Student Online Teaching Advice Notice

The materials and content presented within this session are intended solely for use in a context of teaching and learning at Trinity.

Any session recorded for subsequent review is made available solely for the purpose of enhancing student learning.

Students should not edit or modify the recording in any way, nor disseminate it for use outside of a context of teaching and learning at Trinity.

Please be mindful of your physical environment and conscious of what may be captured by the device camera and microphone during videoconferencing calls.

Recorded materials will be handled in compliance with Trinity's statutory duties under the Universities Act, 1997 and in accordance with the University's policies and procedures.

Further information on data protection and best practice when using videoconferencing software is available at https://www.tcd.ie/info_compliance/data-protection/.

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

CSU33031 Computer Networks

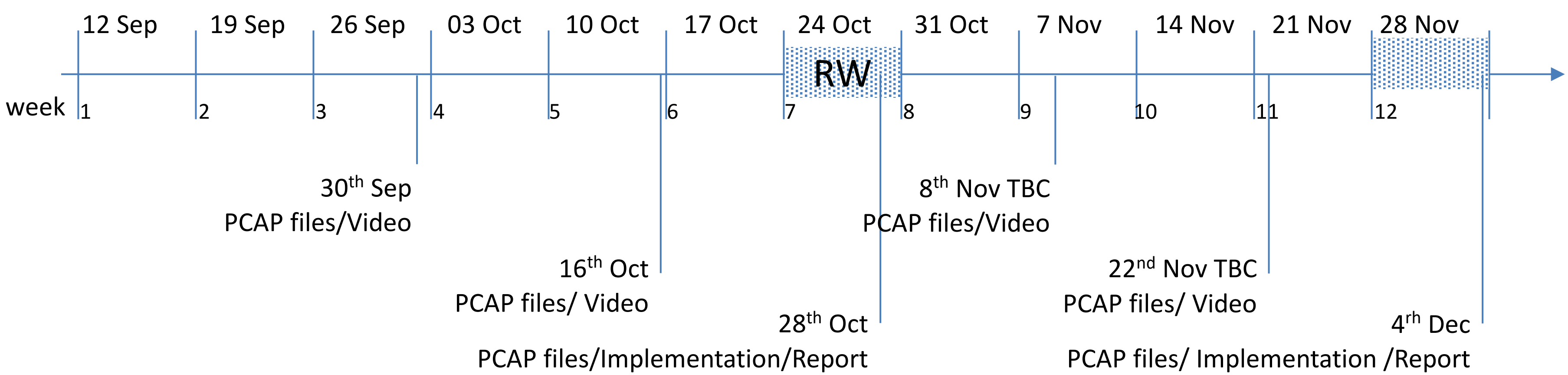
IPv4/v6 Recap

Stefan Weber
email: sweber@tcd.ie
Office: Lloyd 1.41

Plan for today

- Routing and OpenFlow
  - Distance Vector
  - Link State
  - (maybe BGP)

CSU33031 Timeline & Weights



- Report template & example for template
- Overleaf

Report Template

- Describe images
- Justify choices
- Snippets of code; not full code dumps

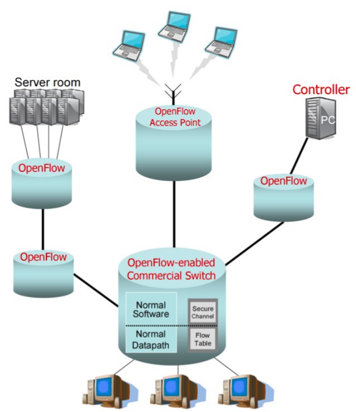


Figure 1: The figure above demonstrates a general topology of a network using Openflow. The switches in the network are connected to a controller. A switch will contact this controller when an incoming packet has no matching rules in the flowtable.

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

CSU33031 Computer Networks
Assignment #1: Protocols

Sean Murphy, Std# 1800123

January 24, 2022

Contents

1 Introduction

The introduction should describe the general problem that is the focus of the assignment and outline the approach that you have taken to address the problem.

2 Theory of Topic

The section on the theory at the basis of the assignment should describe the concepts and protocols that were used to realize a solution.
The beginning of a section should explain to the reader the overall motivation of the section, the parts that make up the section, and why these parts are relevant i.e. why the reader should consider the section.

1

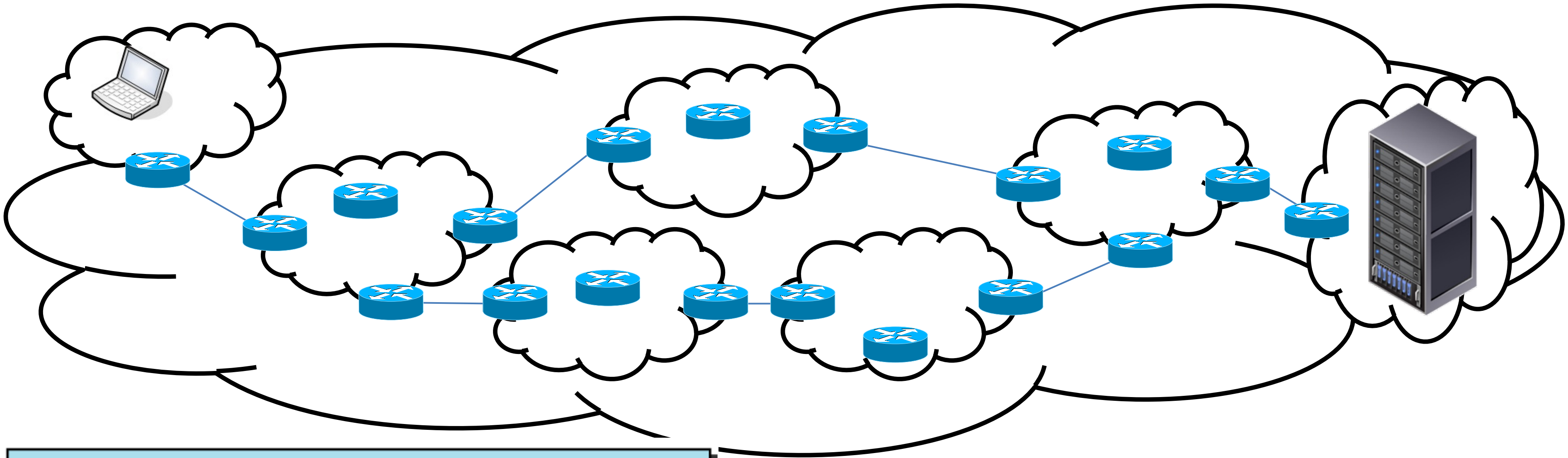
Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

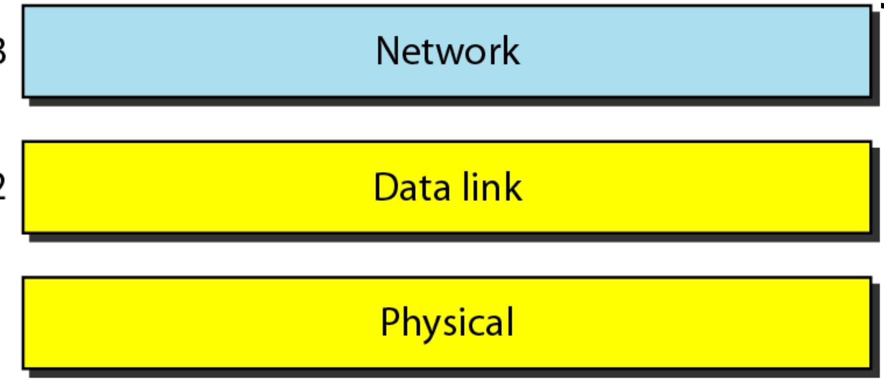
CSU33031 Computer Networks

Rerun: IPv4&IPv6

Stefan Weber
email: sweber@tcd.ie
Office: Lloyd 1.41

The Internet - Network of Networks



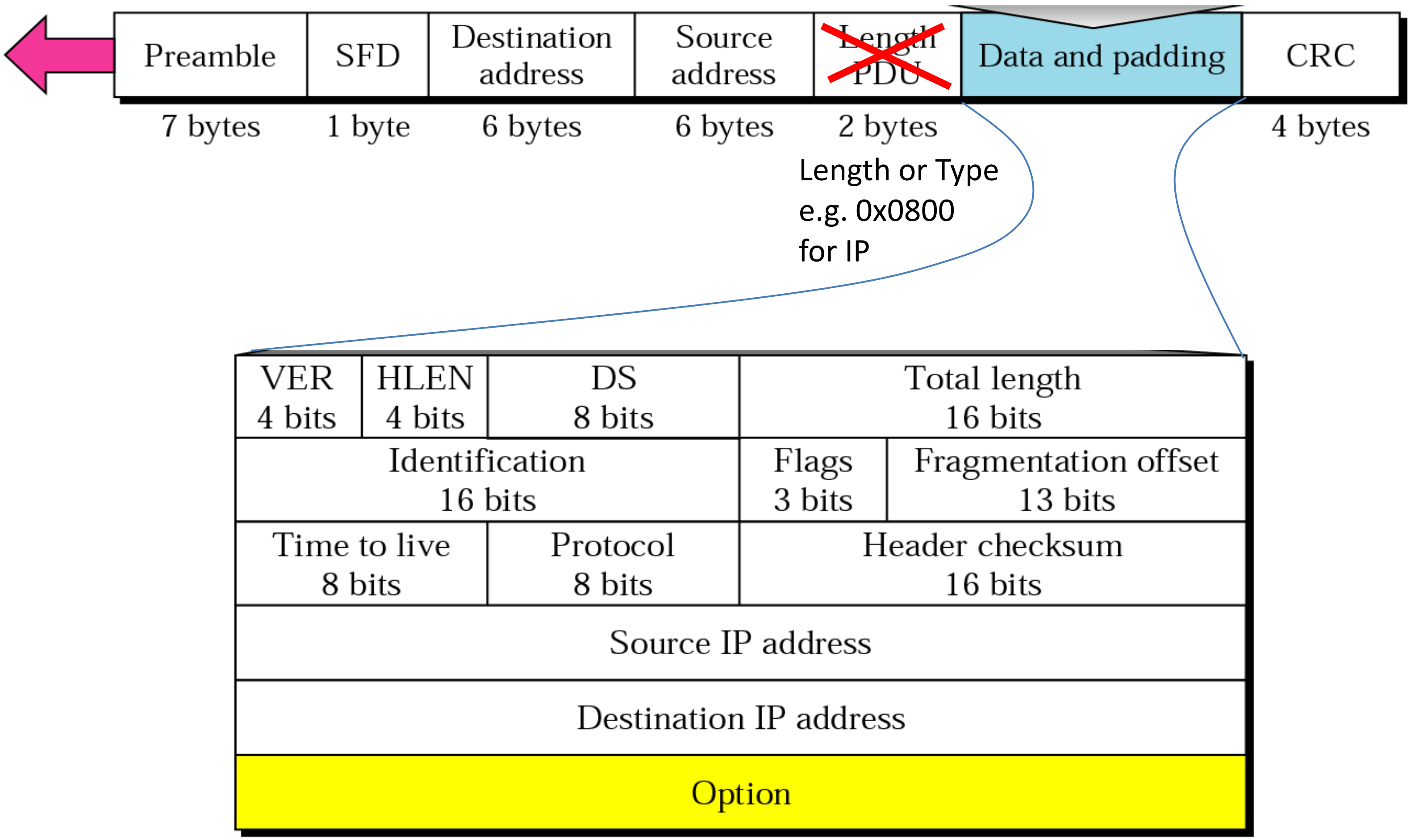


IP Service Model

- Connection-less Communication
- Order not guaranteed
- Best-effort delivery (unreliable service)

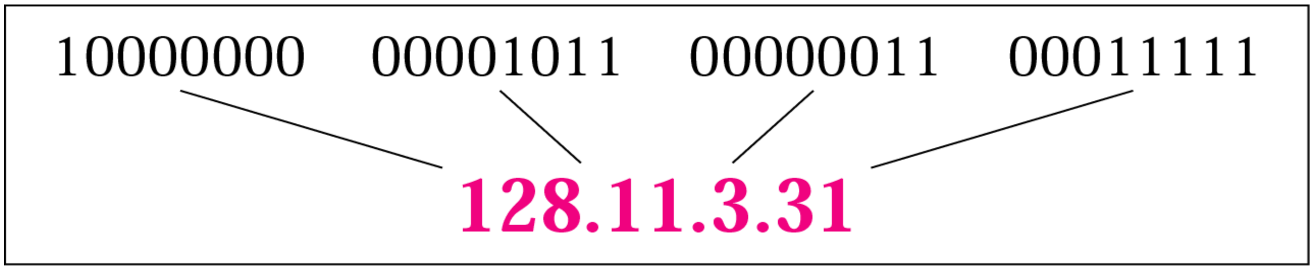
* Figure is courtesy of B. Forouzan

Ethernet & IP



* Figure is courtesy of B. Forouzan

IP Addresses



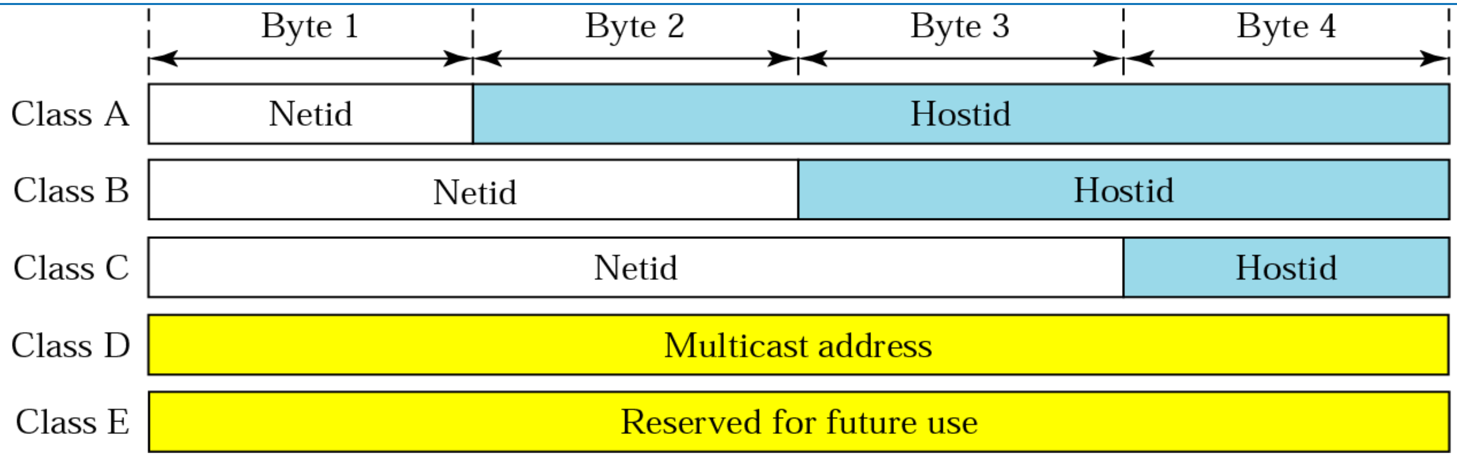
- 32-bit number
  - 4,294,967,296 addresses
- IP addresses are unique and universal
  - except private addresses

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

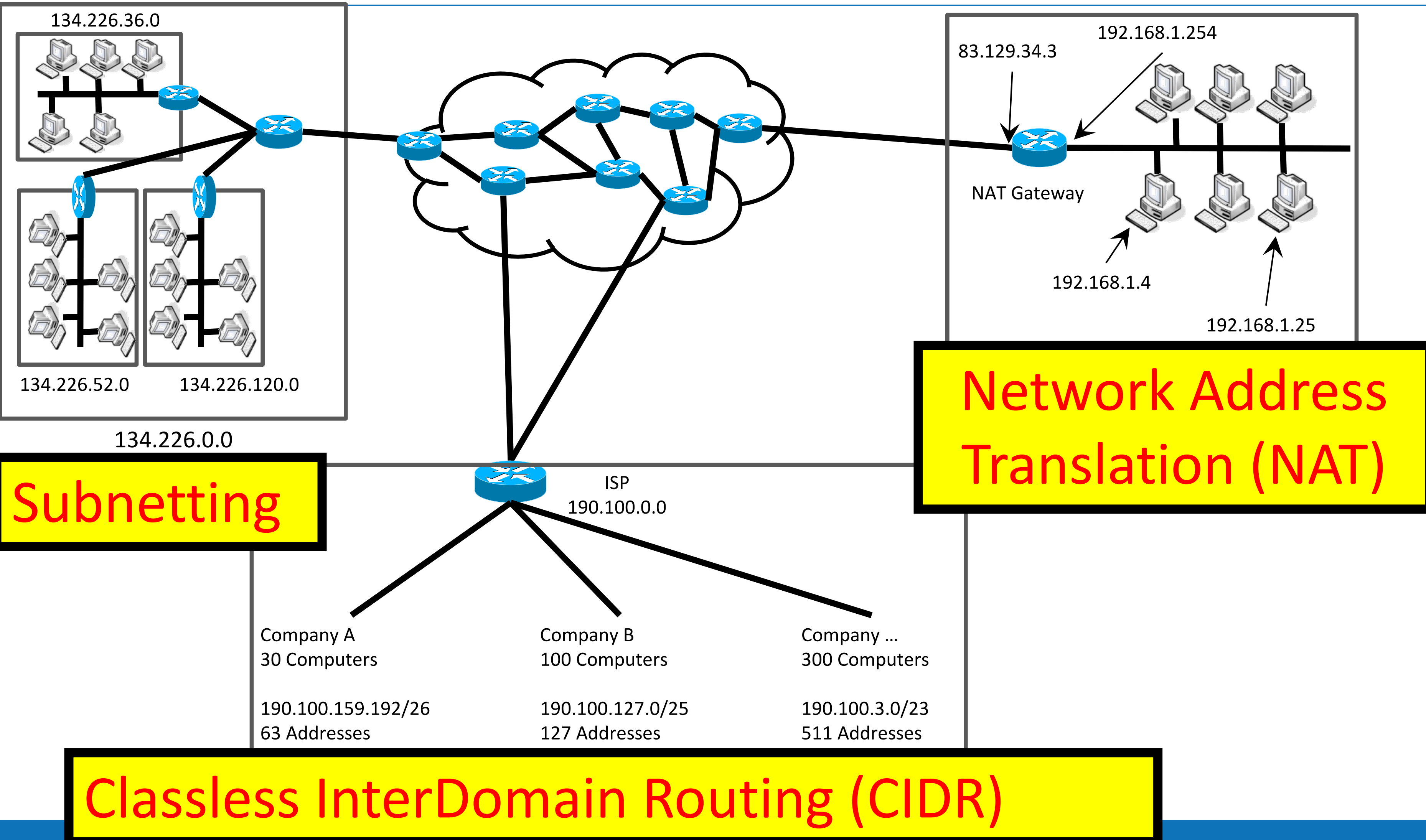


- Class A (international organisations)
  - 126 networks with 16,277,214 hosts each
- Class B (large companies)
  - 16,384 networks with 65,354 hosts each
- Class C (smaller companies)
  - 2,097,152 networks with 254 hosts each

- 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)

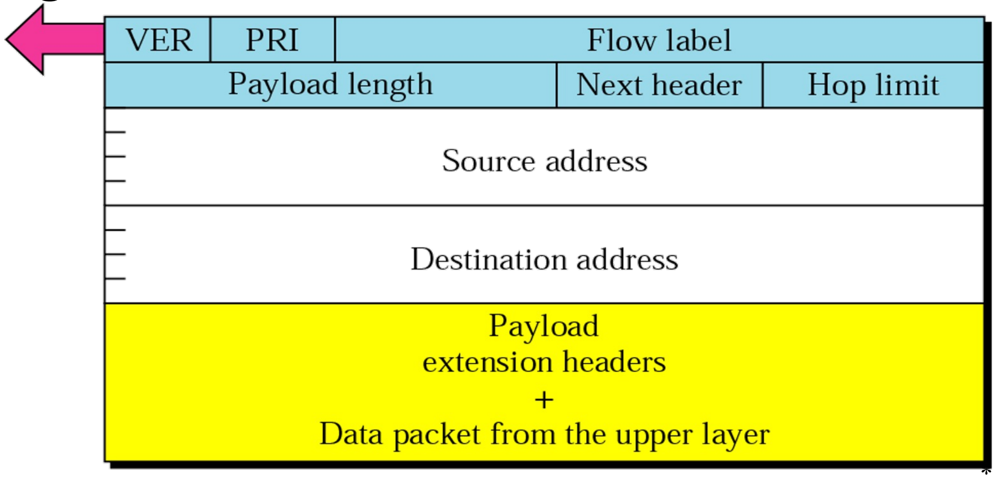
* Figure is courtesy of B. Forouzan

Developments re IP Addresses



IPv6 Header

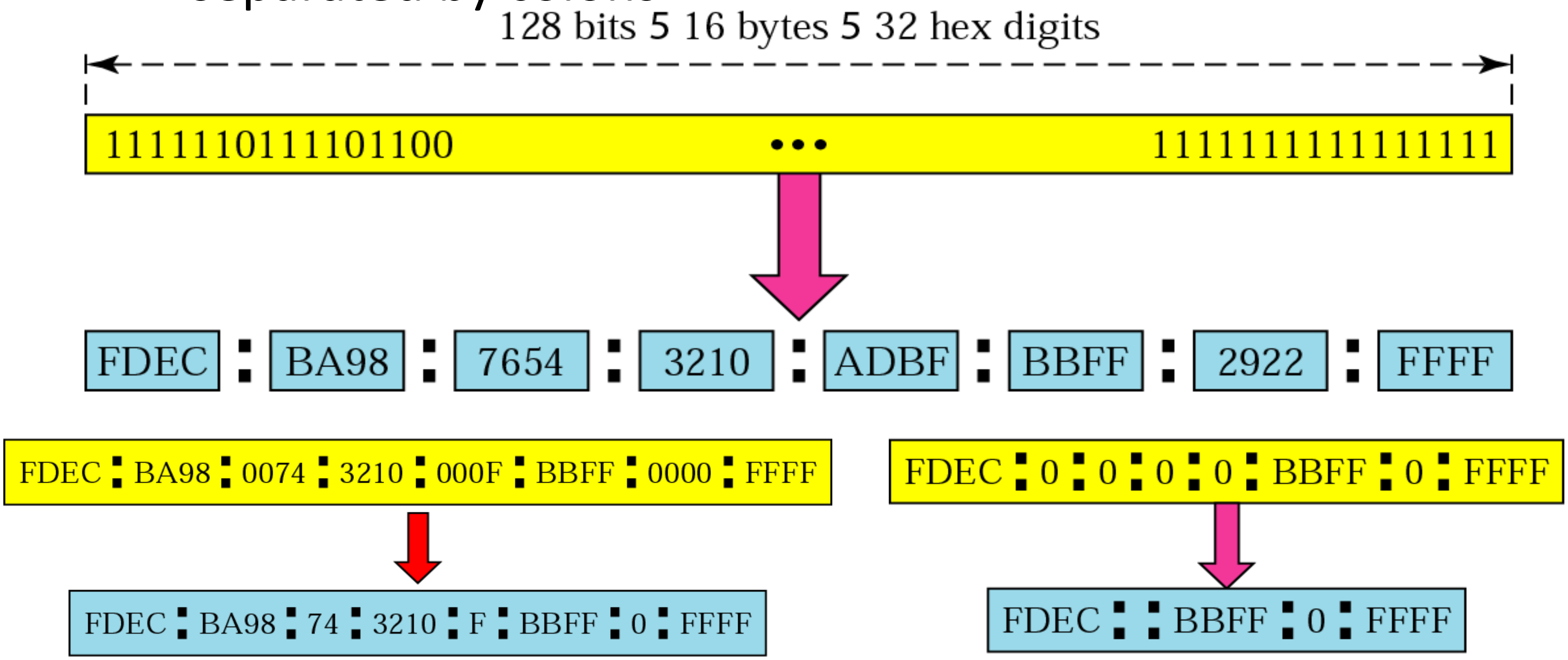
- Fixed length of all fields, header length irrelevant
- Remove Header Checksum – other layers are responsible
- No hop-by-hop fragmentation – fragment offset irrelevant
  - MTU discovery before sending or minimum MTU=1280
- Extension headers – next header type
- Basic Principle: Routers along the way should do minimal processing



* Figure is courtesy of B. Forouzan

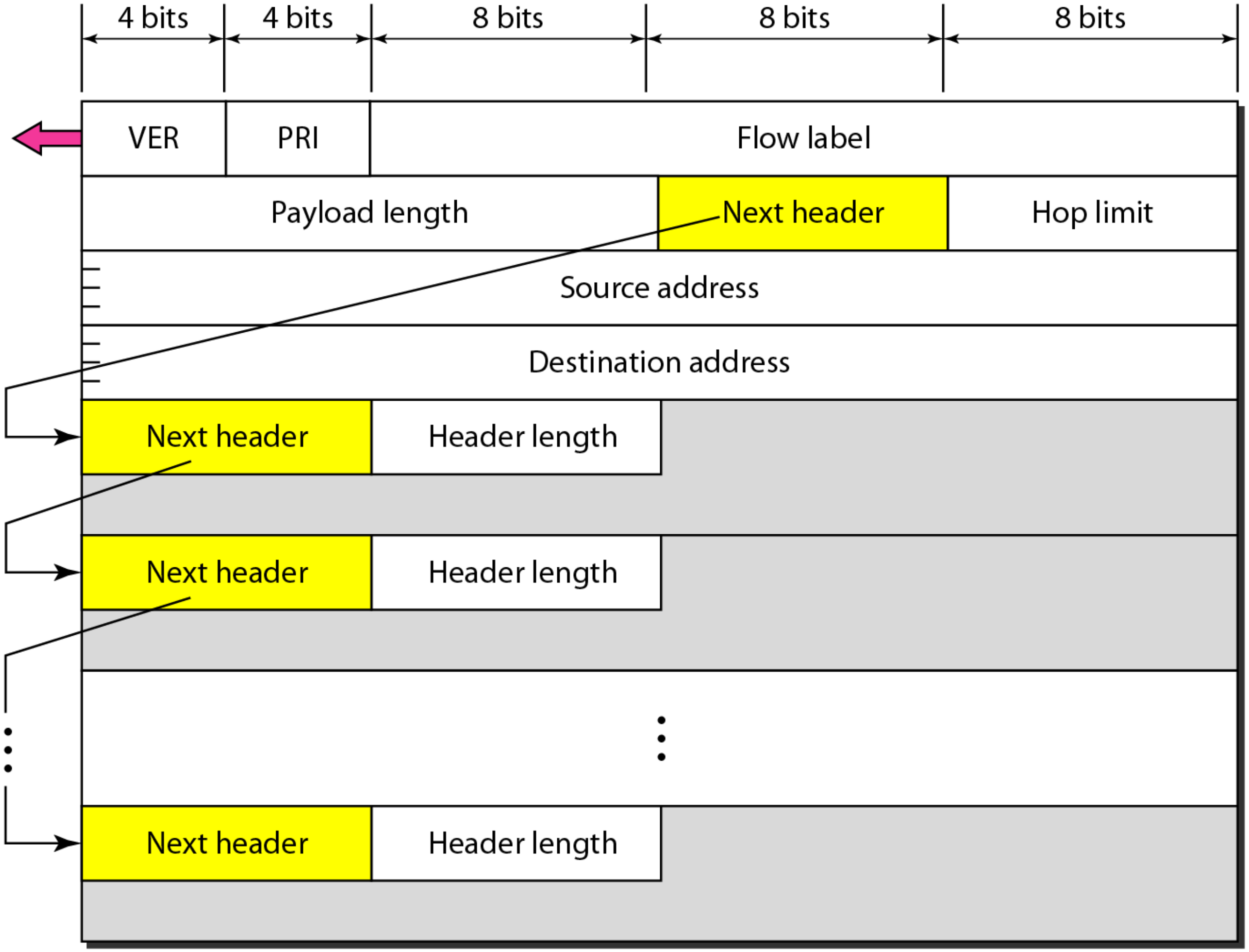
IPv6 Address

- Standard representation is set of eight 16-bit values separated by colons



* Figure is courtesy of B. Forouzan

Extension Headers



* Figure is courtesy of B. Forouzan

Summary: IP

- IP Service Model
  - Connection-less, no order or delivery guaranteed
- IPv4:
  - 32-bit addresses
  - 20 bytes + options
  - Fragmentation
- IPv6:
  - 128-bit addresses
  - Simpler, fixed-sized header
  - Extension headers
- Types of communication
  - Unicast
  - Multicast
  - Anycast

Relevant Programs

- ifconfig (ipconfig on MS Windows) – list interfaces
- netstat – list routes and network information
- ping – send information to test connectivity
- nslookup/dig – resolve name to IP address
- wireshark/tshark/tcpdump – capture network traffic

- nc, socat, ssh, curl – create/redirect network traffic

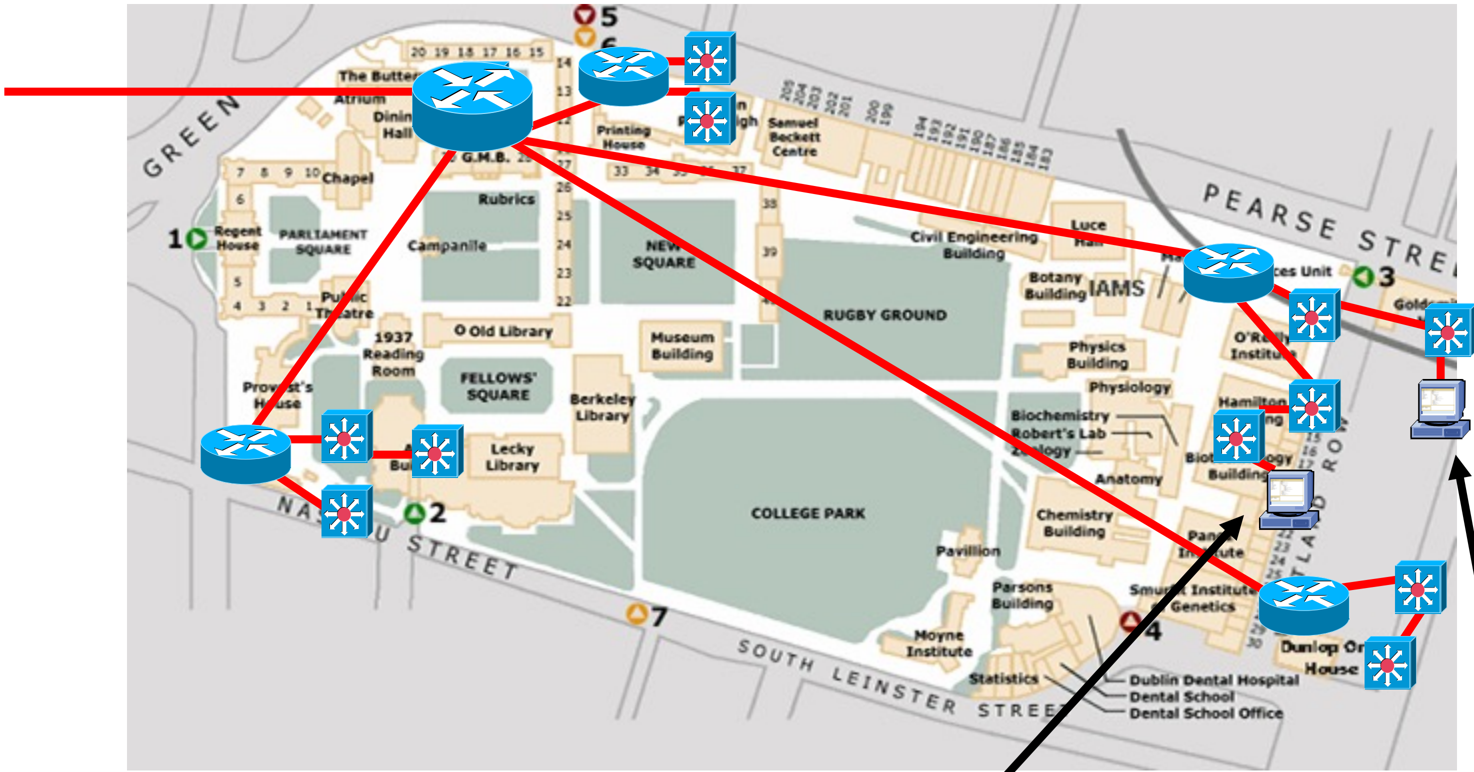
Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

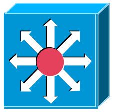
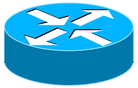
CSU33031 Computer Networks

Routing

Stefan Weber
email: sweber@tcd.ie
Office: Lloyd 1.41

Trinity Network with Routers

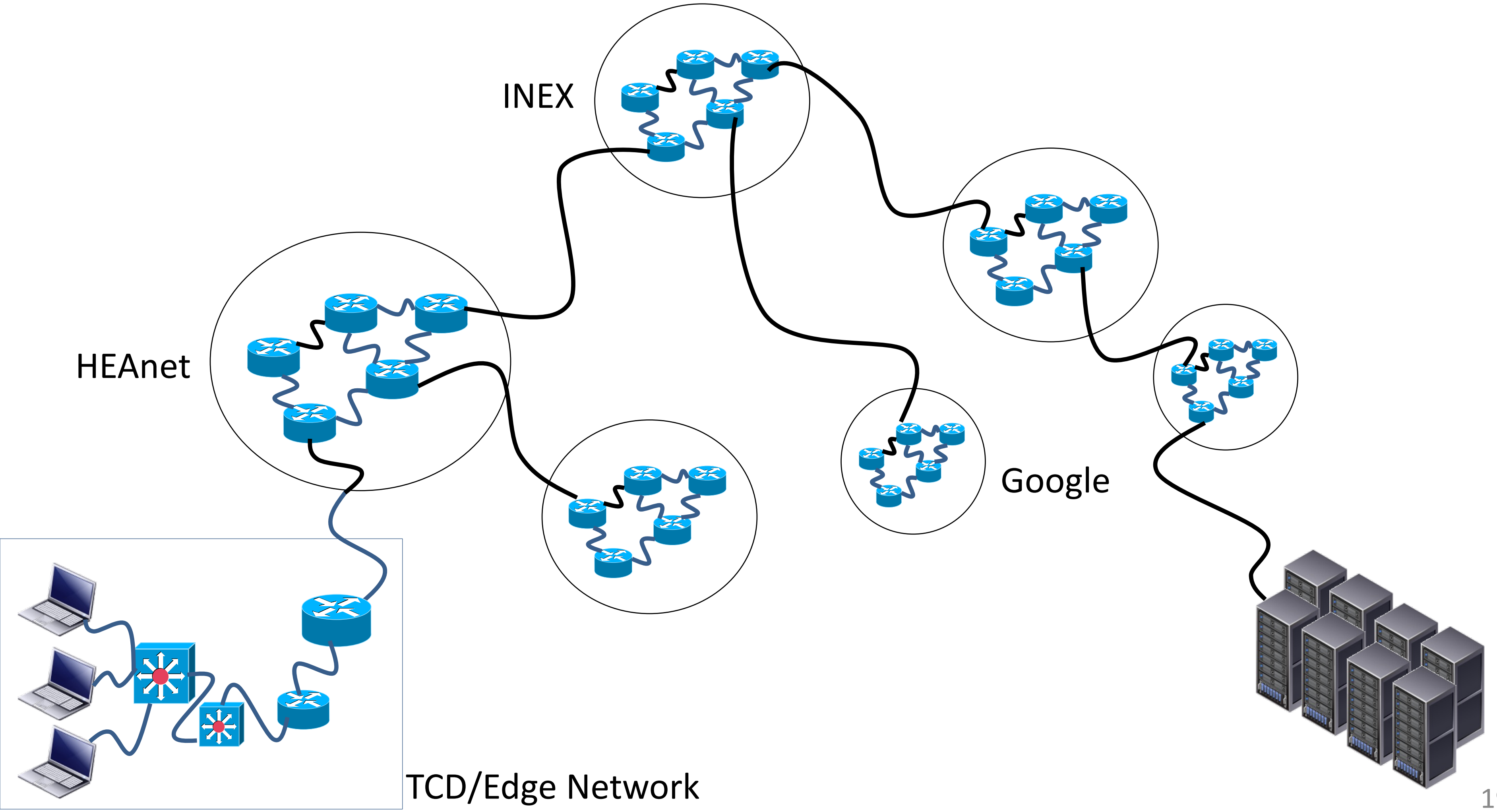


 **Switch**  **Router**

Address: 134.226.38.55
Subnet: 134.226.38.0

Address: 134.226.32.55
Subnet: 134.226.32.0

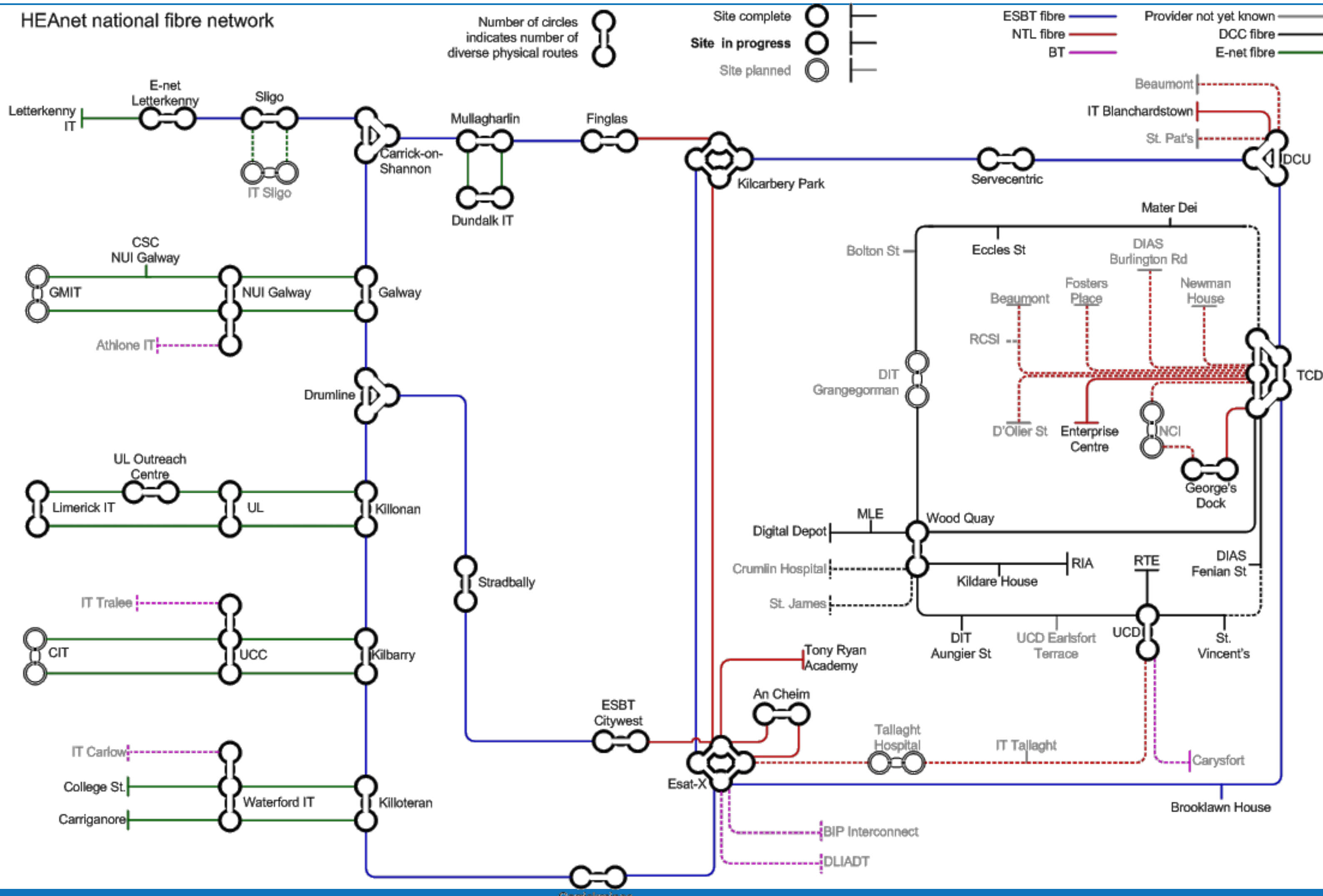
Internet = Network of Networks



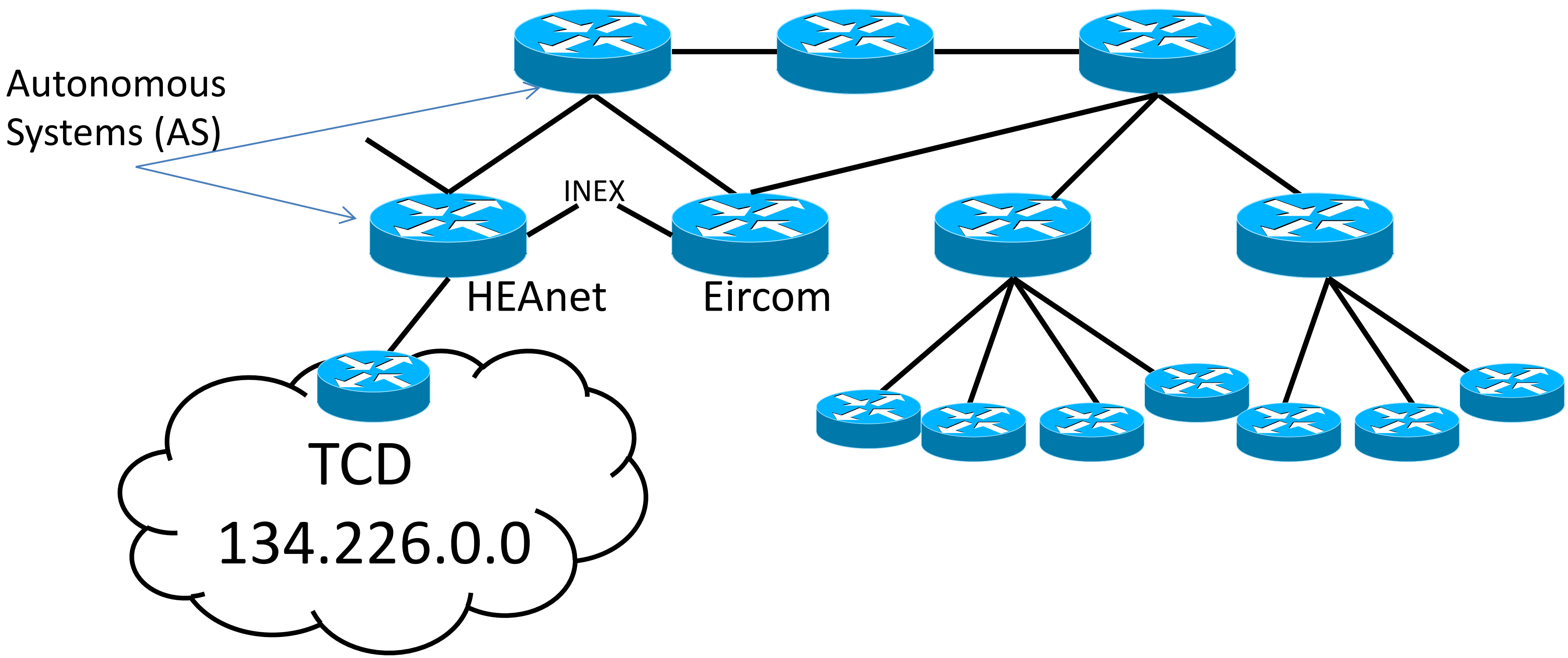
HEAnet Fibre Network

(A little outdated)

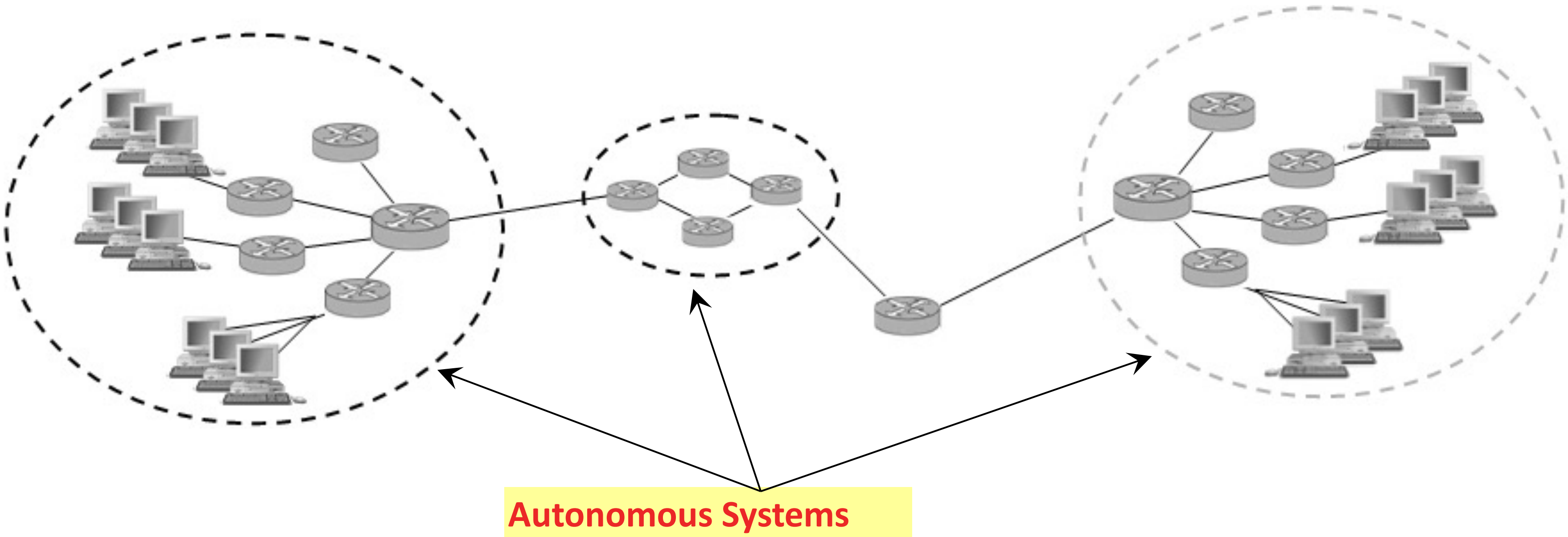
Shout-out to HEAnet: Eoin Kenny who keeps our networks running



Trinity's Connection

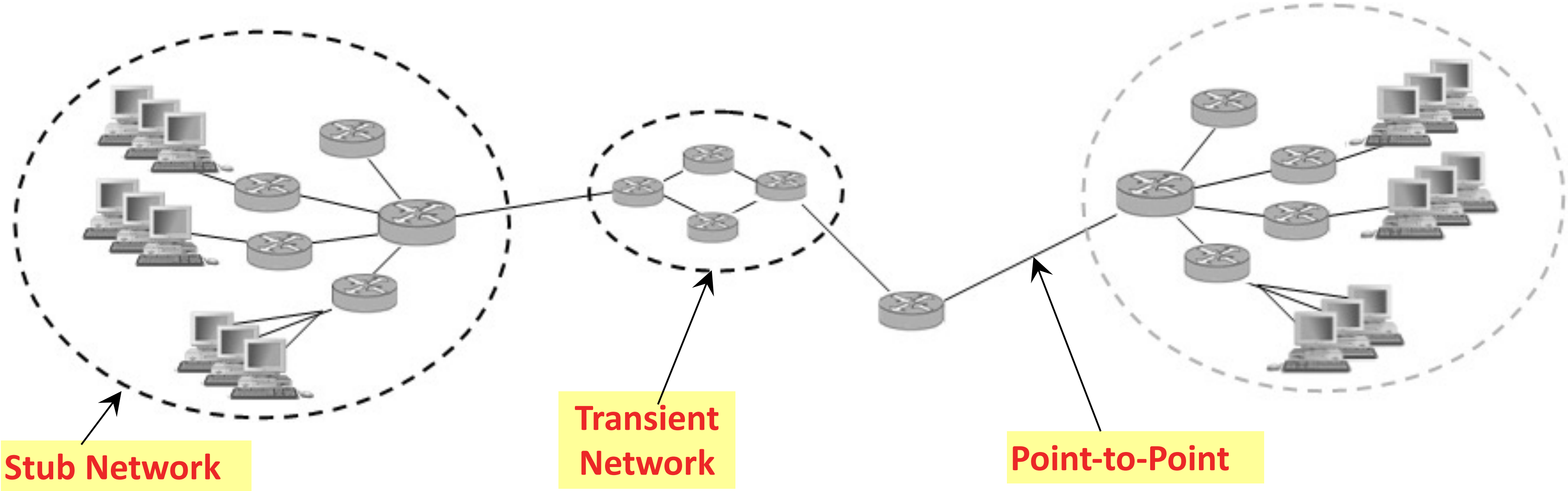


Structure of the Internet



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

Autonomous Systems

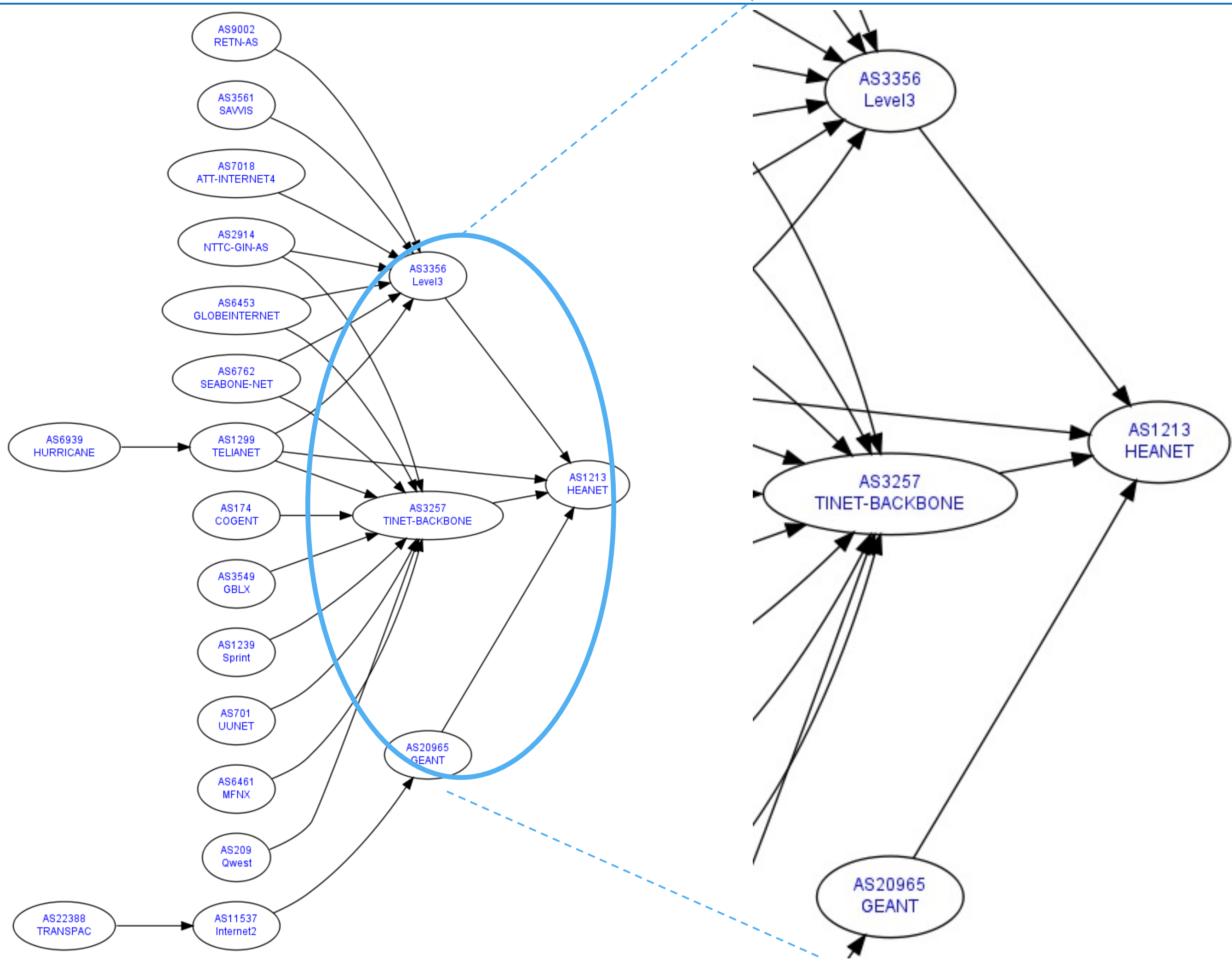


- Stub network
  - Network that does not forward to other network

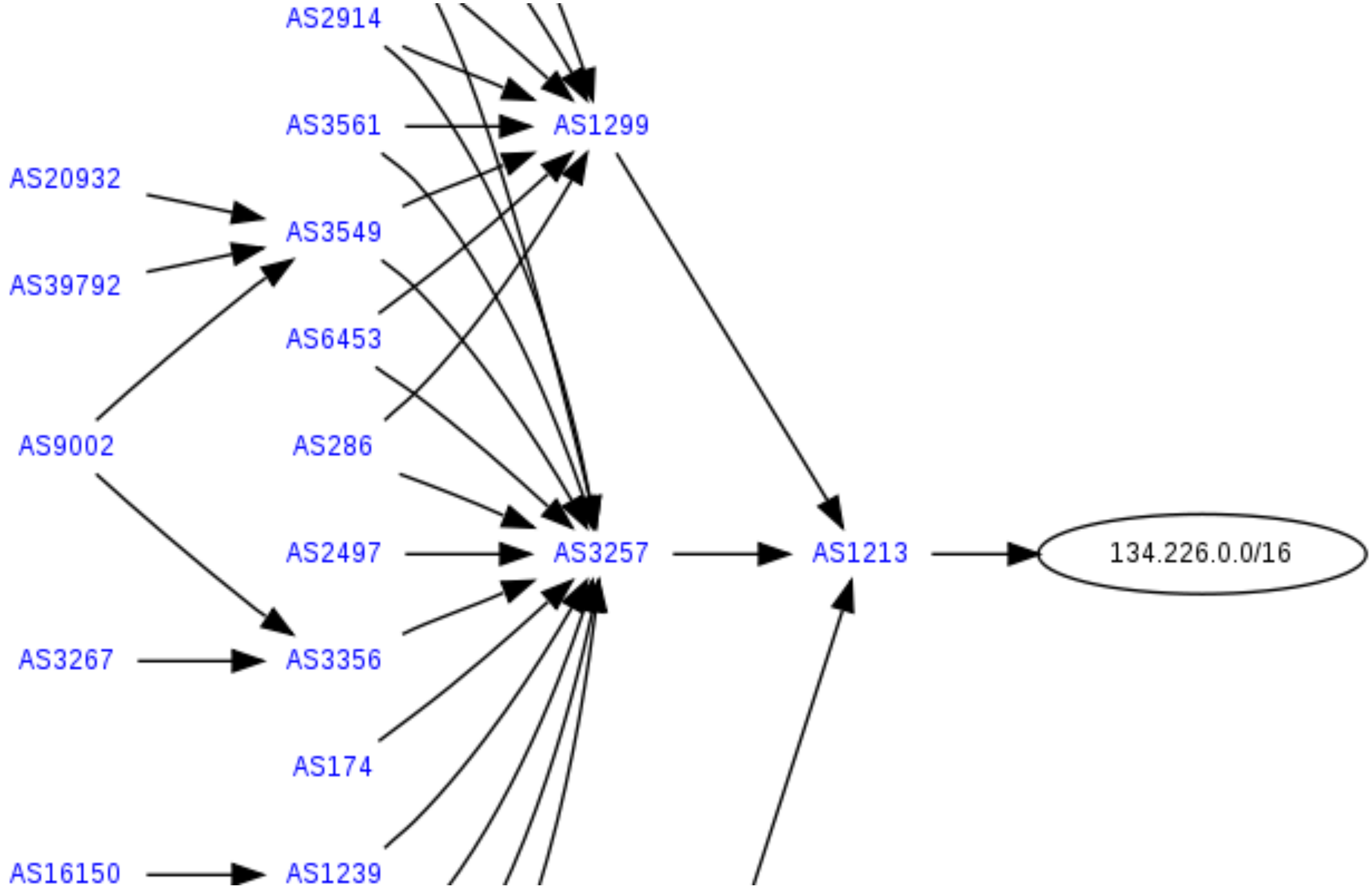
- Transient network
  - Network that forwards traffic between other networks

- Point-to-point link

AS1213 - HEANET



AS1213 - HEANET

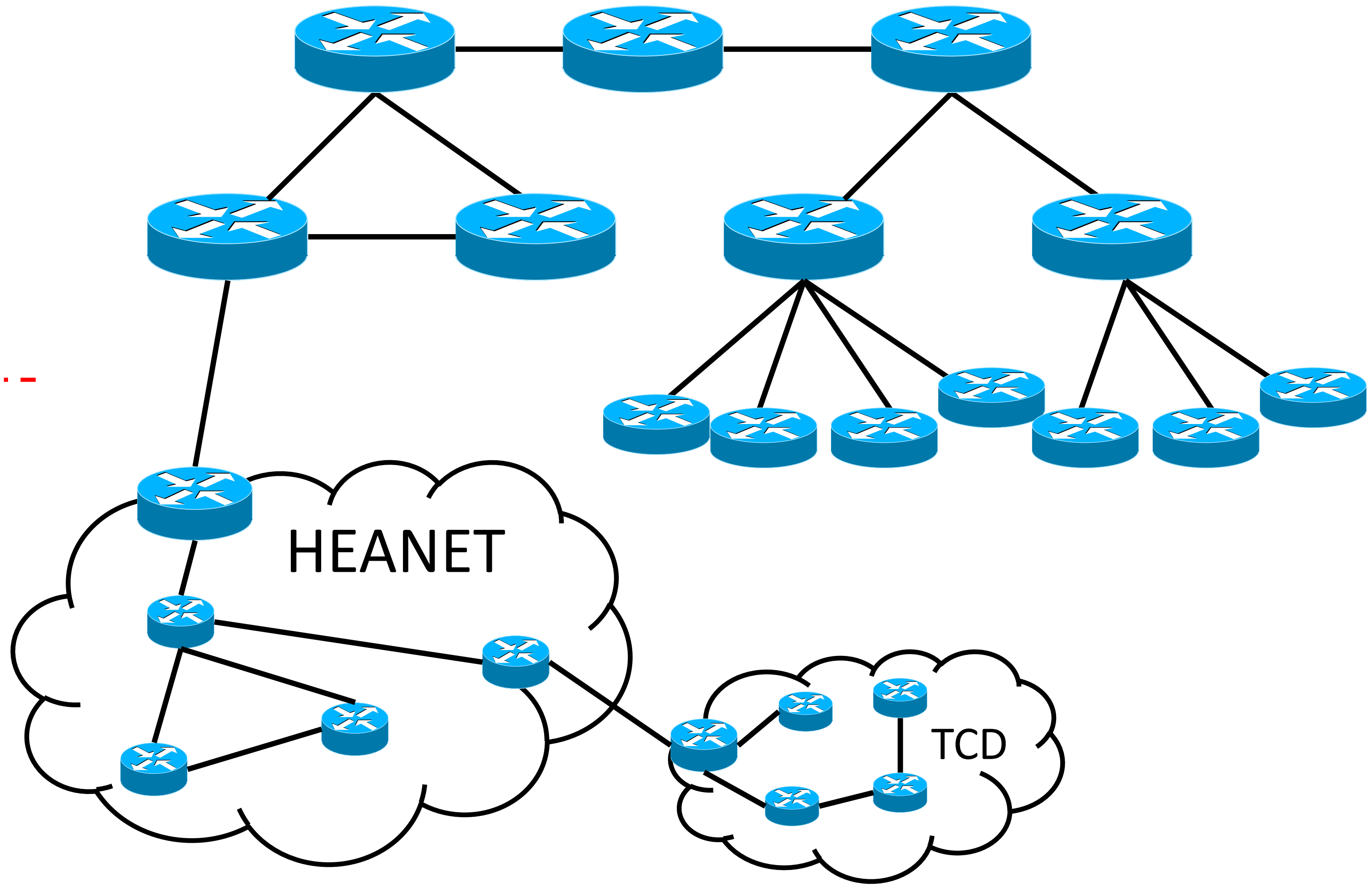


http://www.robtex.com/route/134.226.0.0-16.html

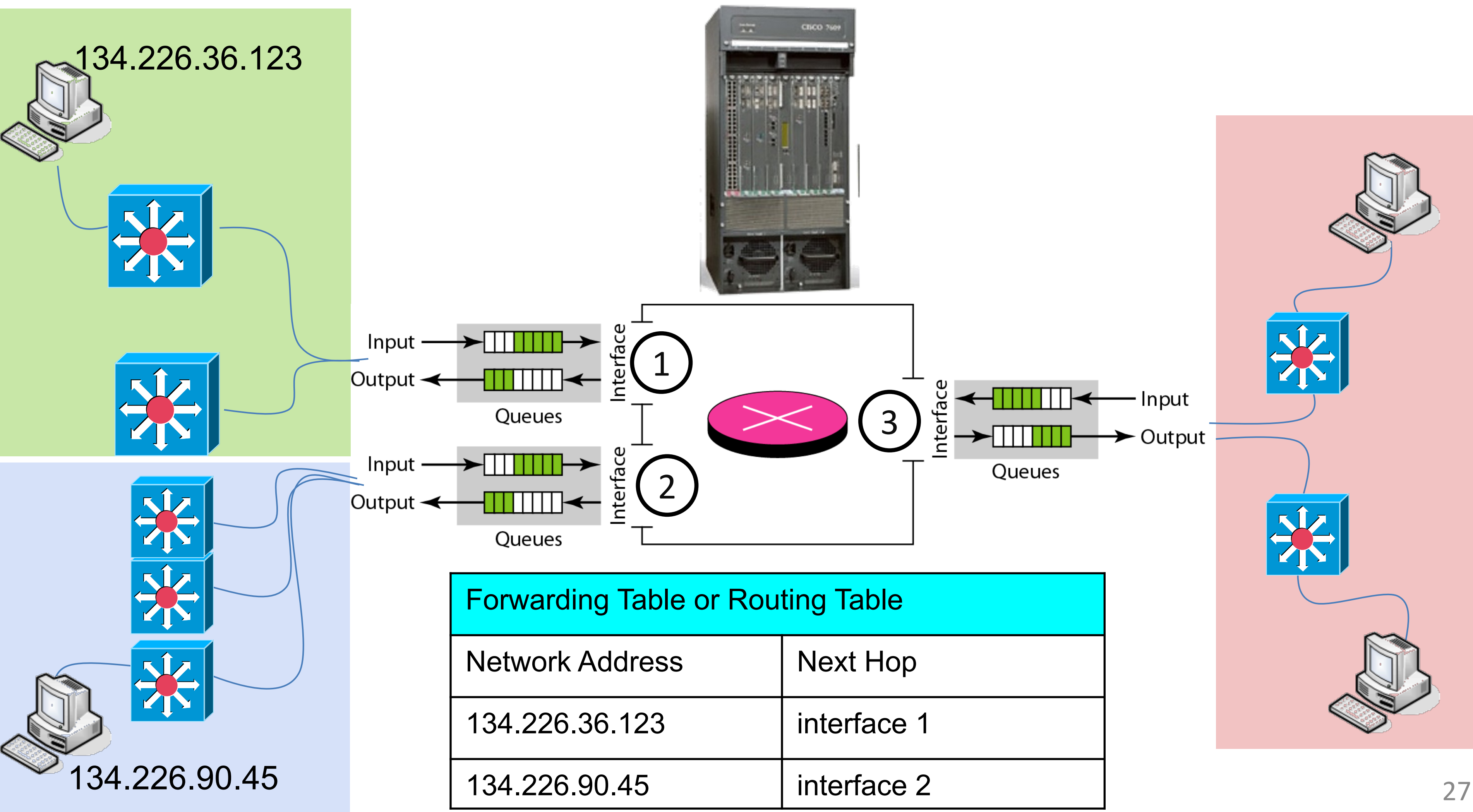
InterAS vs IntraAS Routing

InterAS Routing

IntraAS Routing



Routers (or Network Elements)



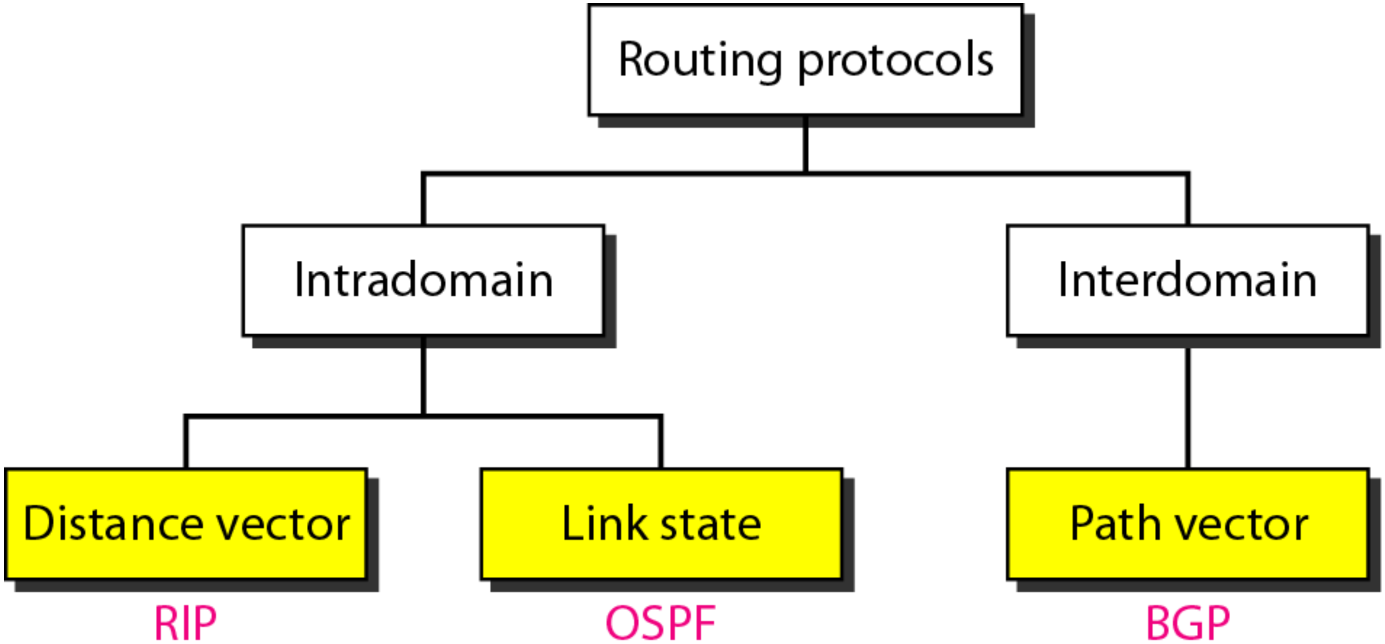
Forwarding Table or Routing Table	
Network Address	Next Hop
134.226.36.123	interface 1
134.226.90.45	interface 2

Everything's a Router

Active Routes:

Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.192.1	192.168.192.37	25
127.0.0.0	255.0.0.0	On-link	127.0.0.1	306
127.0.0.1	255.255.255.255	On-link	127.0.0.1	306
127.255.255.255	255.255.255.255	On-link	127.0.0.1	306
192.168.21.0	255.255.255.0	On-link	192.168.21.1	276
192.168.21.1	255.255.255.255	On-link	192.168.21.1	276
192.168.21.255	255.255.255.255	On-link	192.168.21.1	276
192.168.111.0	255.255.255.0	On-link	192.168.111.1	276
192.168.111.1	255.255.255.255	On-link	192.168.111.1	276
192.168.111.255	255.255.255.255	On-link	192.168.111.1	276
192.168.150.0	255.255.255.0	On-link	192.168.150.1	276
192.168.150.1	255.255.255.255	On-link	192.168.150.1	276
192.168.150.255	255.255.255.255	On-link	192.168.150.1	276
192.168.192.0	255.255.255.0	On-link	192.168.192.37	281
192.168.192.37	255.255.255.255	On-link	192.168.192.37	281
192.168.192.255	255.255.255.255	On-link	192.168.192.37	281

Routing Protocols



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

* Figure is courtesy of B. Forouzan

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

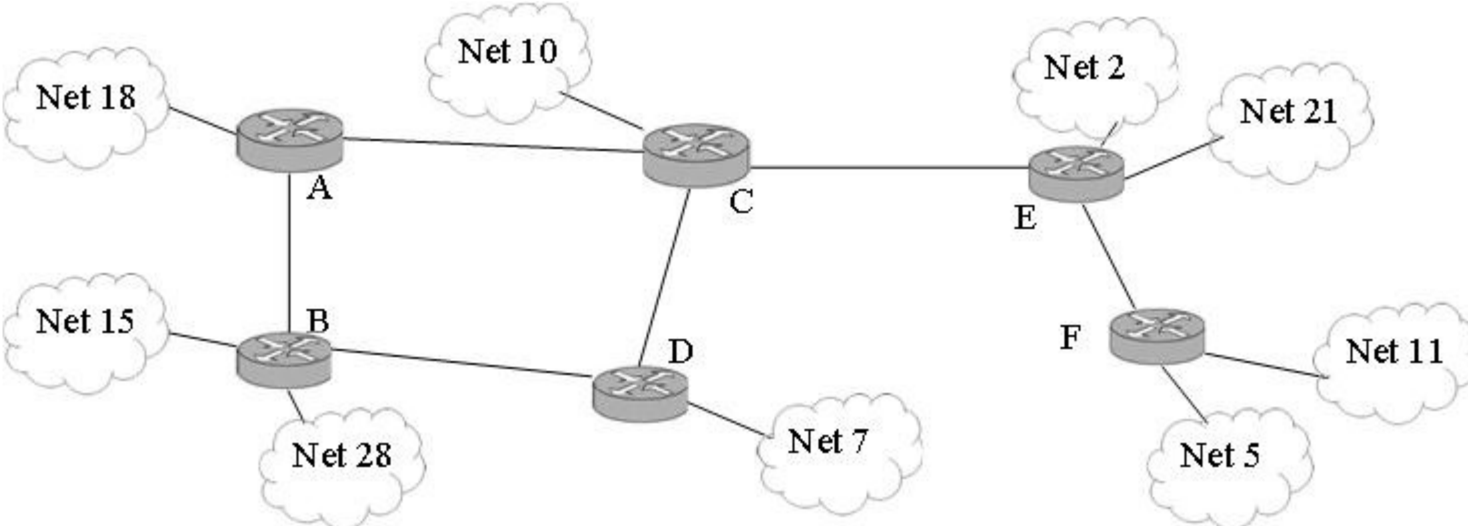
- Each node maintains a set of triples
  - **`(Destination, Cost, NextHop)`**

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	

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

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

Distance Vector: Example



A

Net18	1	-

B

Net15	1	-
Net28	1	-

C

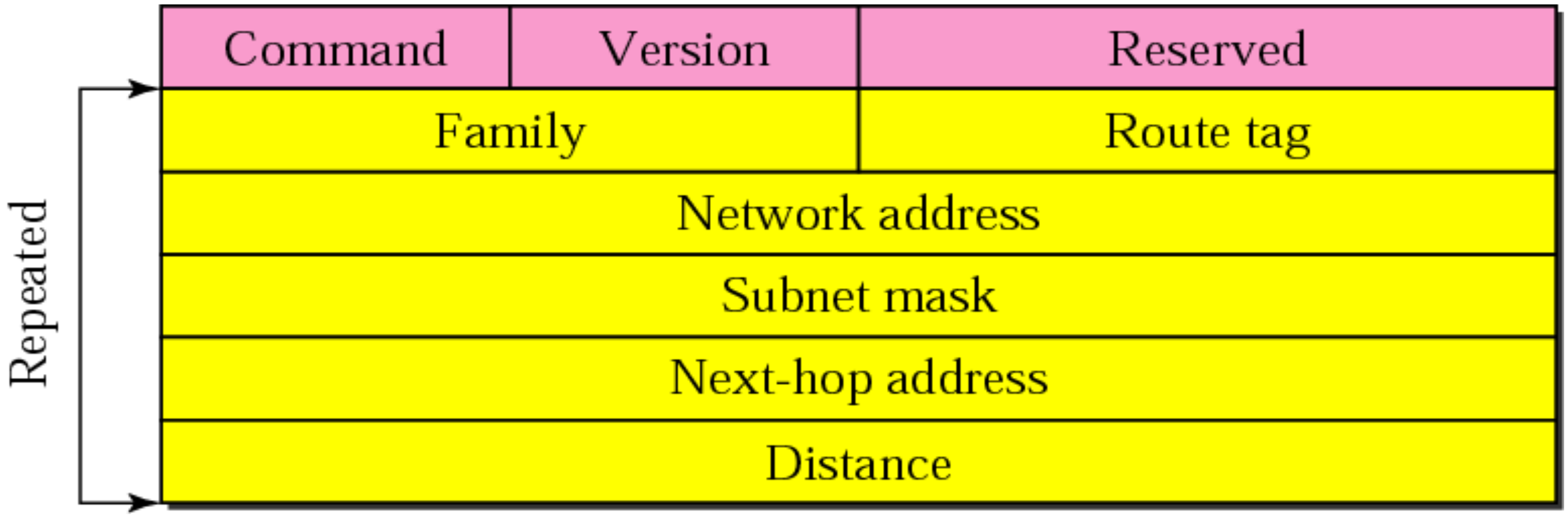
Net10	1	-

D

Net7	1	-

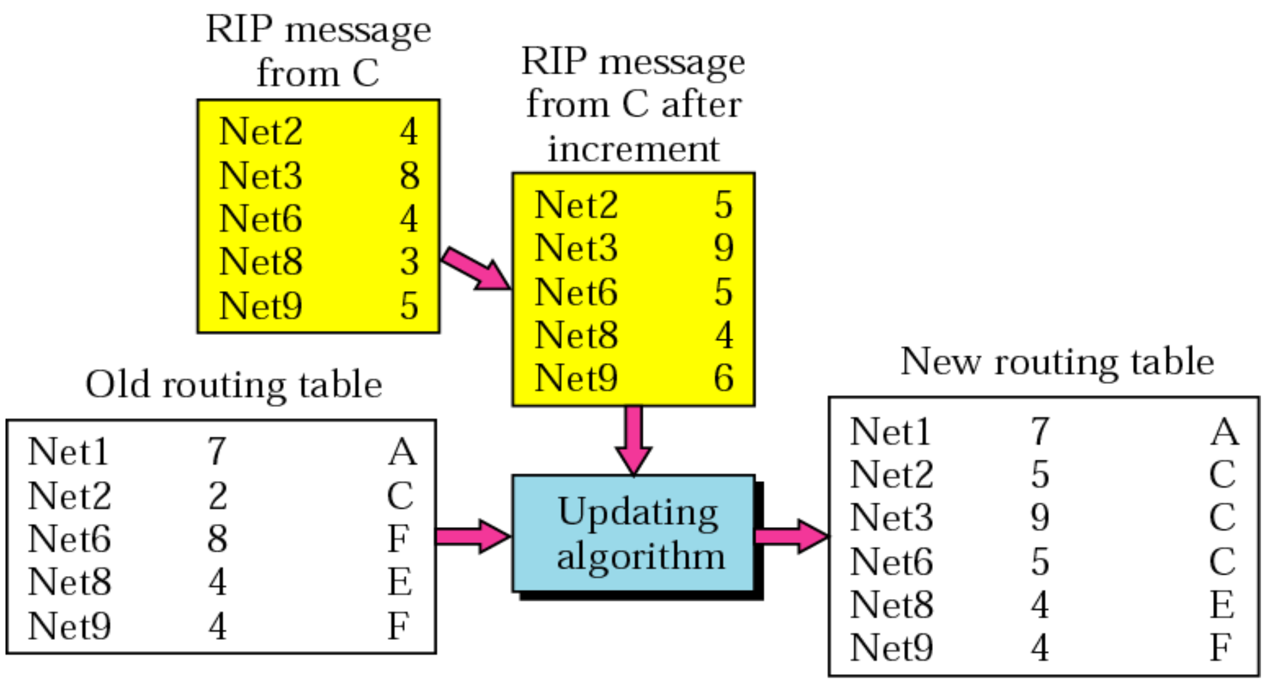
- Each router know the networks that are immediately connected to it

RIPv2 Message Format



* Figure is courtesy of B. Forouzan

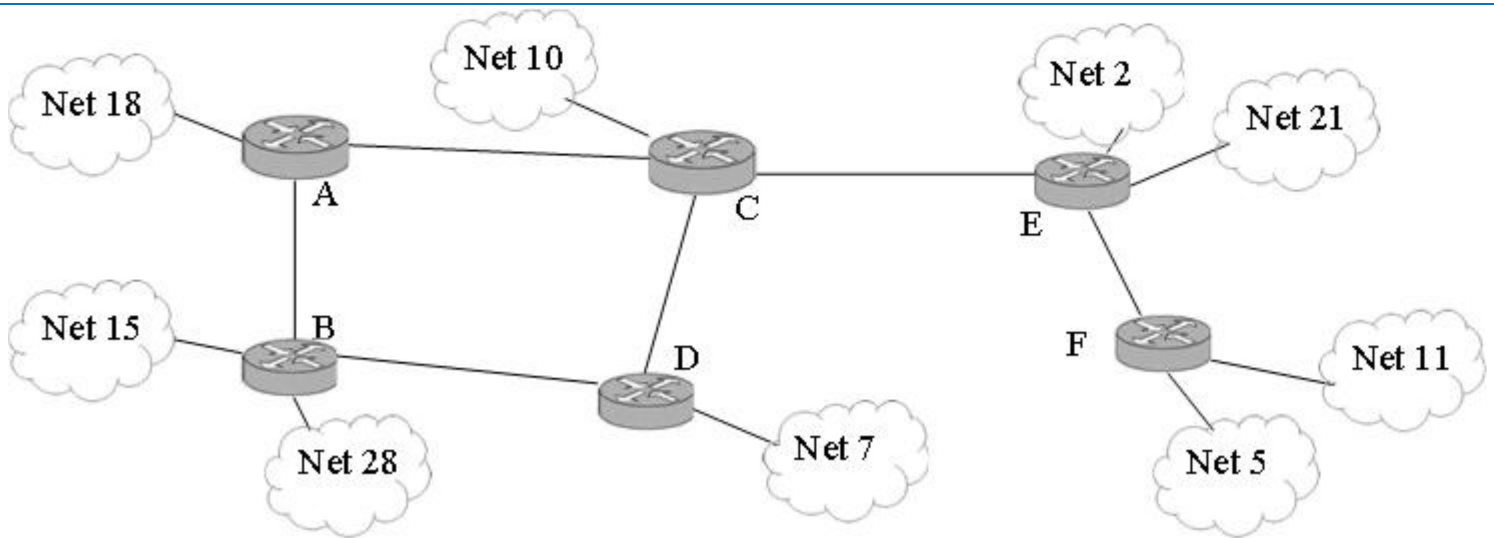
Receiving an Update



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

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

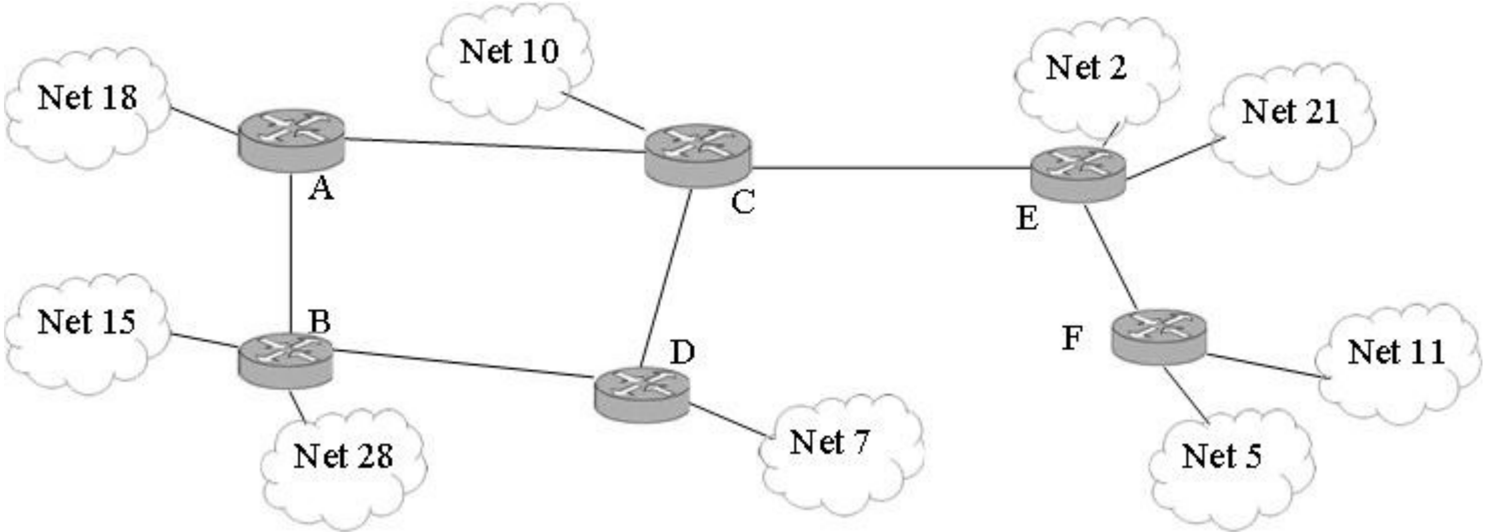
Net10	1	-
Net7	2	D
Net2	2	E
Net21	2	E

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

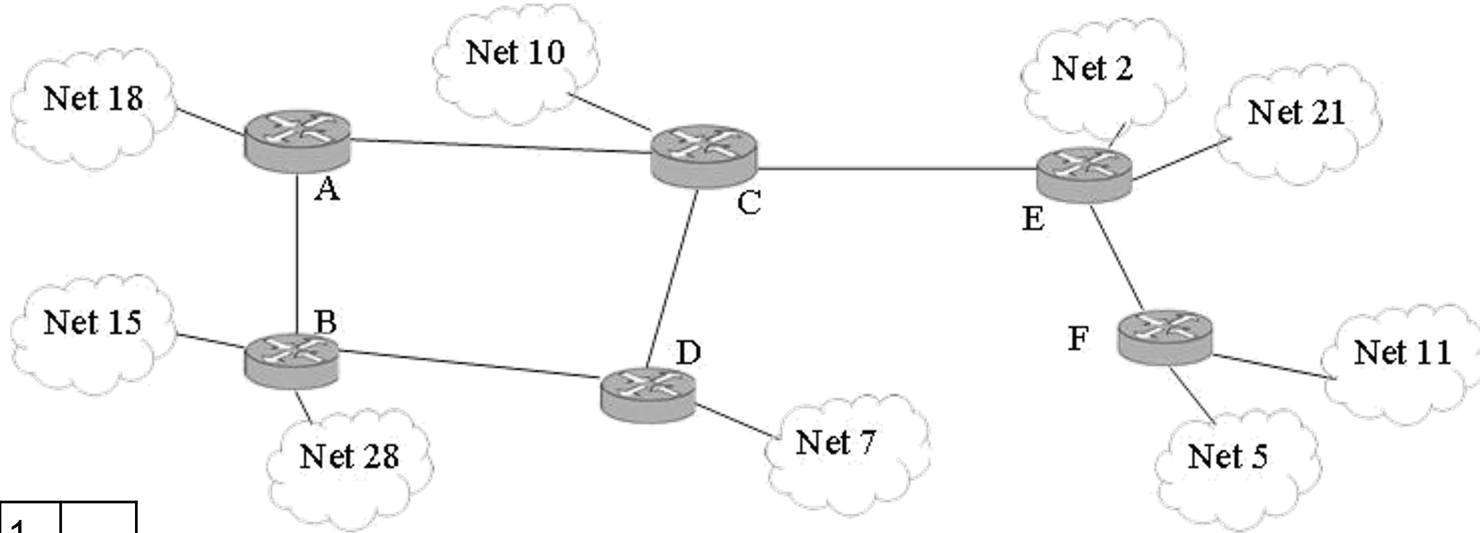
Net10	1	-
Net7	2	D
Net2	2	E
Net21	2	E
Net11	3	E
Net5	3	E

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

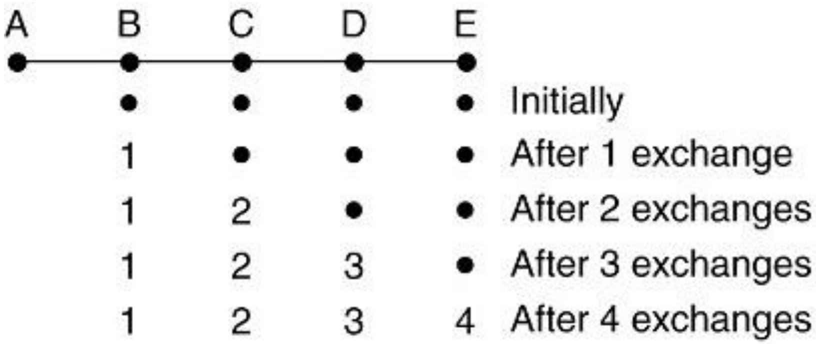


A

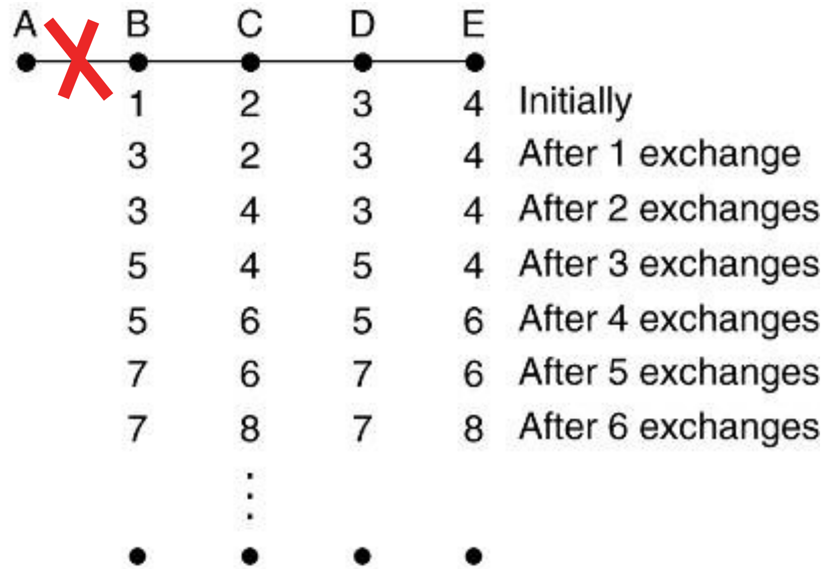
Net18	1	-
Net15	2	B
Net28	2	B
Net10	2	C
Net7	3	B
…	…	…
Net11	5	C
Net5	5	C

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

Count-to-Infinity Problem



(a)



(b)

- 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

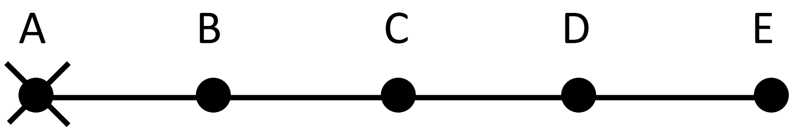
Solutions for Count-to-Infinity

- 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

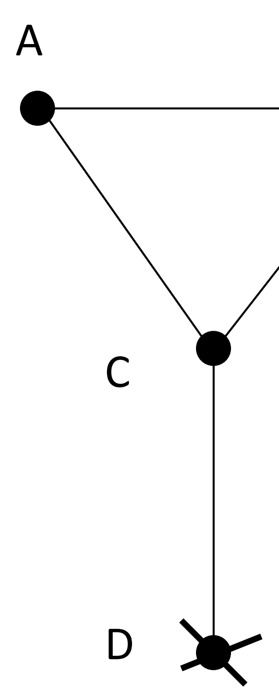
Report “split-horizon” routes as infinity to break loops on the first routing exchange.



inf.	2	3	4	B learns A is dead
inf.	$\xrightarrow{\text{inf.}}$ 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

Summary: Distance Vector R.

- Distance Vector routing
  - Share complete information with neighbours
  - Count-to-Infinity problem
  - Example: Routing Information Protocol (RIP)

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

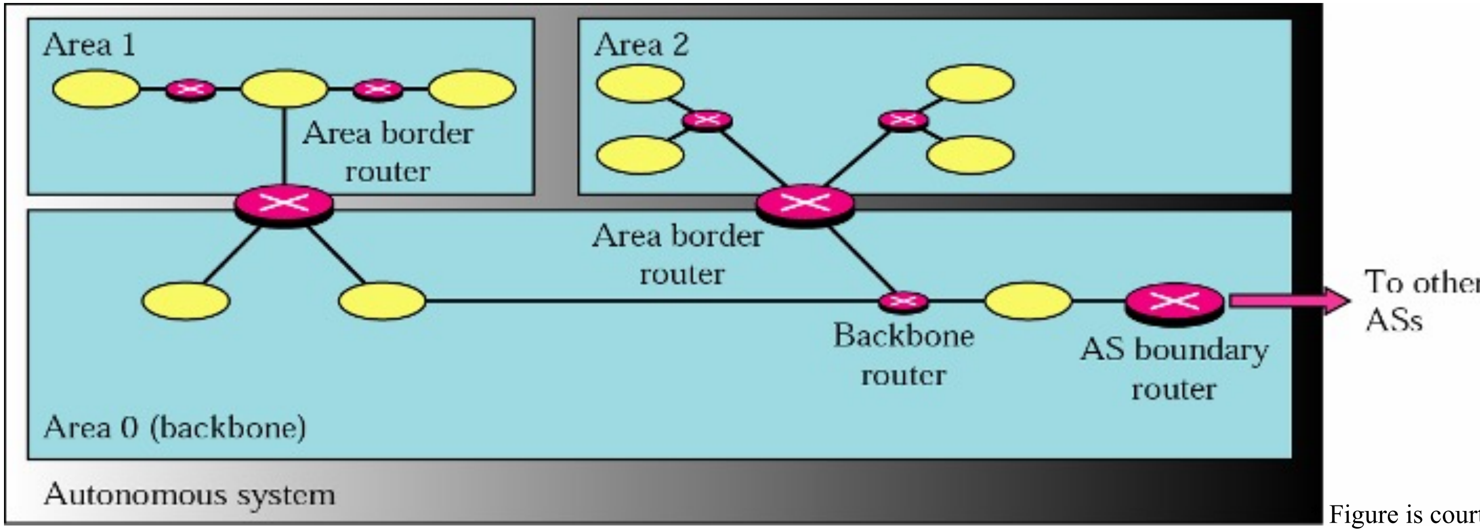
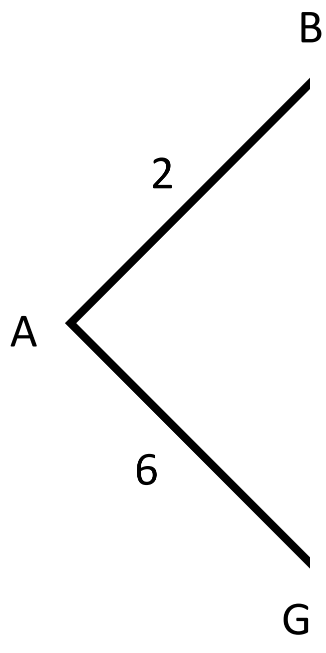


Figure is courtesy of B. Forouzan

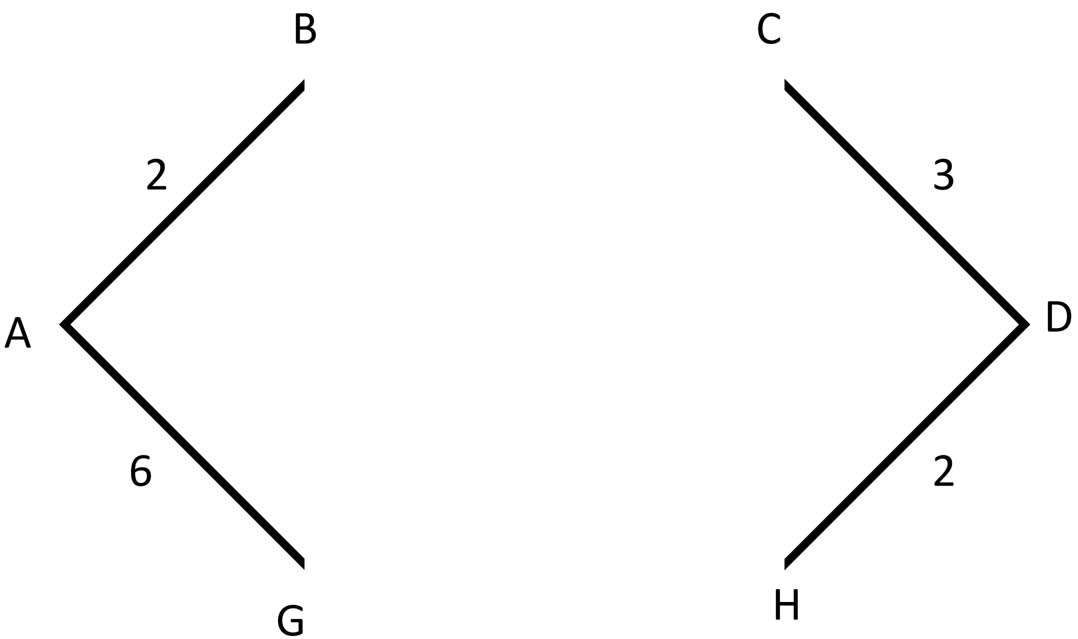
Example: Building a network graph

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



Example: Building a network graph

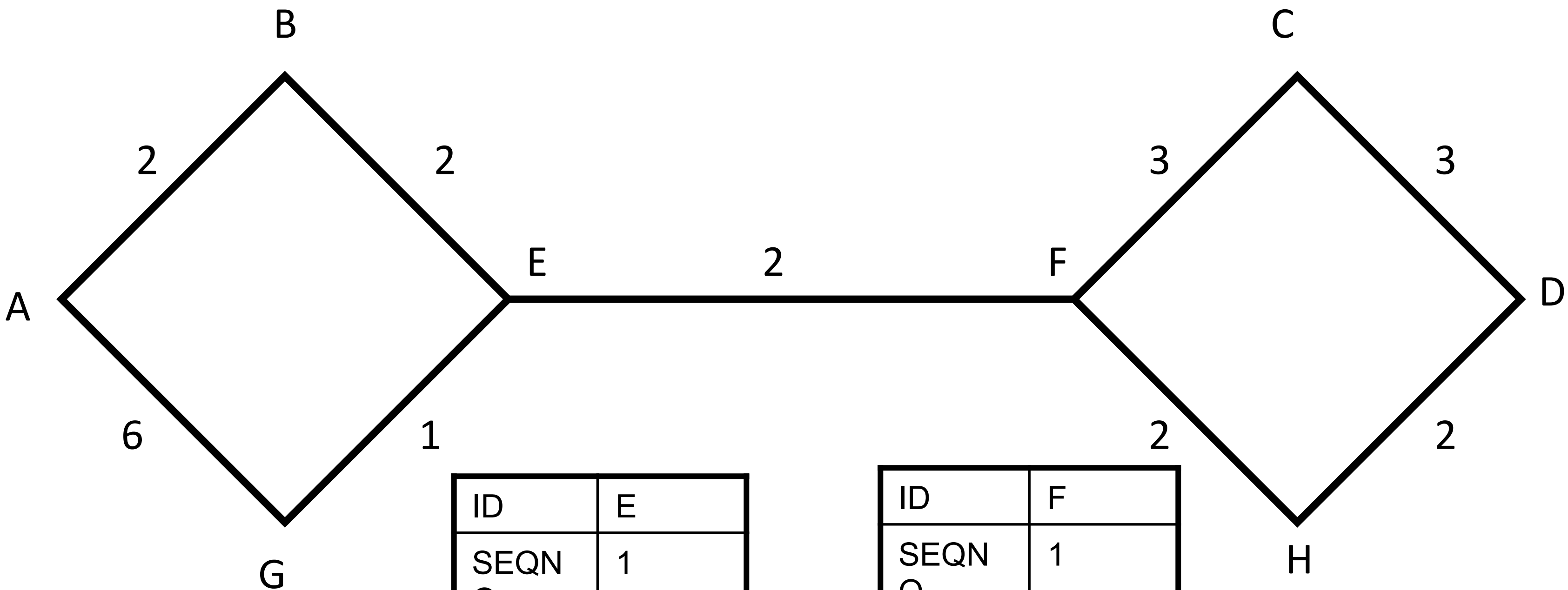
A receives an update from D



ID	D
SEQN O	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

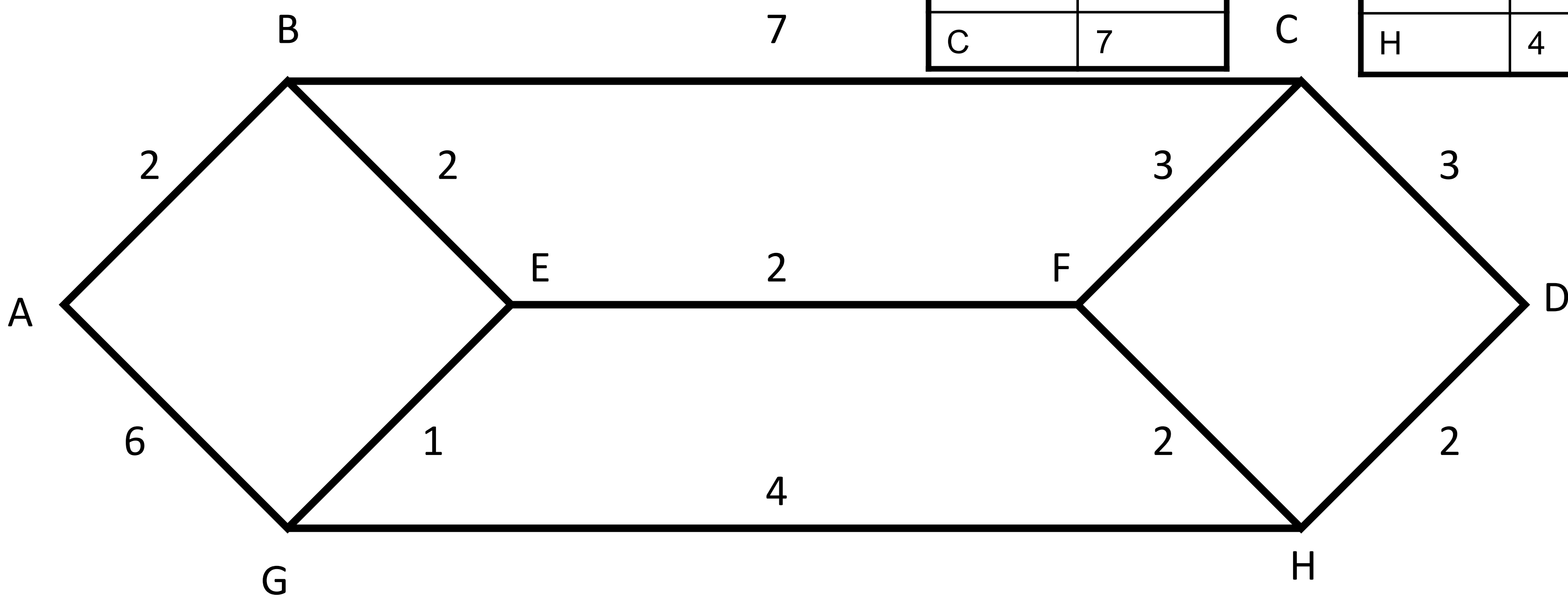
ID	F
SEQNO	1
TTL	21
C	3
H	2
E	2

Example: Building a network graph

A receives updates from B and G

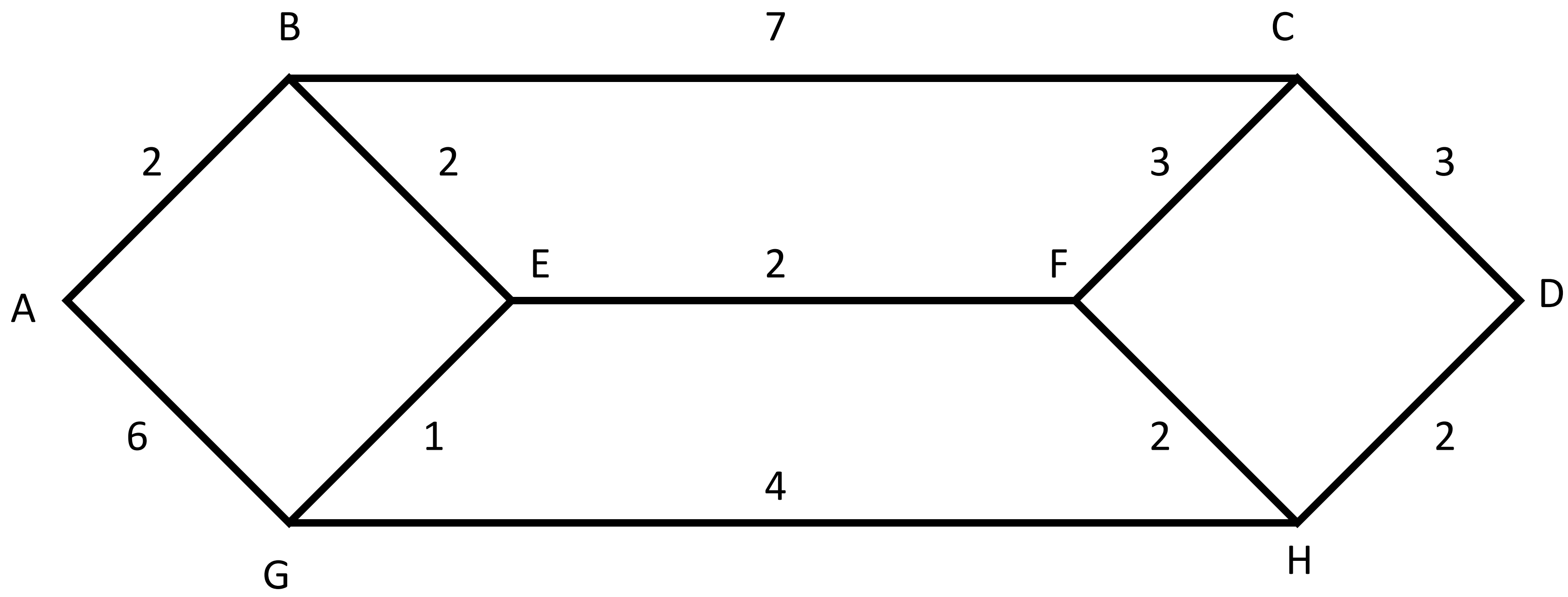
ID	B
SEQNO	2
TTL	23
A	2
E	2
C	7

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

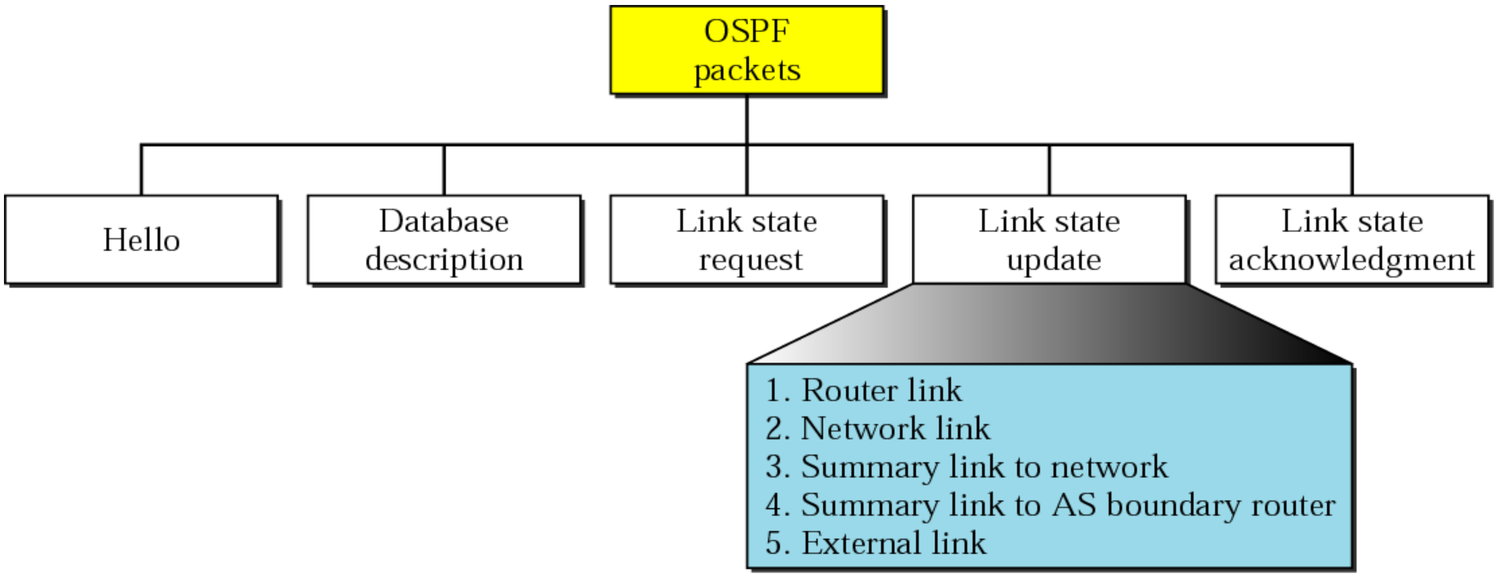


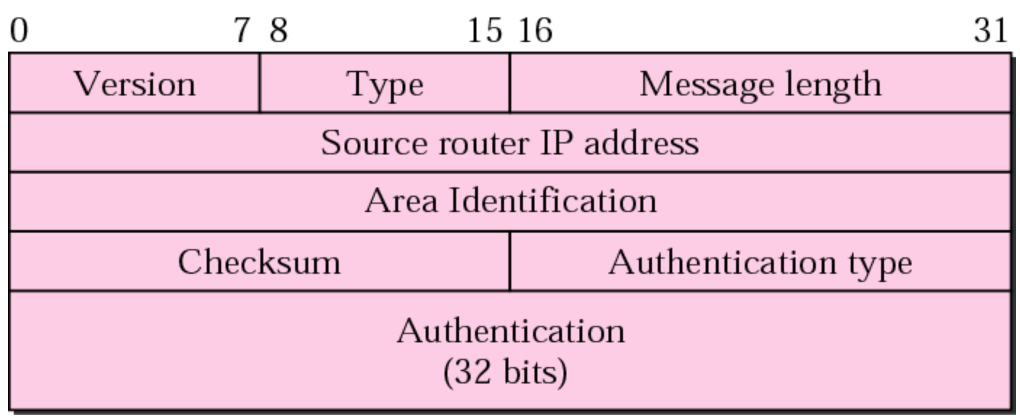
Example: Building a network graph

Final graph



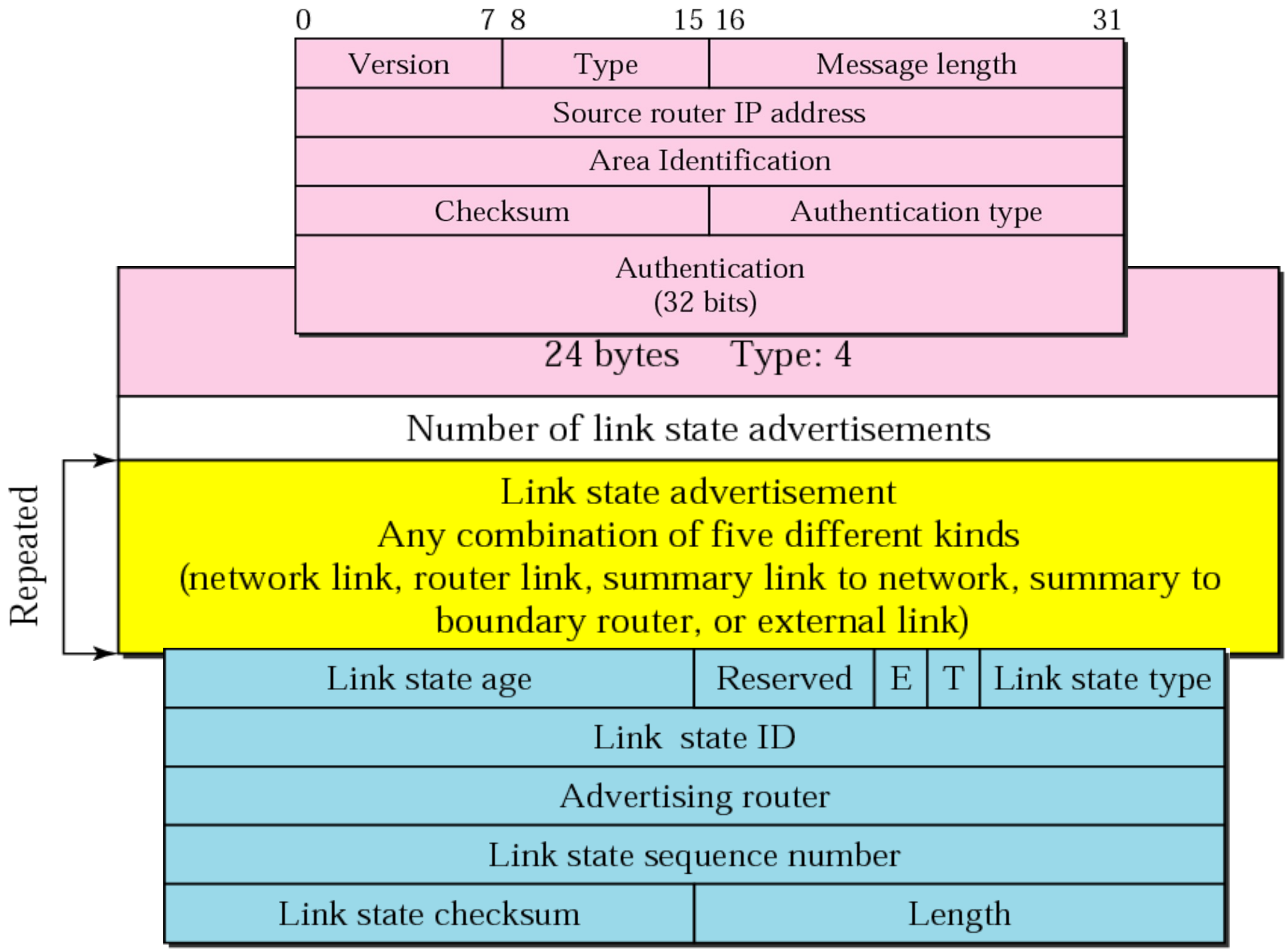
OSPF Packet Types





* Figure is courtesy of B. Forouzan

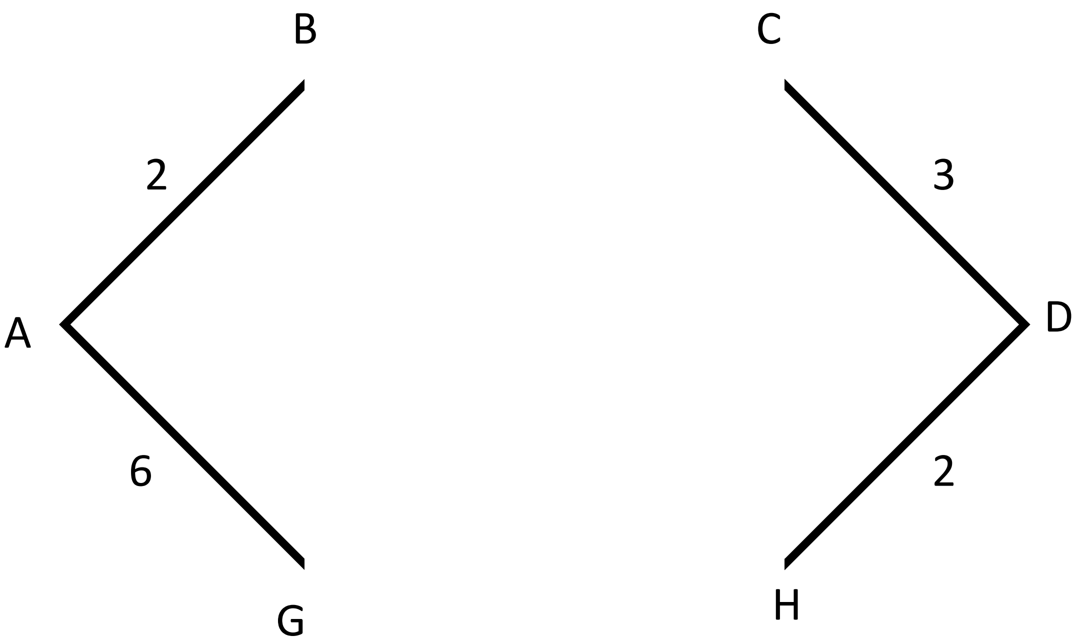
OSPF Link State Advertisement



* Figure is courtesy of B. Forouzan

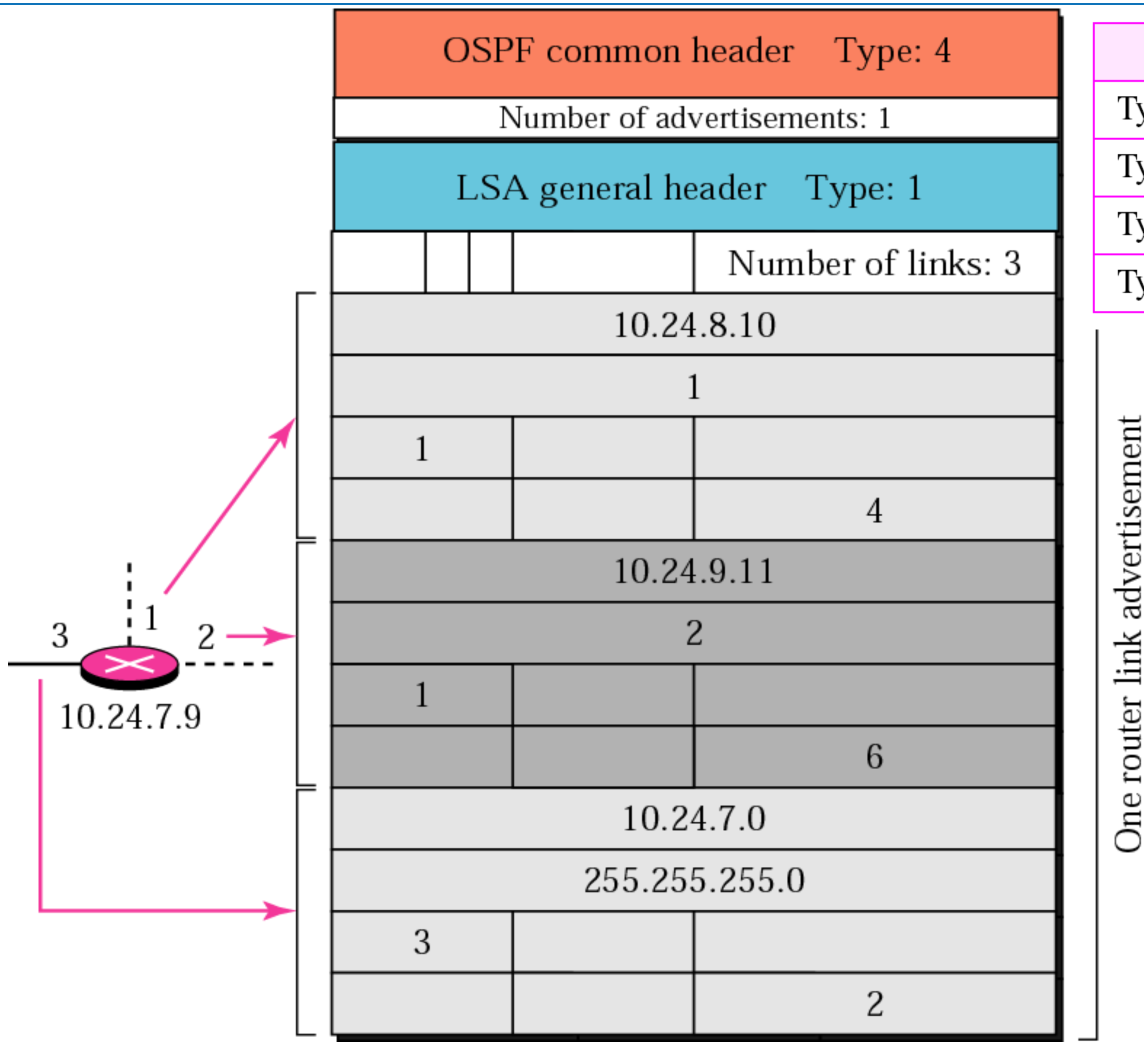
Example: Building a network graph

A receives an update from D



ID	D
SEQNO	0
TTL	21
C	3
H	2

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

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

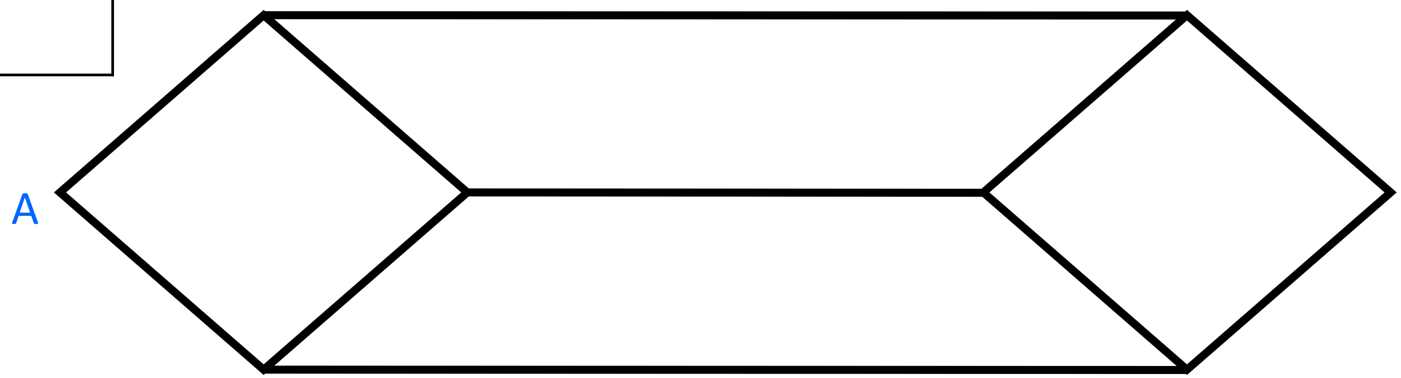
1. Start with the local node (router): the root of the tree.
2. Assign a cost of 0 to this node and make it the first permanent node.
3. Examine each neighbour node of the node that was the last permanent node.
4. Assign a cumulative cost to each node and make it tentative.
5. Among the list of tentative 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	-



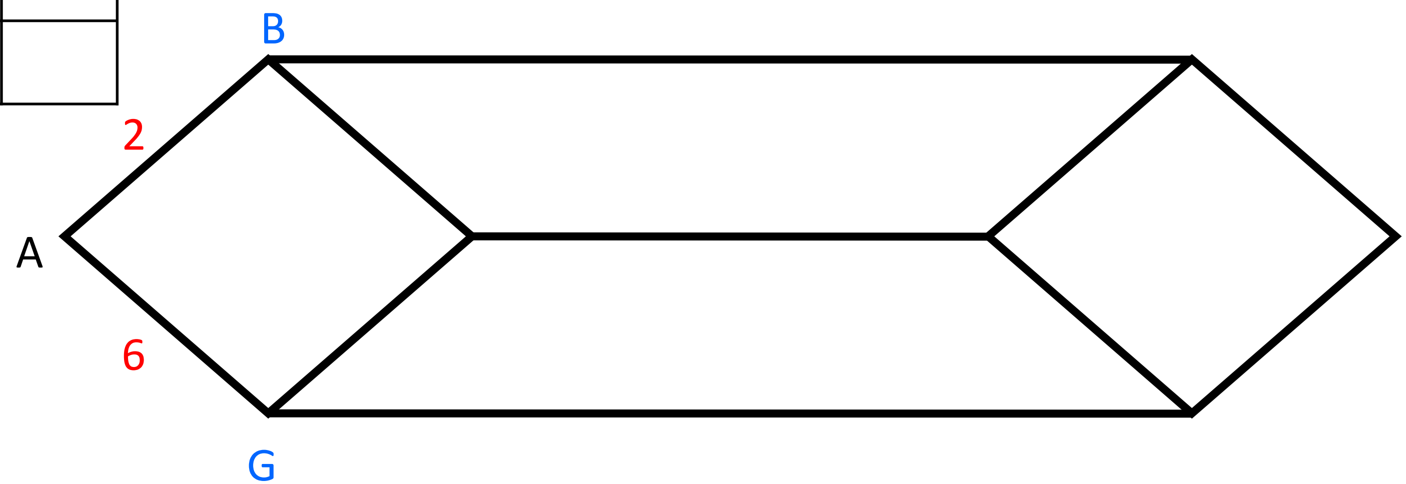
Example: Dijkstra's Algorithm

Routing Table

A	0	-

Tentative Nodes

B	2	A
G	6	A



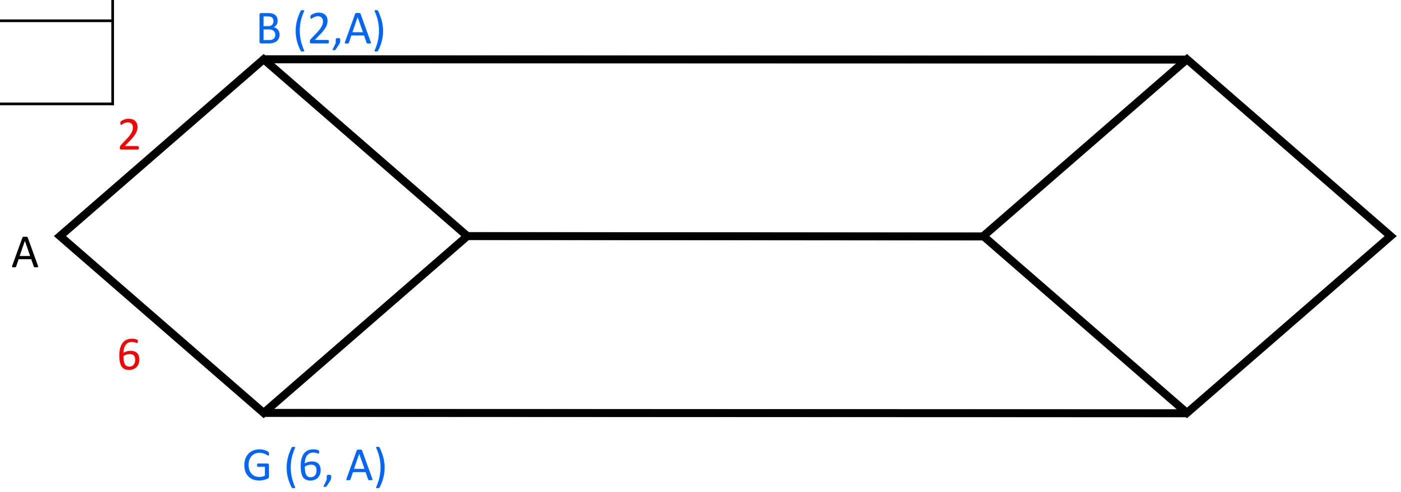
Example: Dijkstra's Algorithm

Routing Table

A	0	-
B	2	A

Tentative Nodes

G	6	A



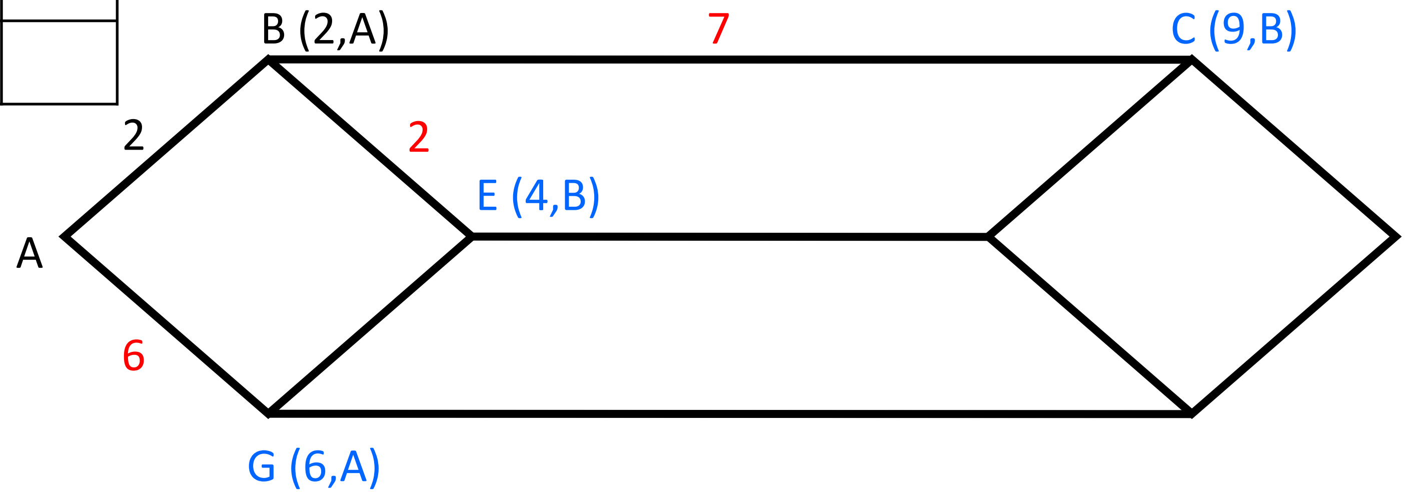
Example: Dijkstra's Algorithm

Routing Table

A	0	-
B	2	A

Tentative Nodes

G	6	A
E	4	B
C	9	B



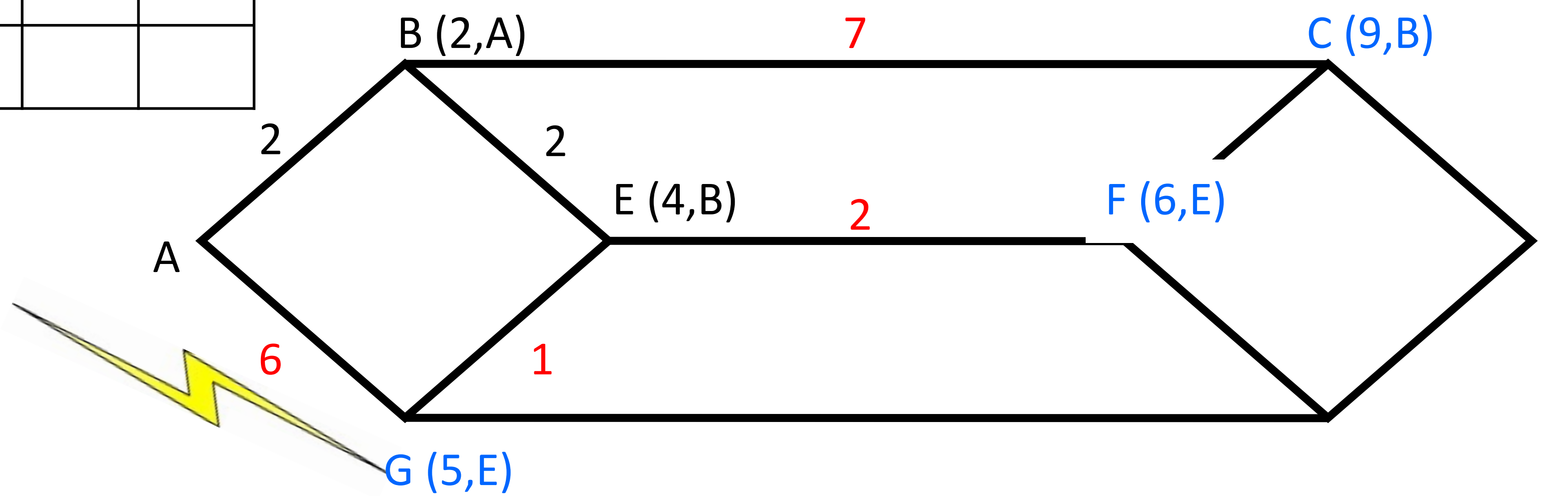
Example: Dijkstra's Algorithm

Routing Table

A	0	-
B	2	A
E	4	B

Tentative Nodes

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



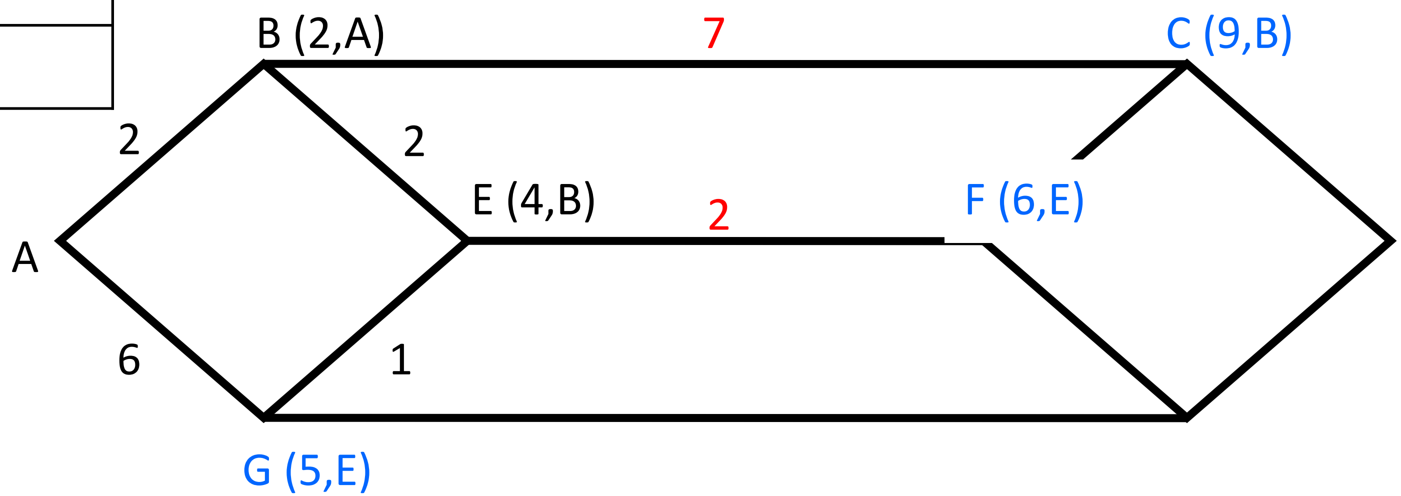
Example: Dijkstra’s Algorithm

Routing Table

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

Tentative Nodes

C	9	B
F	6	E



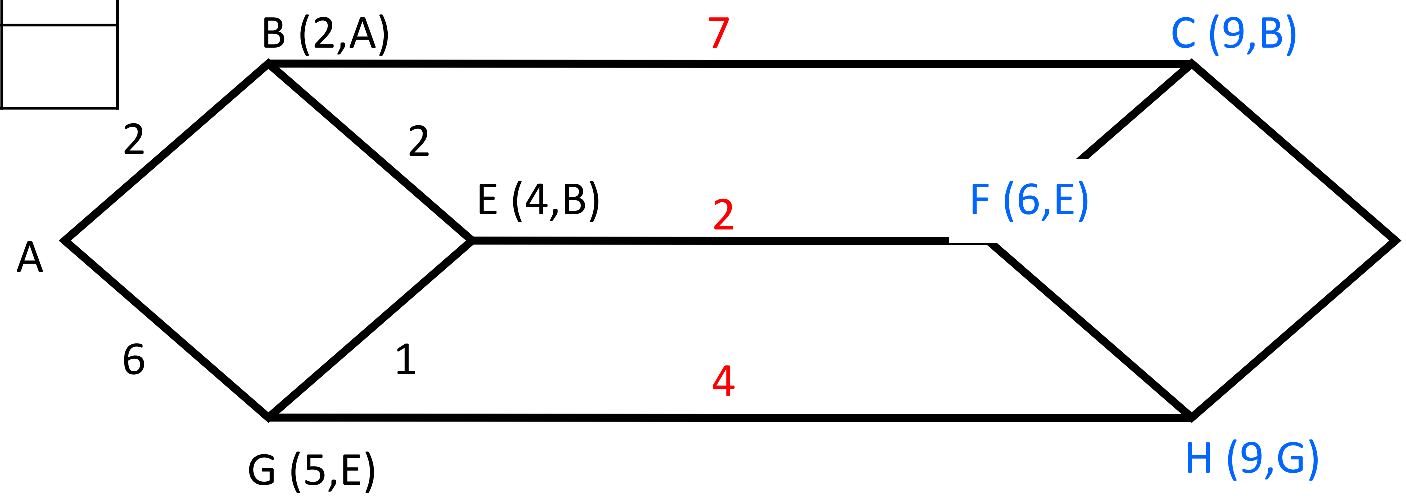
Example: Dijkstra’s Algorithm

Routing Table

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

Tentative Nodes

C	9	B
H	9	G
F	6	E



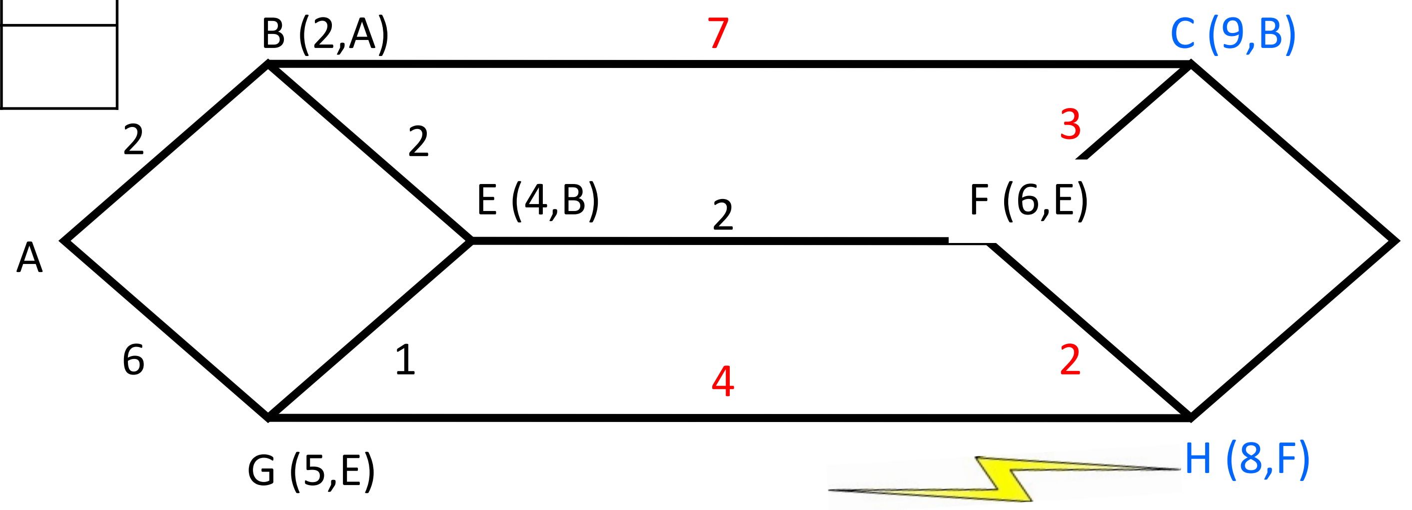
Example: Dijkstra’s Algorithm

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



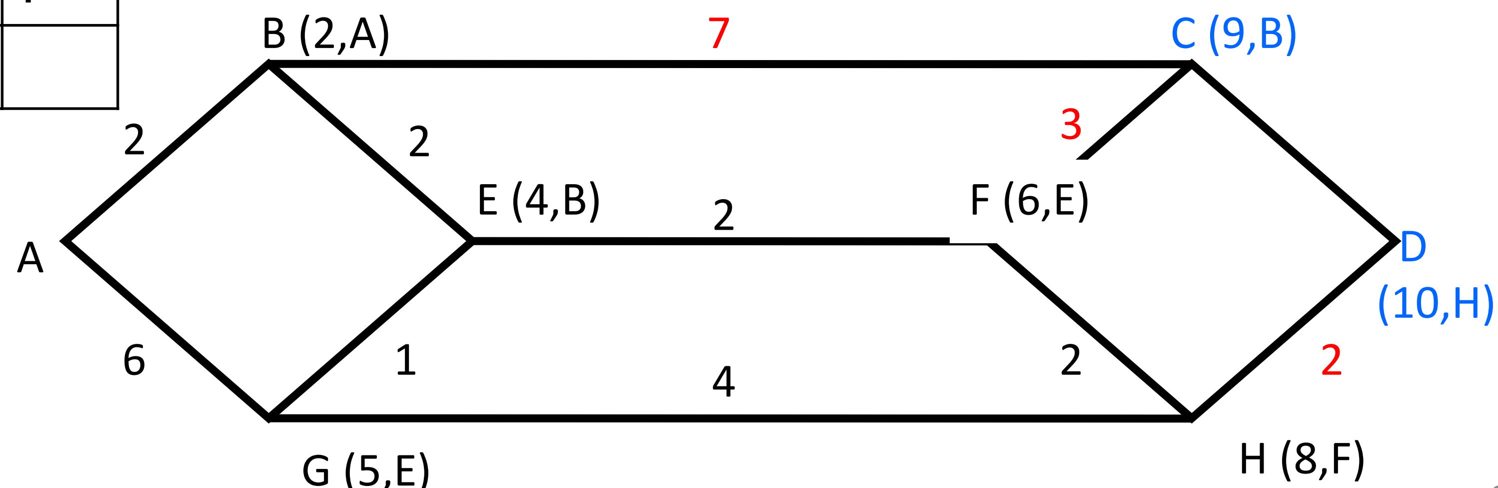
Example: Dijkstra's Algorithm

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



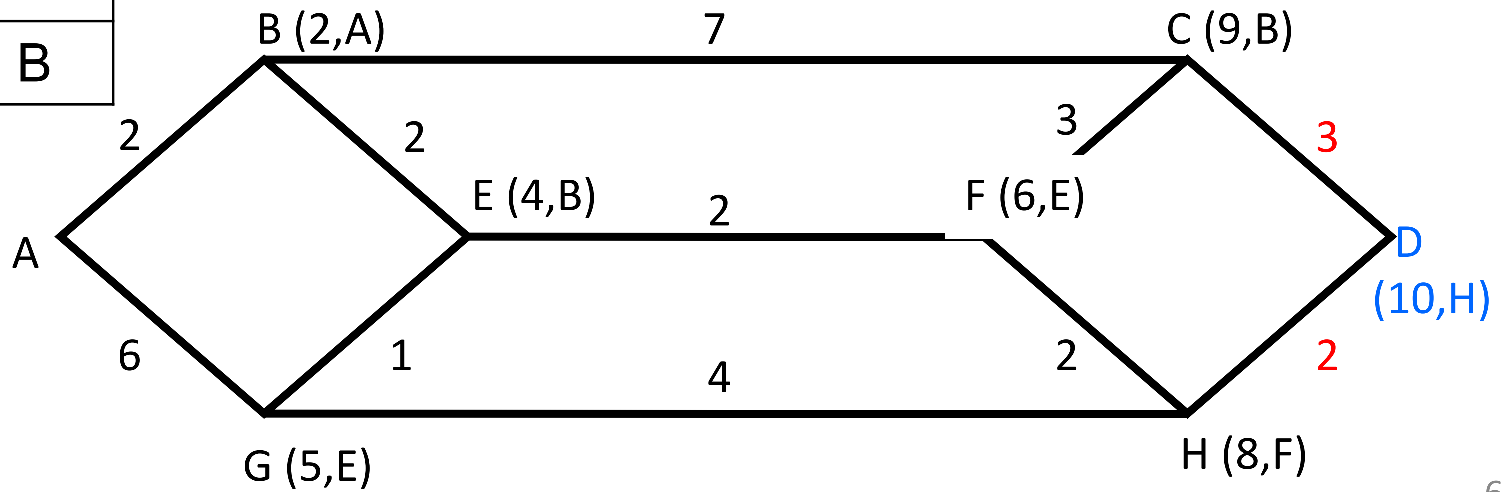
Example: Dijkstra's Algorithm

Routing Table

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

Tentative Nodes

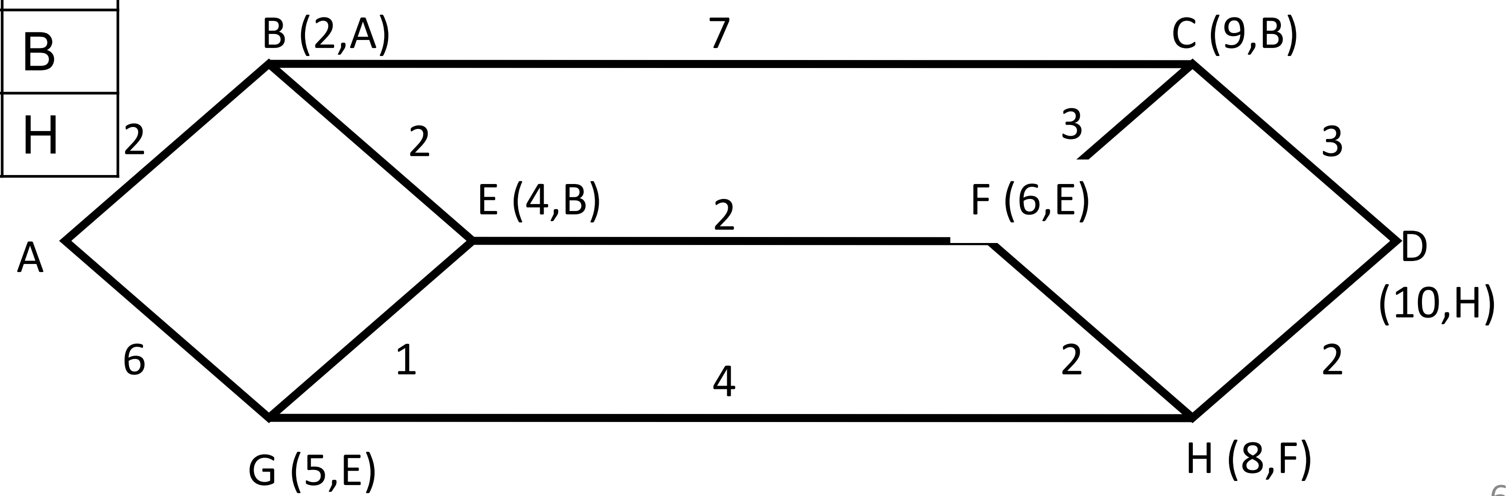
~~D~~	~~12~~	~~C~~
D	10	H



Example: Dijkstra’s Algorithm

Routing Table

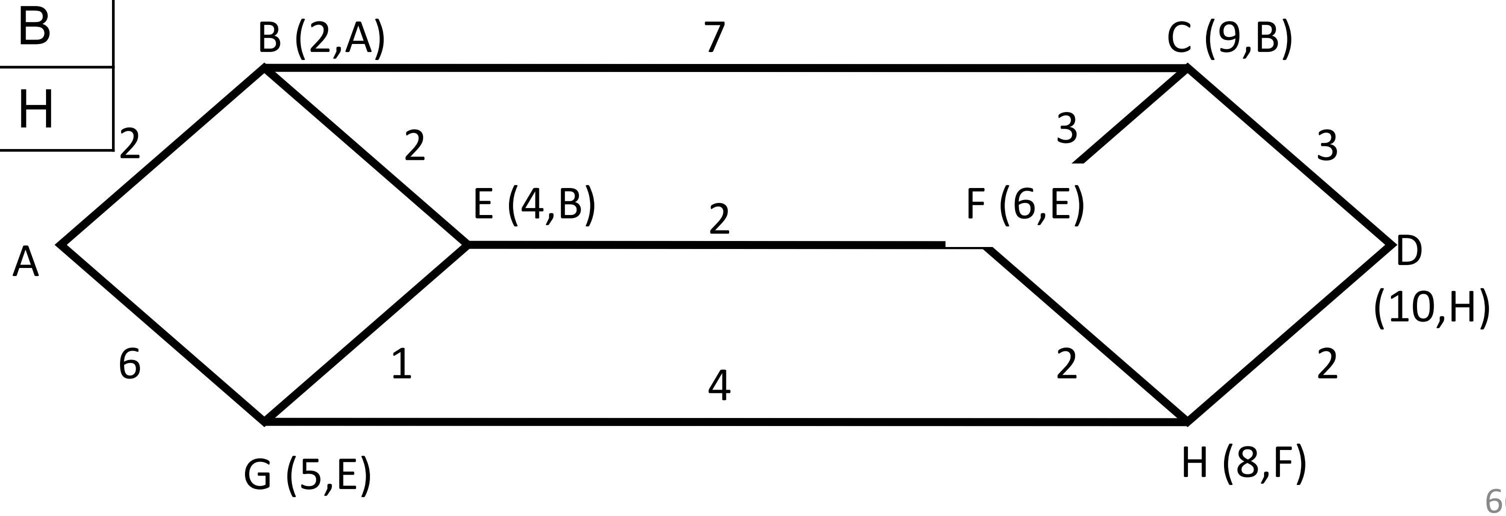
A	0	-
B	2	A
E	4	B
G	5	E
F	6	E
H	8	F
C	9	B
D	10	H



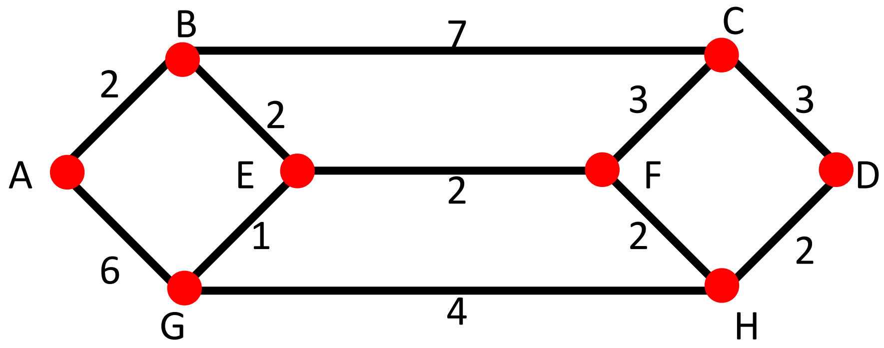
Example: Dijkstra’s Algorithm

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

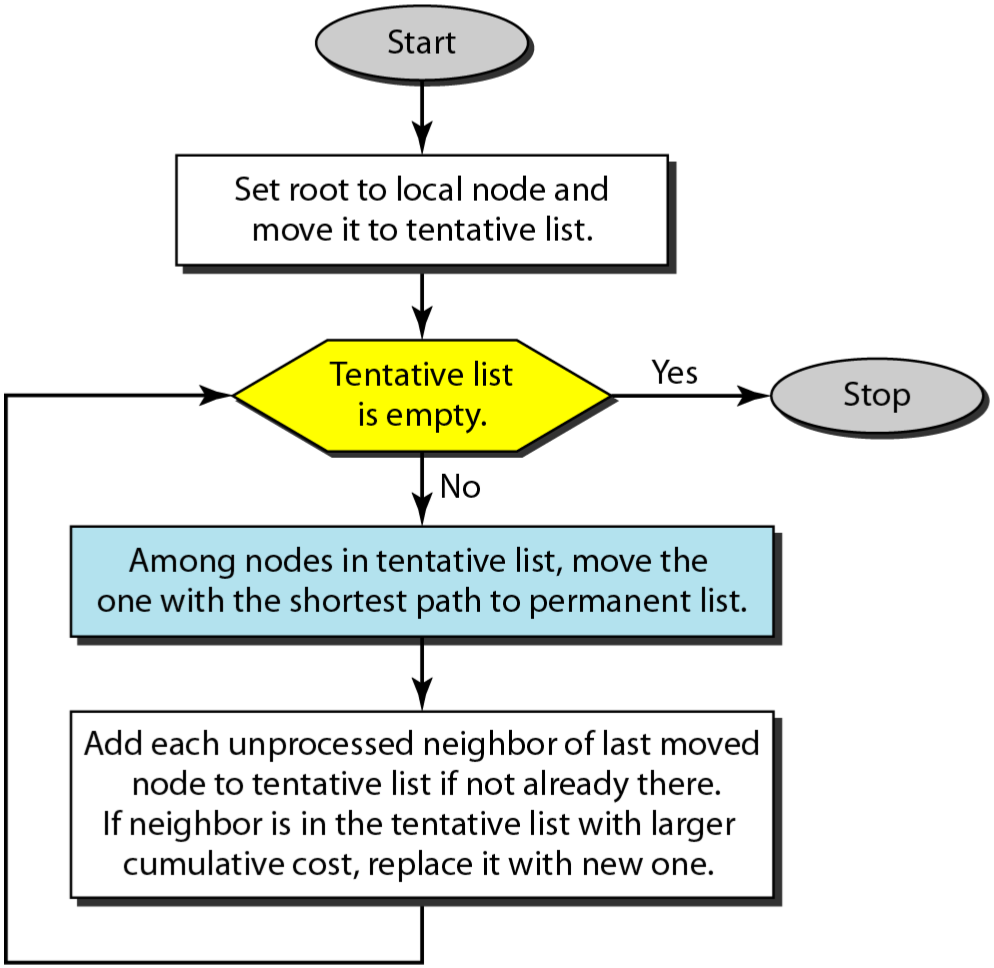
bag = bag of destinations to be resolved

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

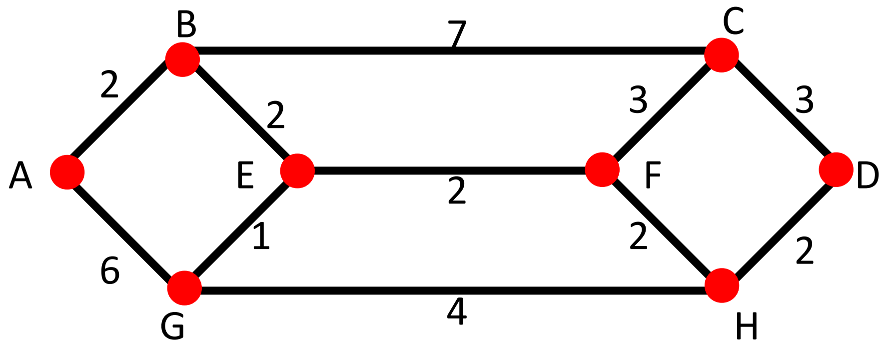
- As long as there are destinations in the bag
  - Take the destination with the smallest distance out of the bag and put it into the table (if the distance is smaller than an already existing entry for the same destination)
  - Put all the neighbours of that destination with their distance to the origin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

Djikstra's Shortest Path



* Figure is courtesy of B. Forouzan

Link State Advertisements



ID	A
SEQ	2
B	2
G	6

ID	B
SEQ	2
A	2
E	2
C	7

ID	C
SEQ	0
B	7
D	3
F	3

ID	D
SEQ	0
C	3
H	2

ID	E
SEQ	1
B	2
G	1
F	2

ID	F
SEQ	1
C	3
H	2
E	2

ID	G
SEQ	3
A	6
E	1
H	4

ID	H
SEQ	3
G	4
F	2
D	2

Processing Example – Node A

Dest	Cost	Path
A	0	-

Processing Example – Node A

Dest	Cost	Path
A	0	-
B	2	A

ID	A
SEQ	2
B	2
G	6

Processing Example – Node A

Dest	Cost	Path
A	0	-
B	2	A

ID	A
SEQ	2
G	6

ID	B
SEQ	2
A	2
E	2(+2)
C	7(+2)

Processing Example – Node A

Dest	Cost	Path
A	0	-
B	2	A
E	4	AB

ID	A
SEQ	2
G	6

ID	B
SEQ	2
C	9

ID	E
SEQ	1
B	2
G	1(+4)
F	2(+4)

Processing Example – Node A

Dest	Cost	Path
A	0	-
B	2	A
E	4	AB
G	5	ABE

ID	A
SEQ	2

ID	B
SEQ	2
C	9

ID	E
SEQ	1
B	2
G	1(+4)
F	2(+4)

ID	G
SEQ	3
A	6
E	1
H	4(+5)

Processing Example – Node A

Dest	Cost	Path
A	0	-
B	2	A
E	4	AB
G	5	ABE

ID	A
SEQ	2

ID	B
SEQ	2
C	9

ID	E
SEQ	1
F	6

ID	G
SEQ	3
H	9

Processing Example – Node A

Dest	Cost	Path
A	0	-
B	2	A
E	4	AB
G	5	ABE
F	6	ABE

ID	A
SEQ	2

ID	B
SEQ	2
C	9

ID	E
SEQ	1
F	6

ID	G
SEQ	3
H	9

ID	F
SEQ	1
C	3(+6)
H	2(+6)
E	2

Processing Example – Node A

Dest	Cost	Path
A	0	-
B	2	A
E	4	AB
G	5	ABE
F	6	ABE
H	8	ABEF

ID	A
SEQ	2

ID	B
SEQ	2
C	9

ID	E
SEQ	1

ID	G
SEQ	3

ID	F
SEQ	1

ID	H
SEQ	3
G	4
F	2
D	2(+8)

Processing Example – Node A

Dest	Cost	Path
A	0	-
B	2	A
E	4	AB
G	5	ABE
F	6	ABE
H	8	ABEF
C	9	AB

ID	A
SEQ	2

ID	B
SEQ	2

ID	E
SEQ	1

ID	G
SEQ	3

ID	F
SEQ	1

ID	H
SEQ	3
D	10

ID	C
SEQ	0
B	7
D	3(+9)
F	3

Processing Example – Node A

Dest	Cost	Path
A	0	-
B	2	A
E	4	AB
G	5	ABE
F	6	ABE
H	8	ABEF
C	9	AB
D	10	ABEFH

ID	D
SEQ	0
C	3
H	2

ID	A
SEQ	2

ID	B
SEQ	2

ID	E
SEQ	1

ID	G
SEQ	3

ID	F
SEQ	1

ID	H
SEQ	3
D	10

ID	C
SEQ	0

Processing Example – Node A

Dest	Cost	Path
A	0	-
B	2	A
E	4	AB
G	5	ABE
F	6	ABE
H	8	ABEF
C	9	AB
D	10	ABEFH

ID	D
SEQ	0

ID	A
SEQ	2

ID	B
SEQ	2

ID	E
SEQ	1

ID	G
SEQ	3

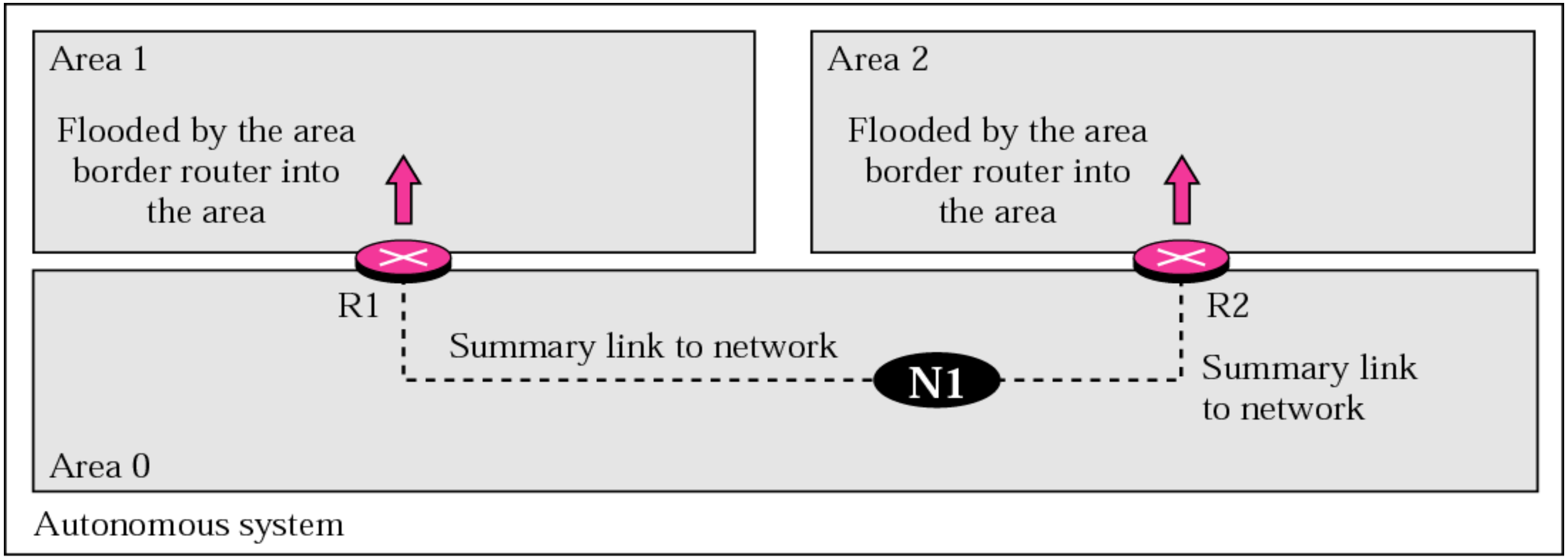
ID	F
SEQ	1

ID	H
SEQ	3

ID	C
SEQ	0

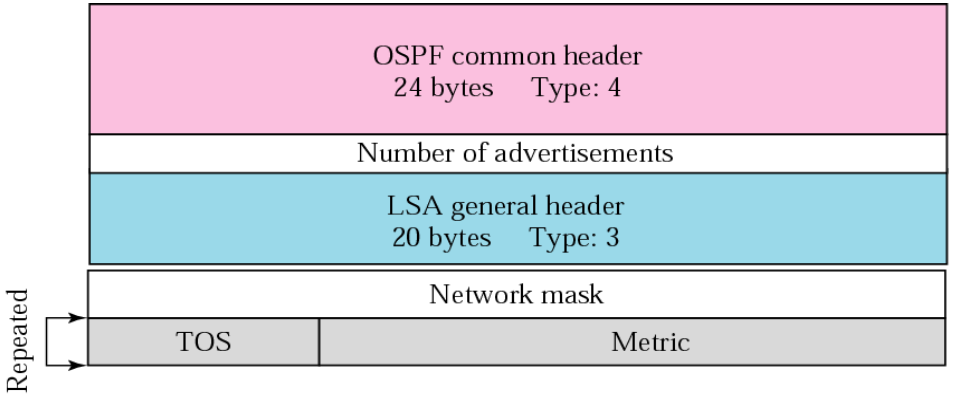
Areas in OSPF

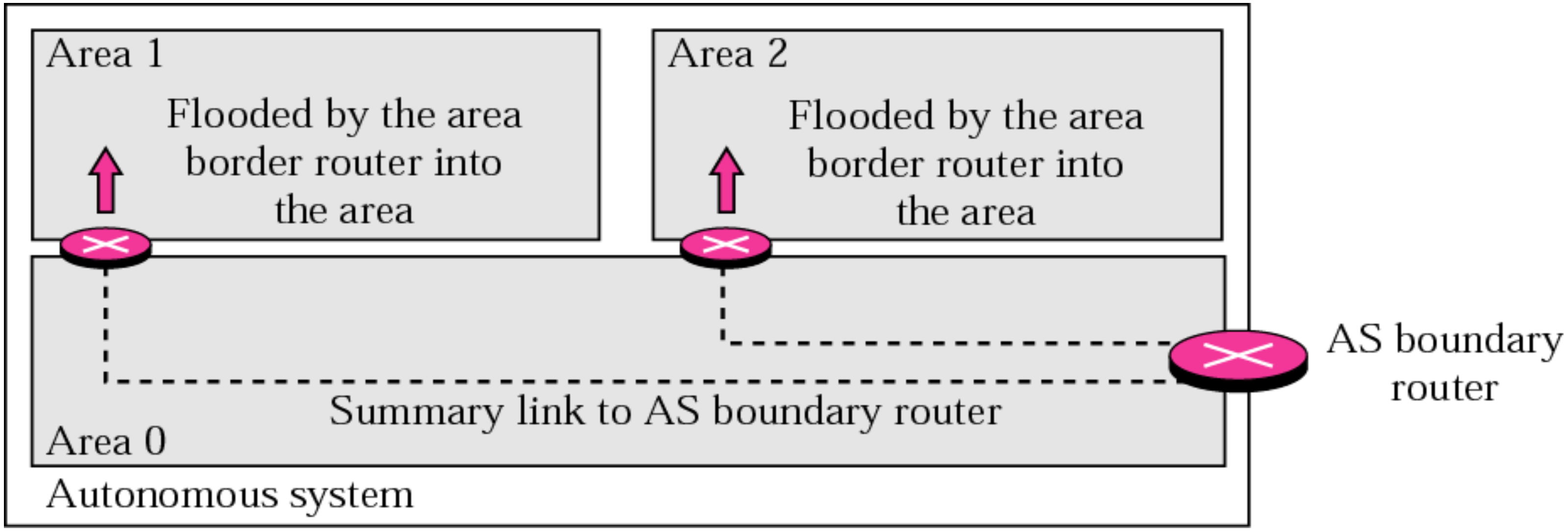
- Limits flooding of advertisements to areas



* Figure is courtesy of B. Forouzan

OSPF Summary Link





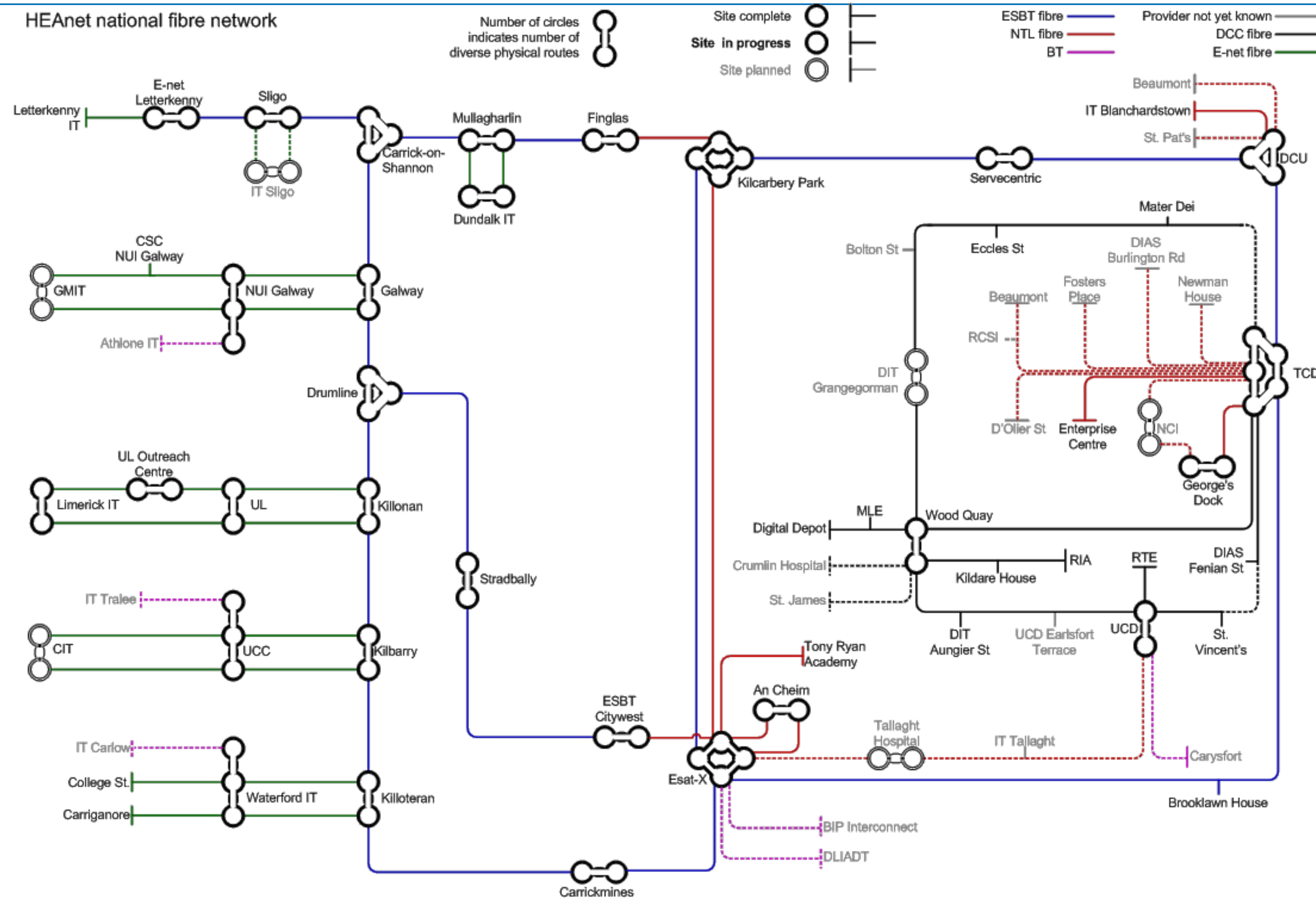
* Figure is courtesy of B. Forouzan

Summary: IntraAS Routing

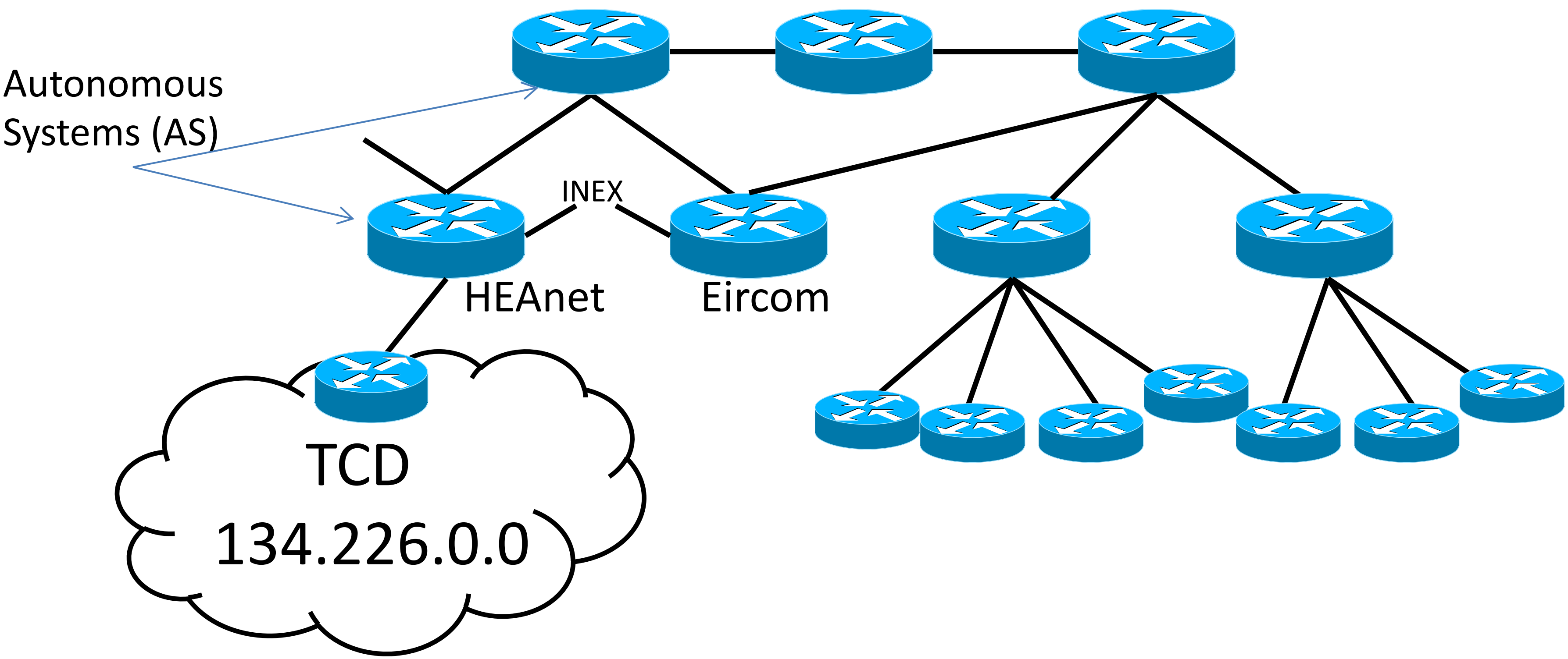
- Distance Vector routing
  - Share complete information with neighbours
  - Count-to-Infinity problem
  - Example: Routing Information Protocol (RIP)

- Link State routing
  - Share information about neighbours with everyone
  - Dijkstra’s Shortest-Path Algorithm
  - Example: Open Shortest Path First (OSPF)

HEAnet Fibre Network



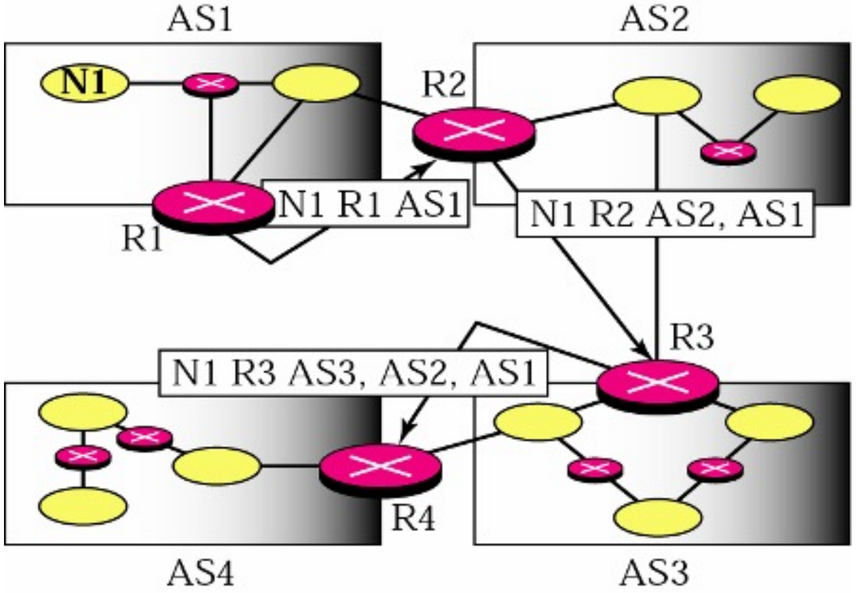
Trinity's Connection



Border Gateway Protocol (BGP)

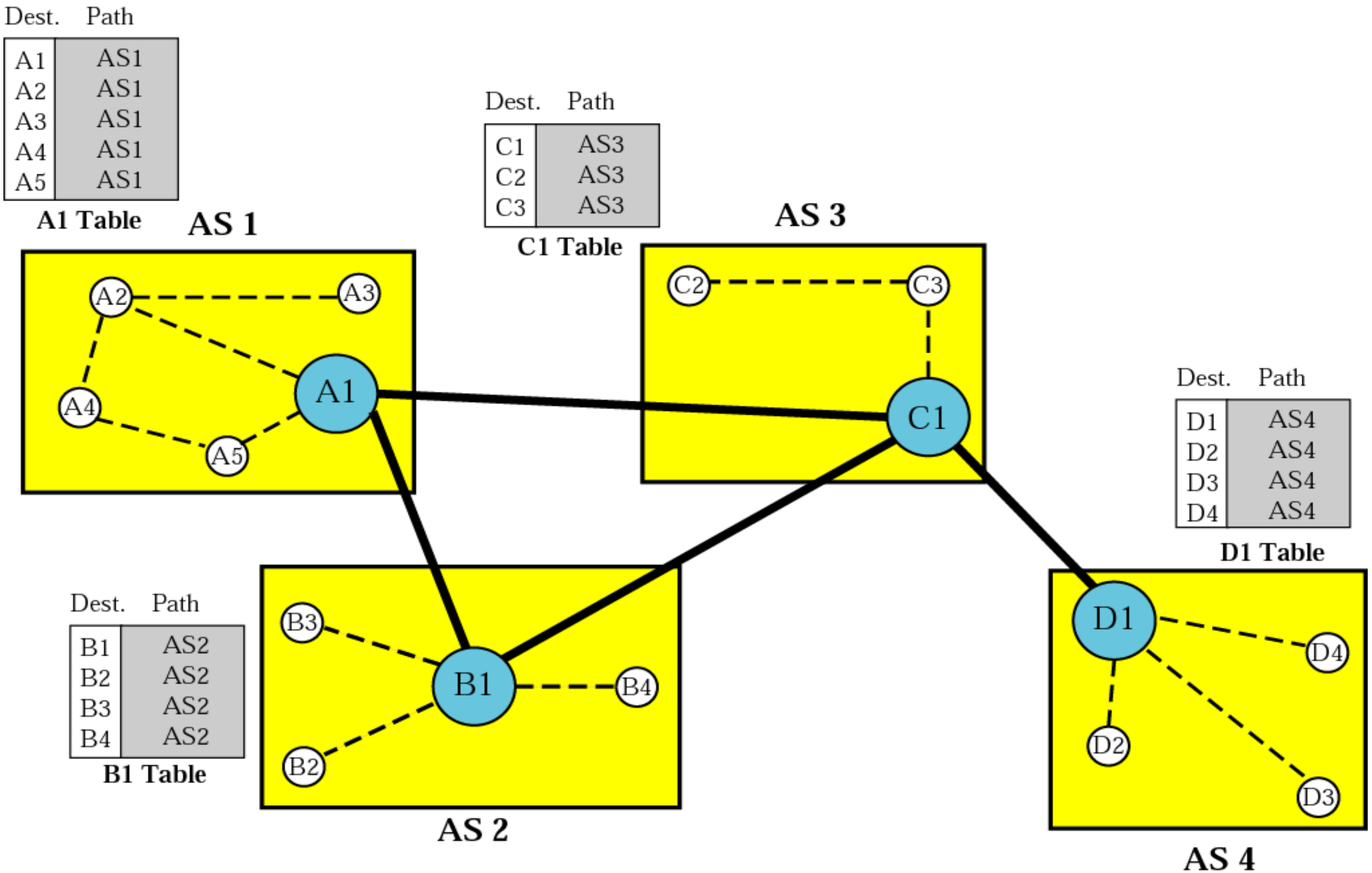
- Uses Path Vector Routing
- Advertisements include complete path to destination
- Router that forwards advertisement adds itself to the list
- Path can be checked for loops
- Policies are applied when incorporating new routes

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

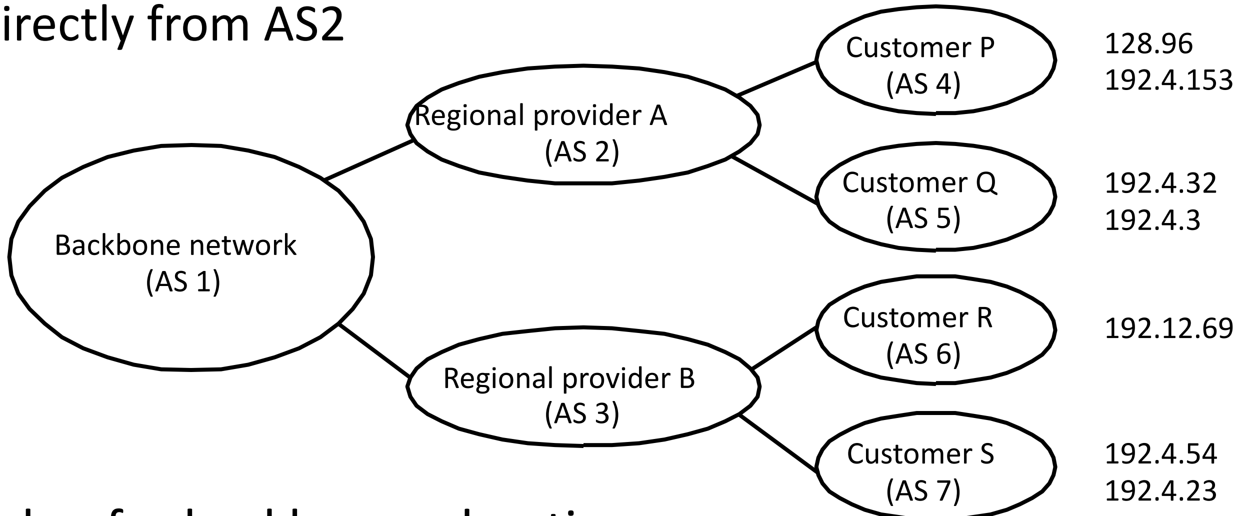
Tables at Autonomous Systems



* Figure is courtesy of B. Forouzan

BGP Example

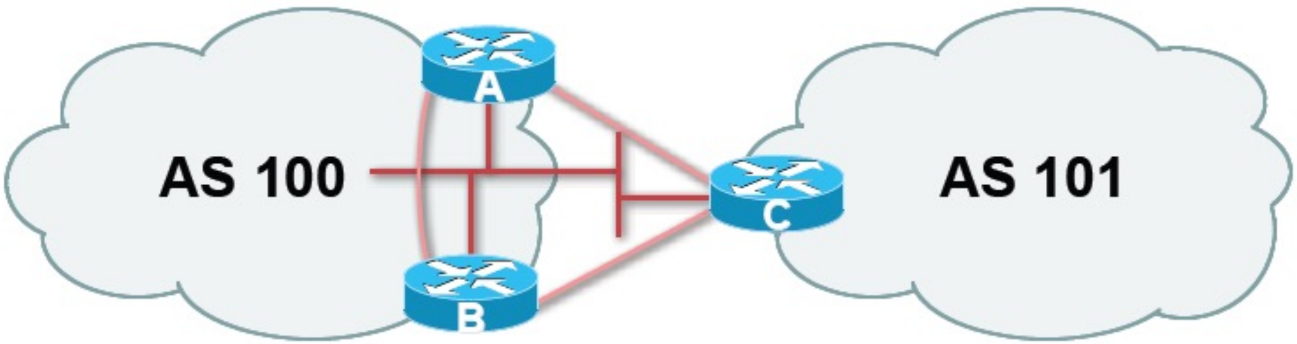
- Speaker for AS2 advertises reachability to P and Q
  - network 128.96, 192.4.153, 192.4.32, and 192.4.3, can be reached directly from AS2



- 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

BGP Example



Router A in AS100

ip address on ethernet interface

```
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
!
```

Local ASN

Remote ASN

ip address of Router C ethernet interface

Inbound and outbound filters

Router C in AS101

ip address on ethernet interface

```
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
!
```

Local ASN

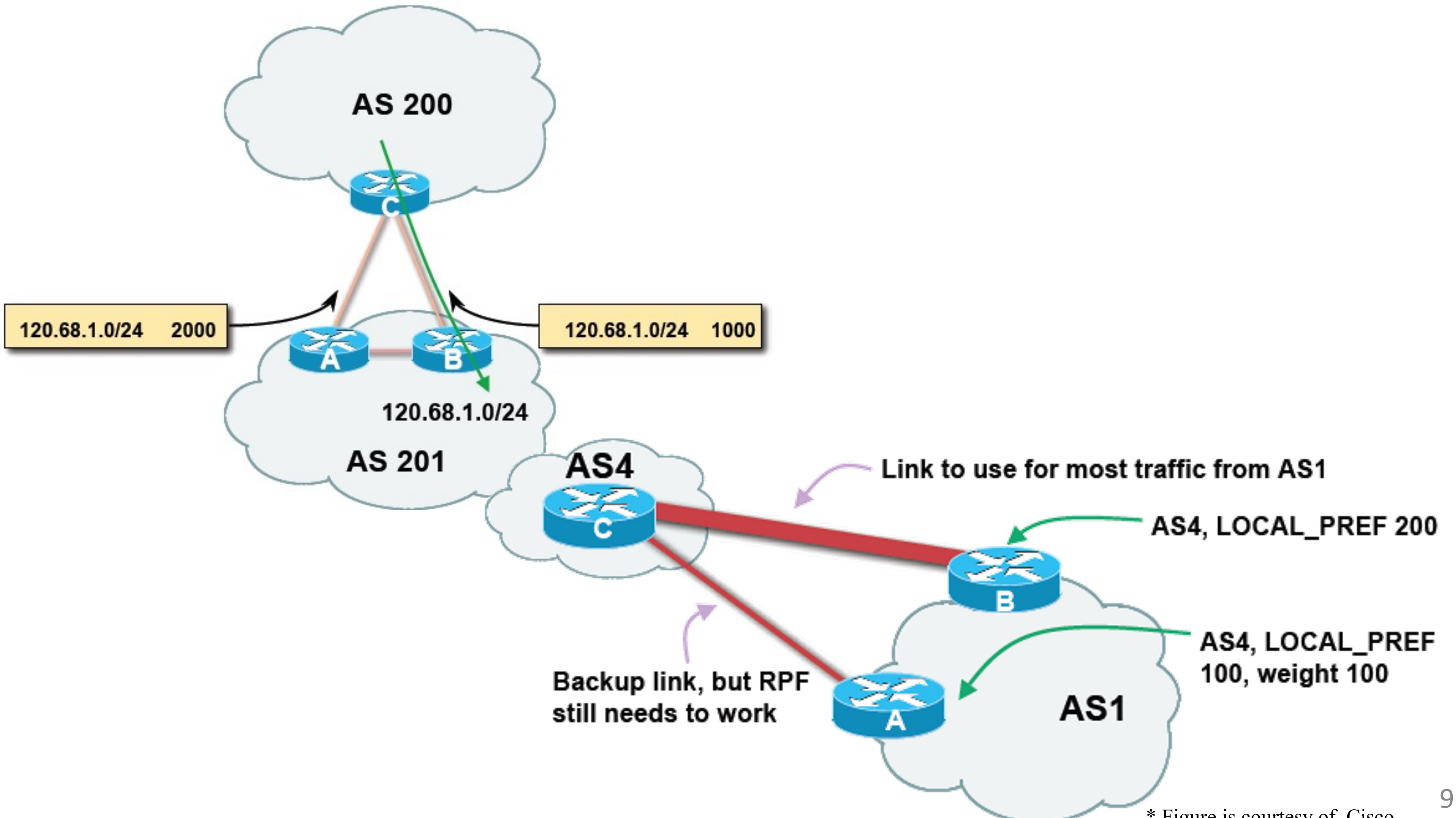
Remote ASN

ip address of Router A ethernet interface

Inbound and outbound filters

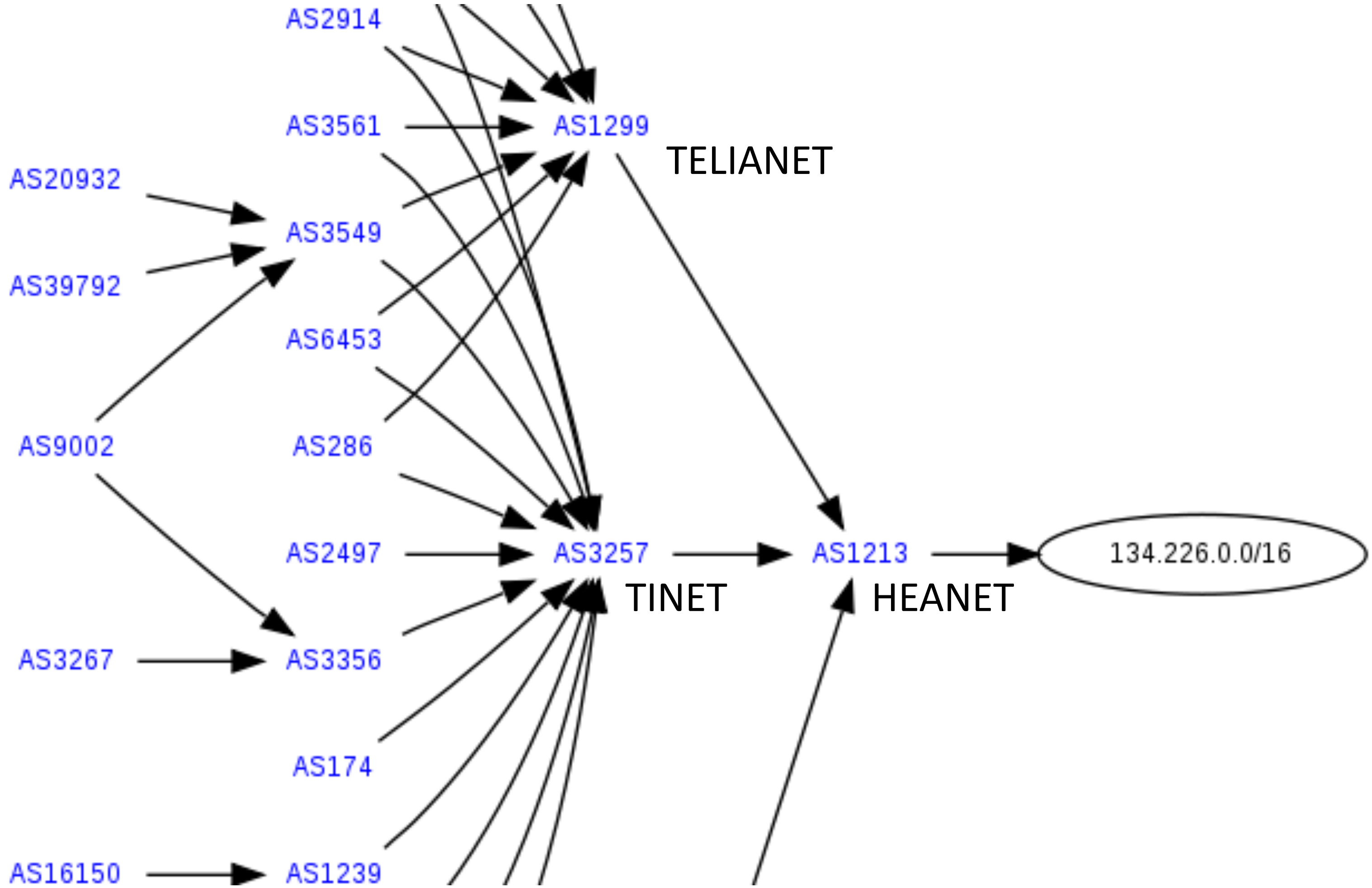
* Figure is courtesy of Cisco

Multiple Exits & Backup Links



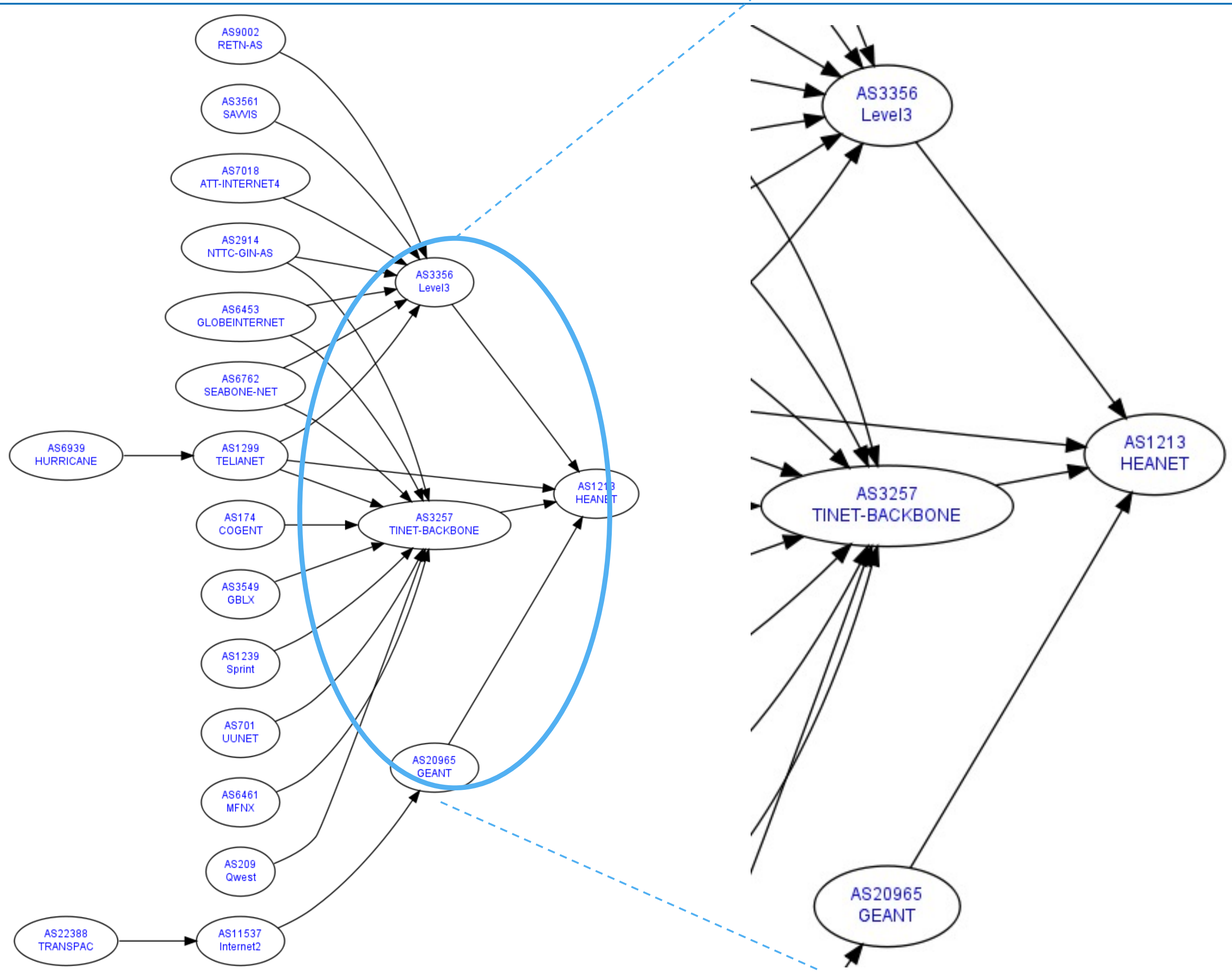
* Figure is courtesy of Cisco

AS1213 - HEANET

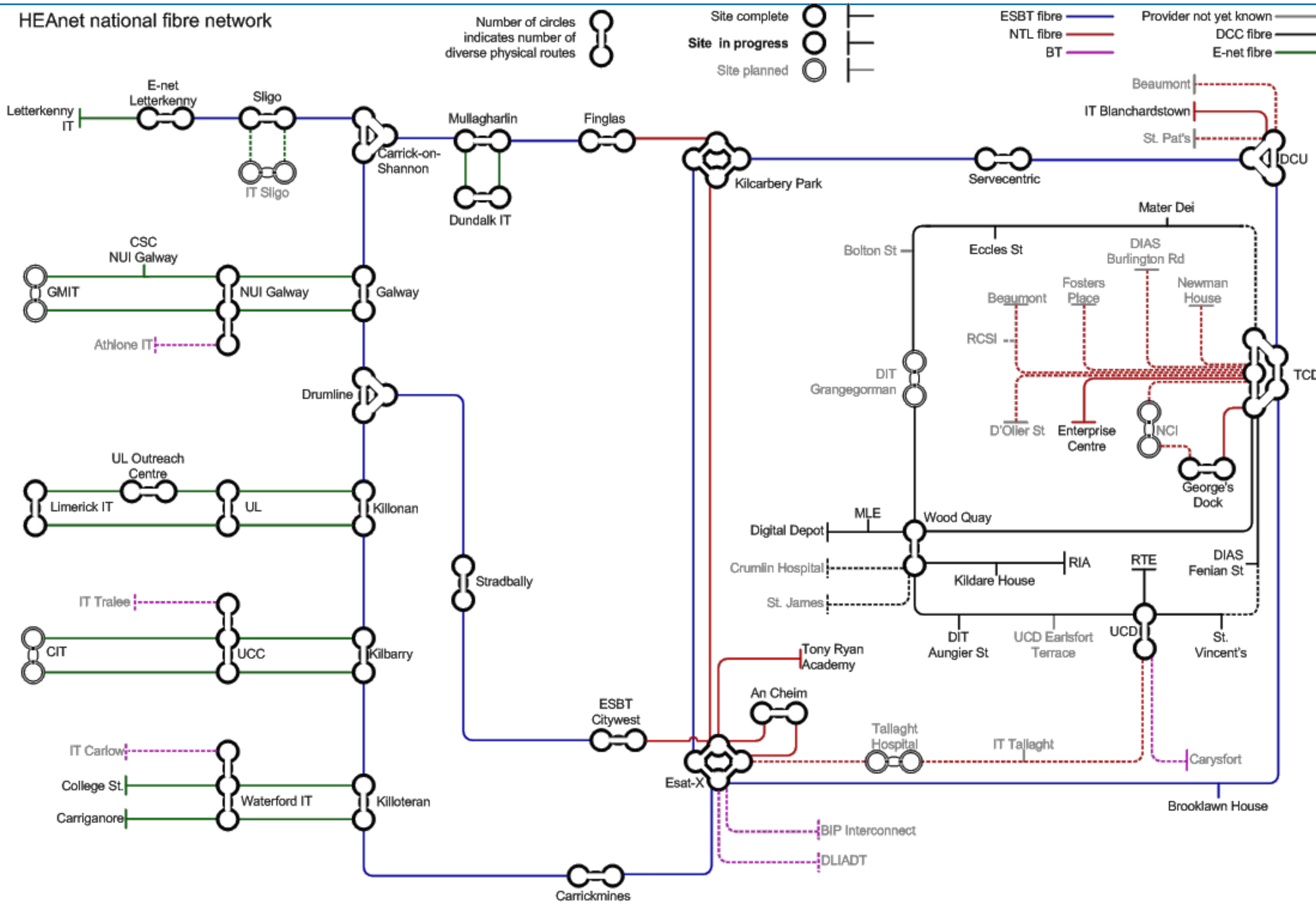


http://www.robtex.com/route/134.226.0.0-16.html

AS1213 - HEANET



HEAnet Fibre Network



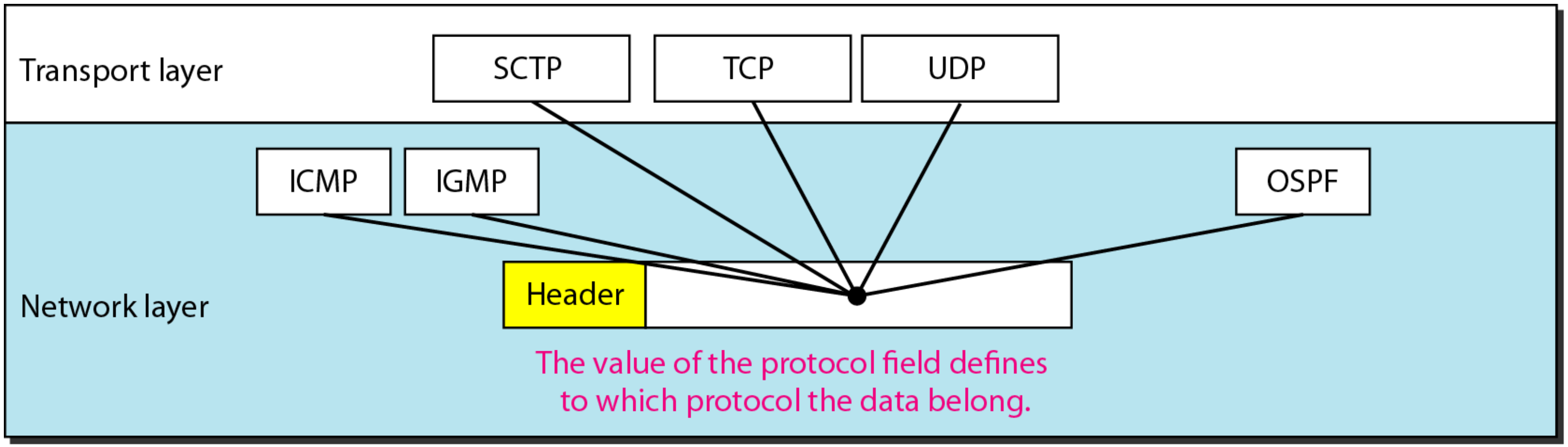
BGP-4: Border Gateway Protocol

- 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

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)

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin