Stub AS: has a single connection to one other AS
    - Carries local traffic only
  - Multihomed AS: has connections to more than one AS
    - Refuses to carry transit traffic
  - Transit AS: has connections to more than one AS
    - Carries both transit and local traffic

- Each AS has:
  - One or more border routers
  - One BGP *speaker* that advertises:
    - Local networks
    - Other reachable networks (transit AS only)
    - Gives *path* information

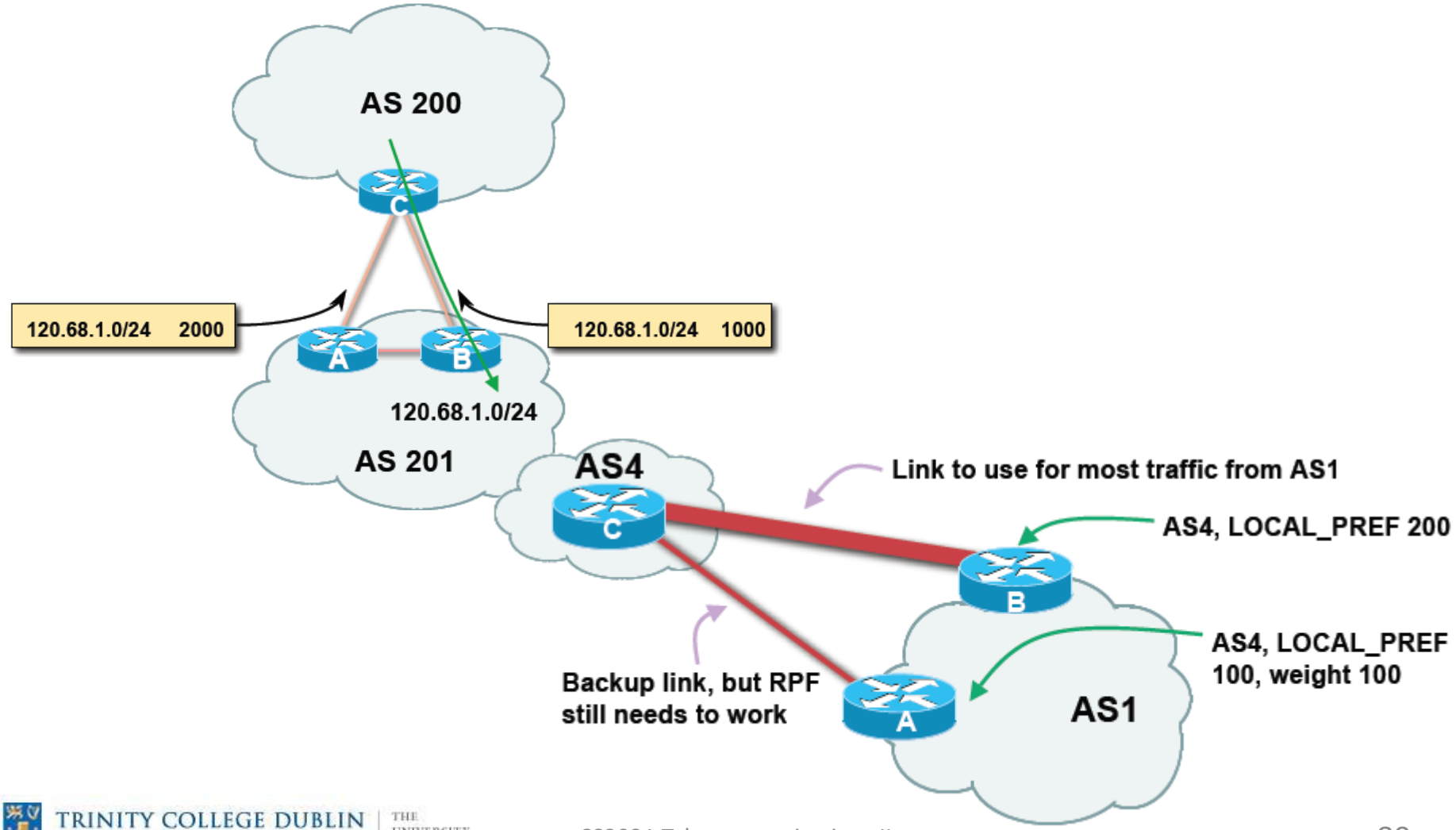
* Taken from Vivek Pai, Princeton - L Peterson & B Davie

Autonomous Systems

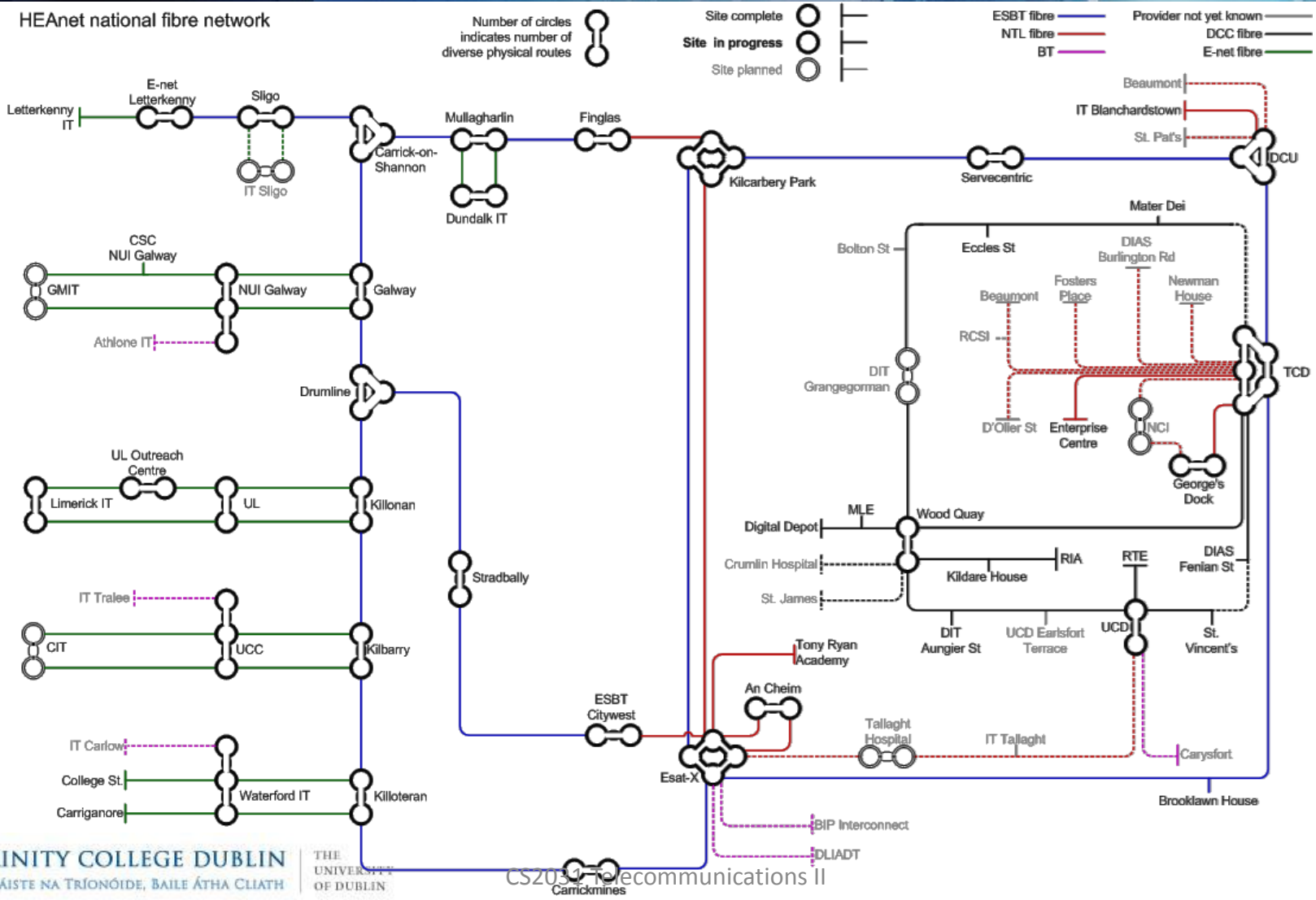


- Stub network
  - Network that does not forward to other network
- Transit network
  - Network that forwards traffic between other networks
- Point-to-point link

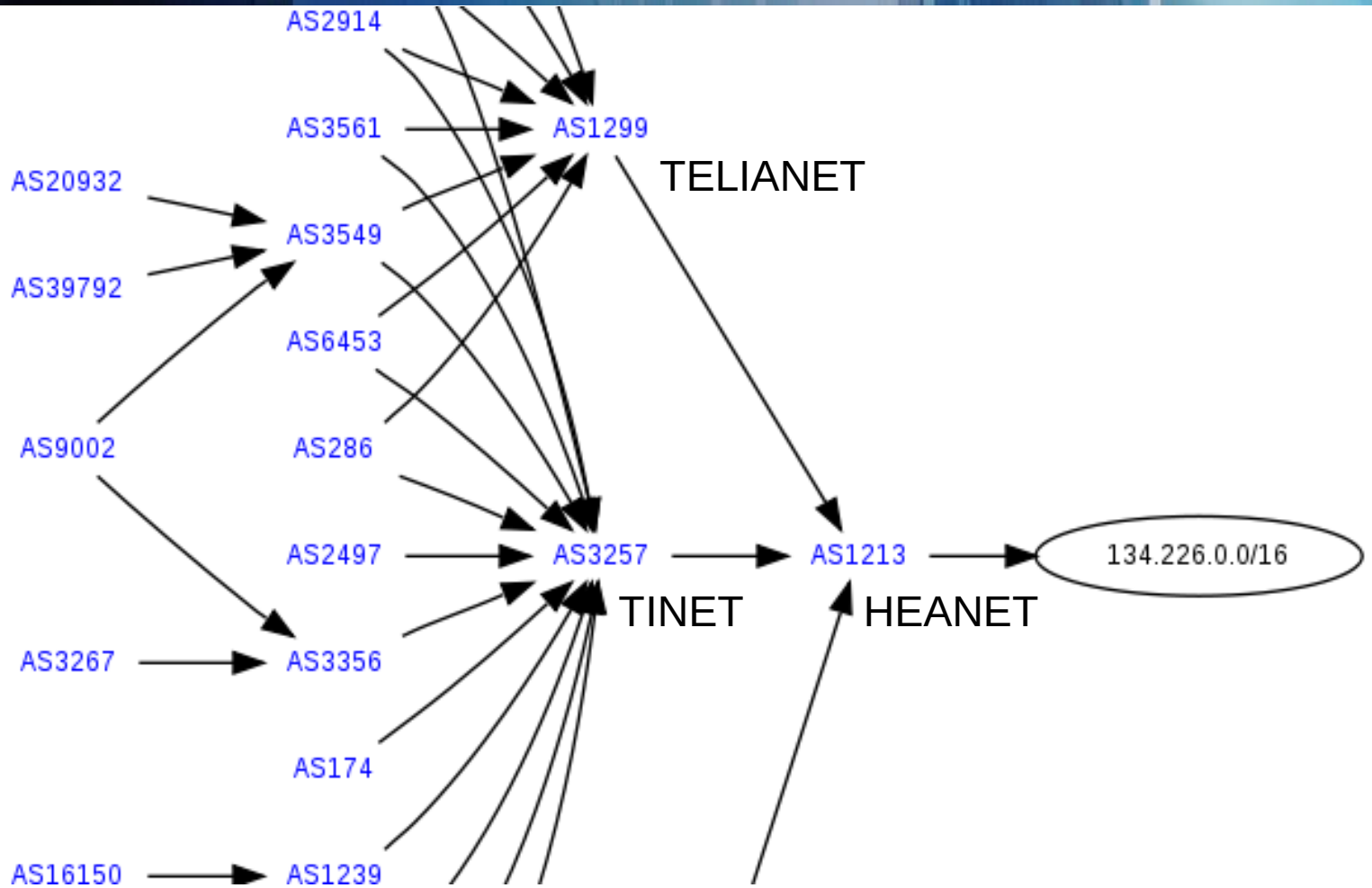
Multiple Exits & Backup Links



HEAnet Fibre Network

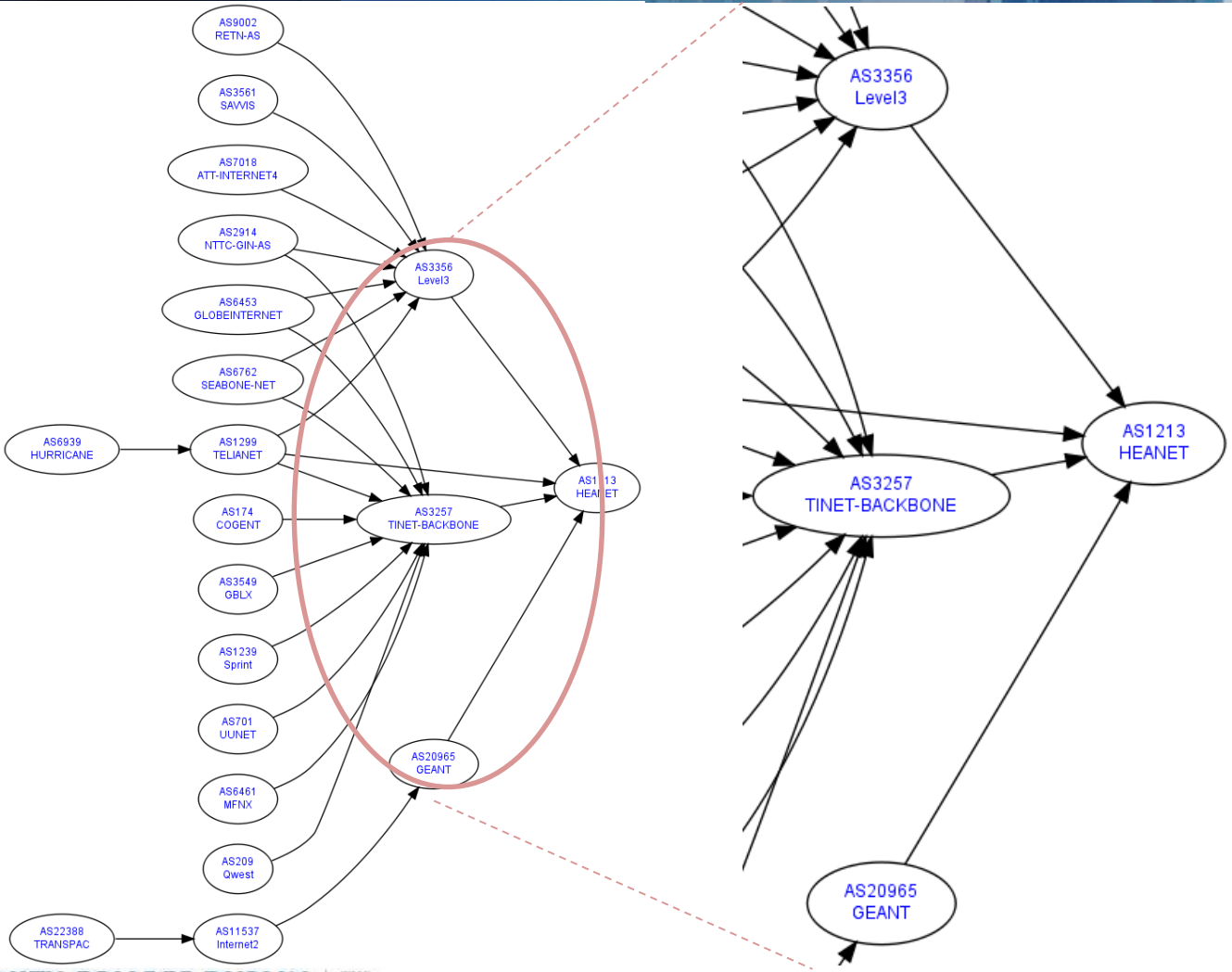


AS1213 - HEANET



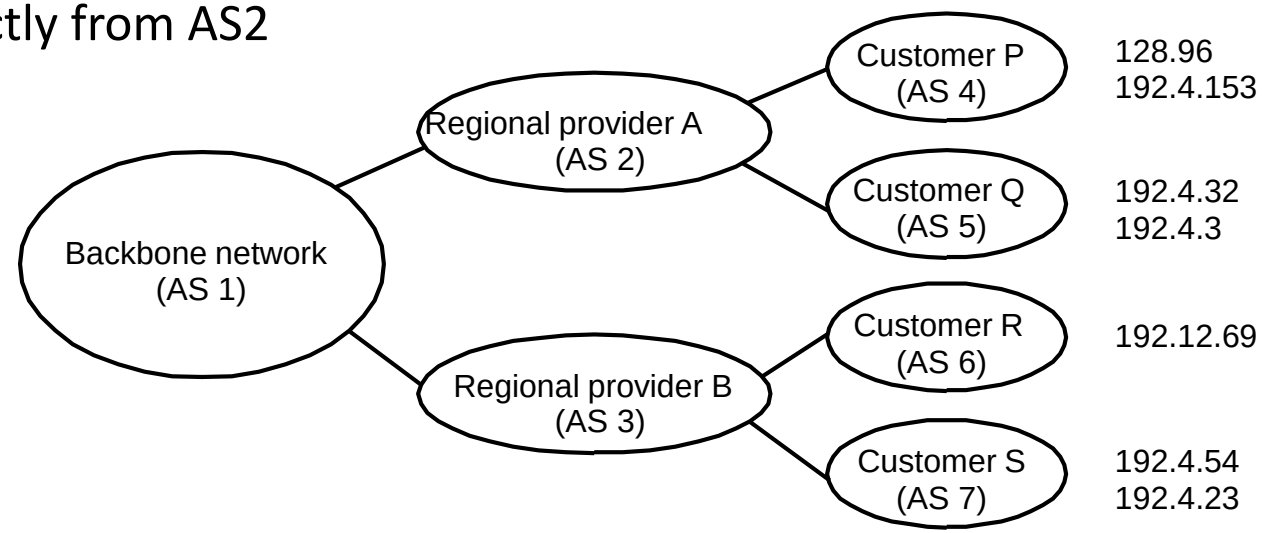
http://www.robtex.com/route/134.226.0.0-16.html

AS1213 - HEANET



BGP Example

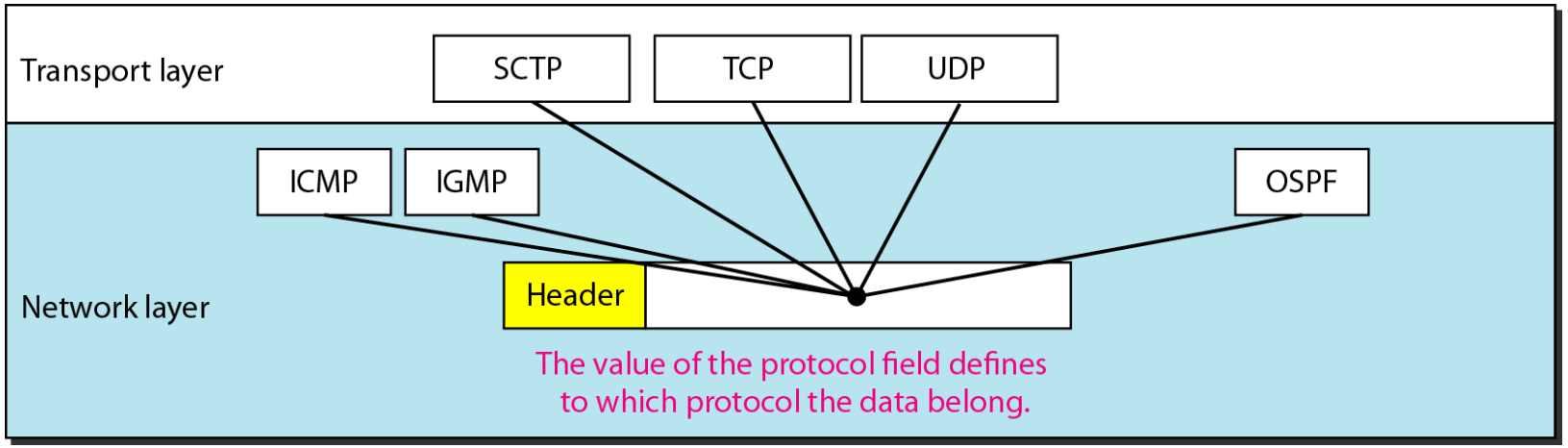
- Speaker for AS2 advertises reachability to P and Q
  - network 128.96, 192.4.153, 192.4.32, and 192.4.3, can be reached directly from AS2



- Speaker for backbone advertises
  - networks 128.96, 192.4.153, 192.4.32, and 192.4.3 can be reached along the path (AS1, AS2).
- Speaker can cancel previously advertised paths

* Taken from Vivek Pai, Princeton - L Peterson & B Davie

Routing Protocols in the Stack



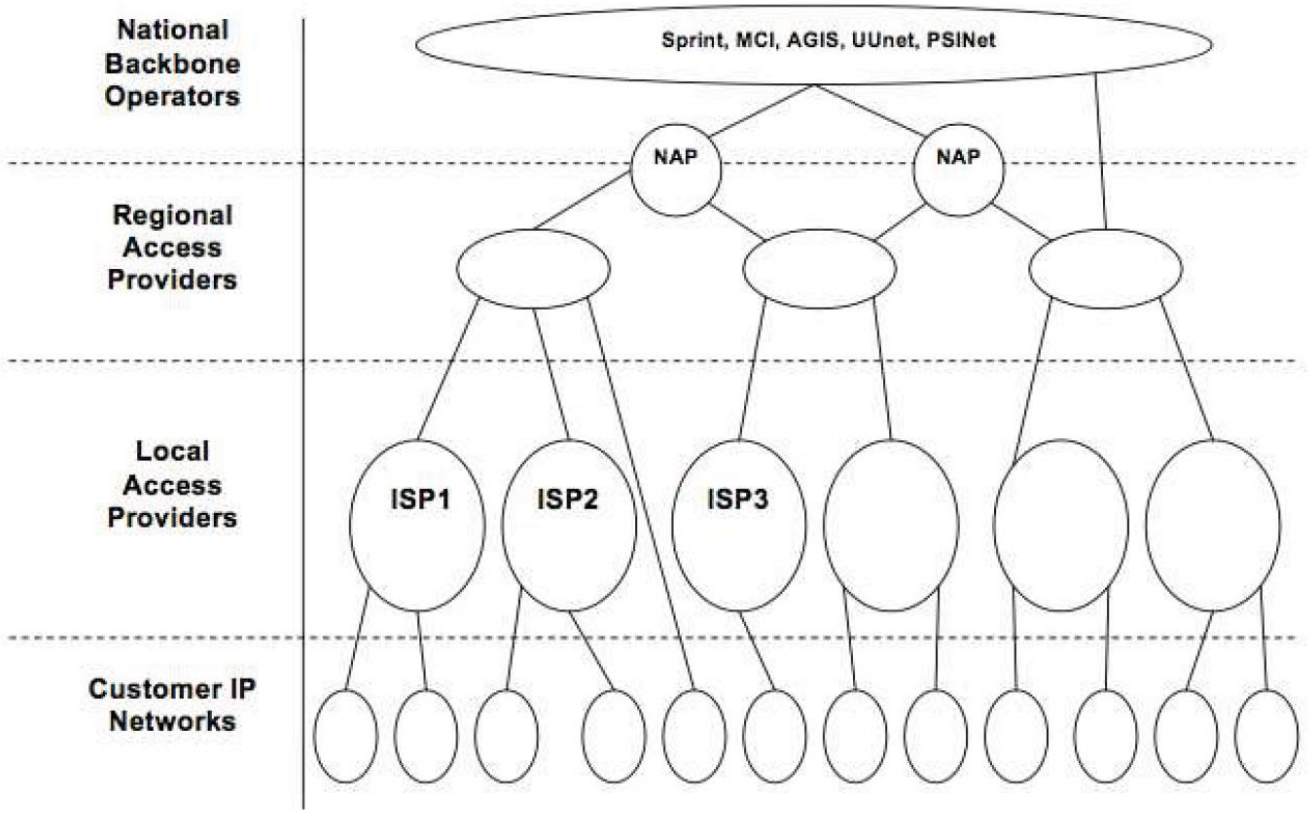
Protcol	Underlying Protcol	Protocol ID or Port
OSPF	IP	89
RIPv2	UDP	520
BGP	TCP	179

* Figure is courtesy of B. Forouzan

Summary: Routing

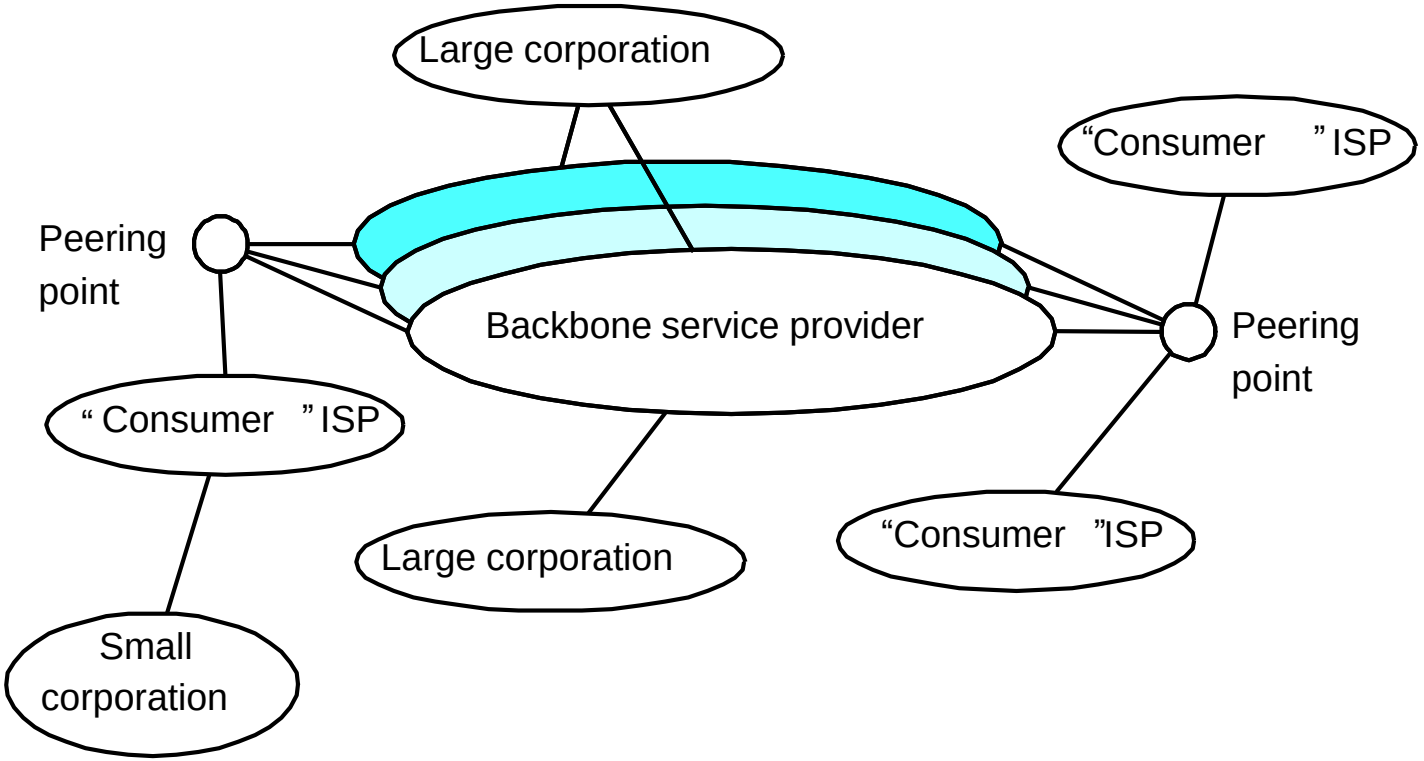
- Autonomous Systems
  - Stub network
  - Transient network
  - Point-to-point link
- Distance Vector routing
  - Share complete information with neighbours
  - Count-to-Infinity problem
  - Example: Routing Information Protocol (RIP)
- Link State routing
  - Share information about neighbours with everyone
  - Dijkstra's Shortest-Path Algorithm
  - Example: Open Shortest Path First (OSPF)

Traditional Logical Internet Topology



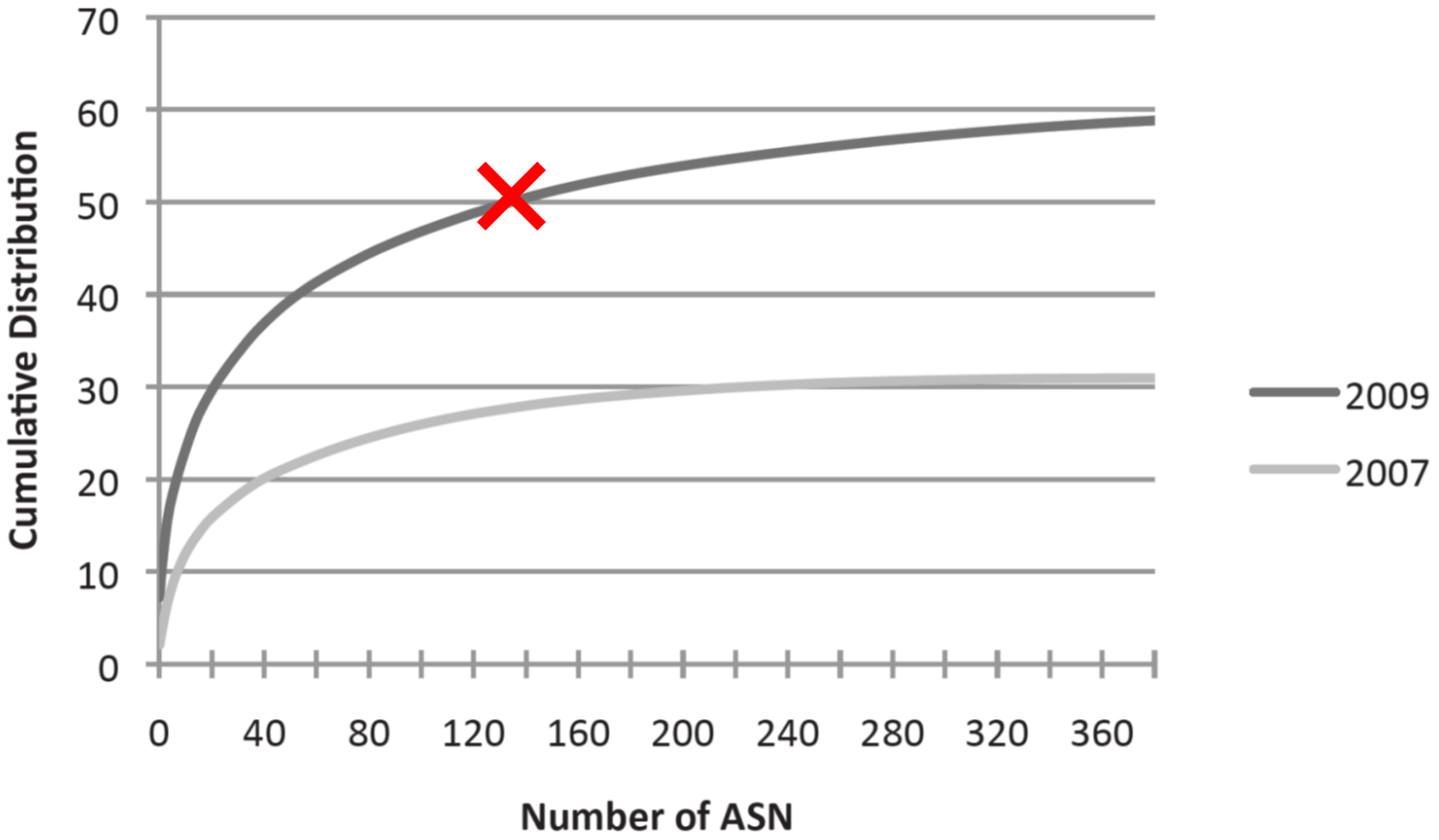
* Figure is courtesy of C. Labovitz, et al, SIGCOMM 2010

Structure of the Internet



- Tier 1 provider eg. Verizon (WorldCom/MCI), Global Crossing
- Peering points – Exchange points between ISPs

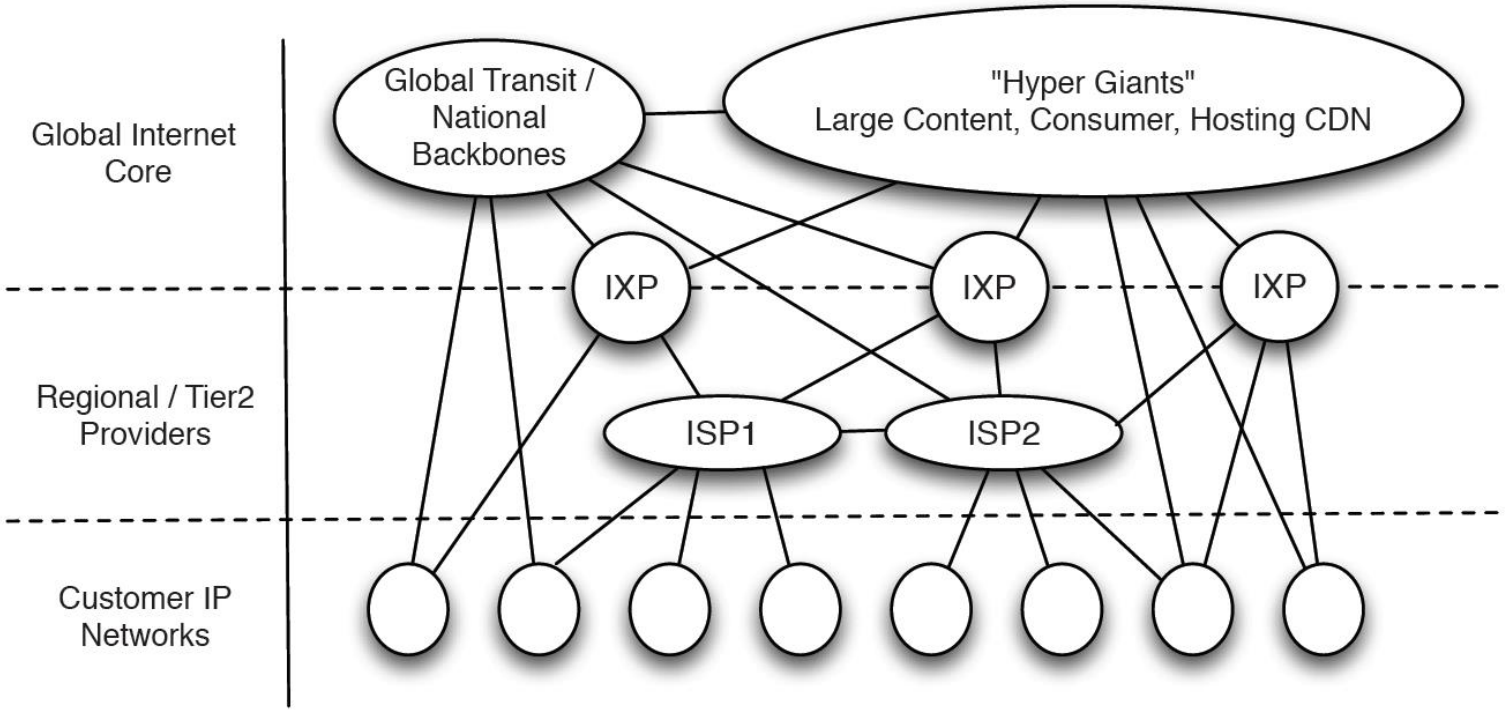
Cumulative Inter-Domain Traffic



150 ASN in 2009 – According to C. Labovitz now 30

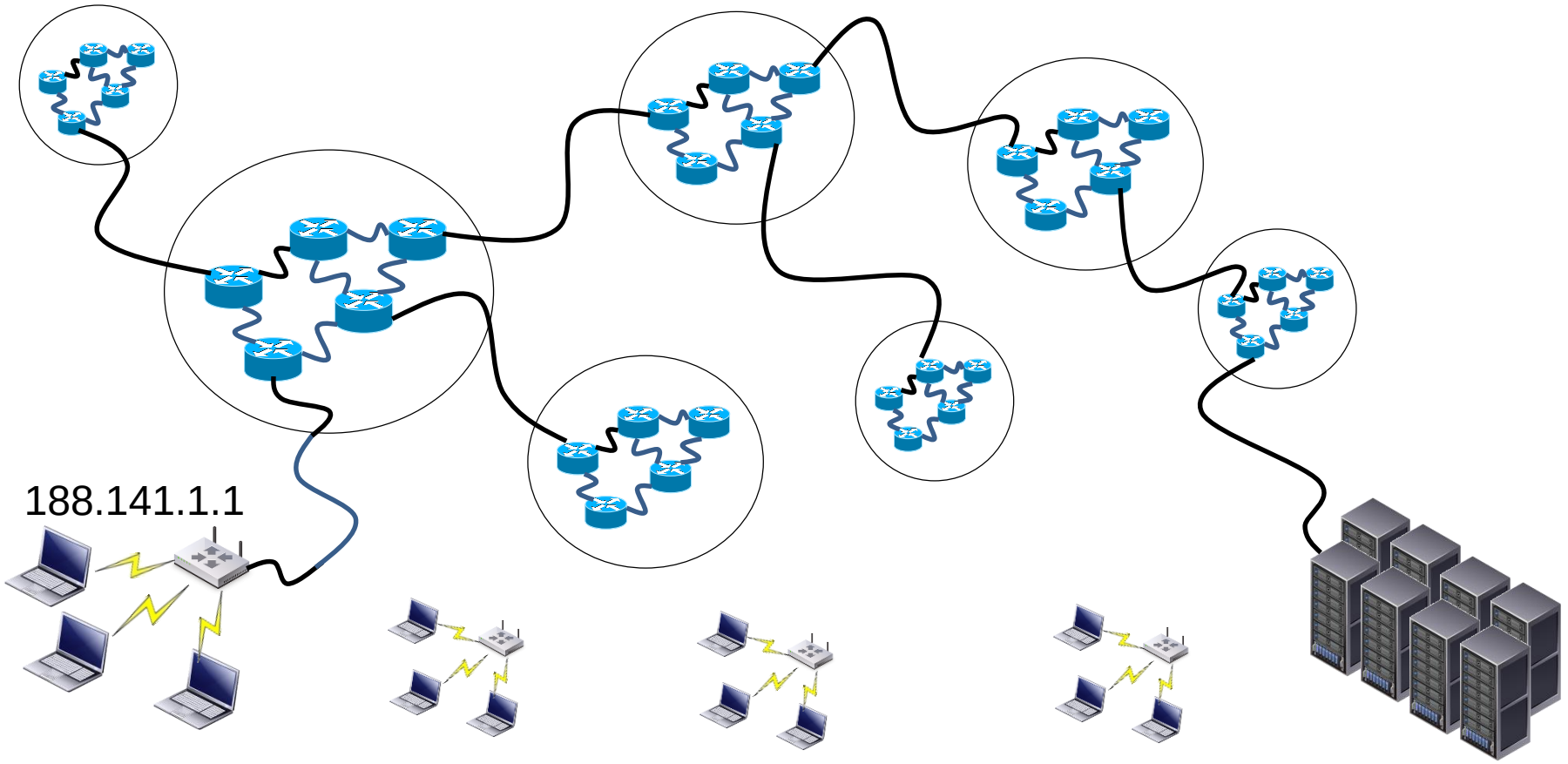
* Figure is courtesy of C. Labovitz, et al, SIGCOMM 2010

Emerging Logical Internet Topology



* Figure is courtesy of C. Labovitz, et al, SIGCOMM 2010

Internet = Network of Networks




That's all folks
Telecomms II