University of BRISTOL

Advanced Networks

(EENGM4211)

Rasheed Hussain

rasheed.hussain@bristol.ac.uk

Best contact : Microsoft Team

Slides courtesy: Prof. Reza Nejabati and Slides from Ross's Book

Unit Outline

1. Introduction
2. Internet Routing and Switching
3. IP Multicast
4. Networking for Real-time Applications
5. Routing in Wireless Networks
6. Quality of Service

Learning outcomes

- At the end of the lecture, the students will be able to:
  - Demonstrate their knowledge of the routing
  - Discuss the rationale for and benefits of link state algorithms
  - Understand and demonstrate the knowledge of Dijekstra routing algorithm
  - Demonstrate the knowledge of Autonomous Systems (AS)
  - Discuss the different types of routers and inter- and intra-AS routing protocols

Internet Routing

Objective

- Describe the major routing/switching mechanisms used in routing in the Internet - unicast, multicast, switching

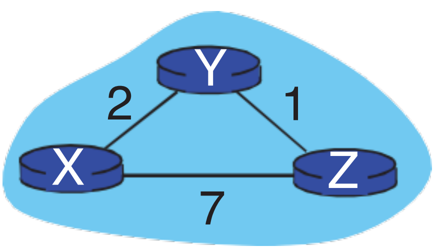
Outcomes

- Describe and explain routing algorithms and protocols
- Compare routing and switching
- Illustrate the above through examples

- Next couple of slides are repeted from previous lecture for the sake of warm-up

Distance Vector Algorithm: example

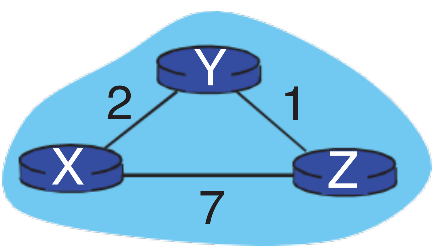
D^X	cost via Y	cost via Z
dest Y	2	∞
dest Z	∞	7



Distance Vector Algorithm: example

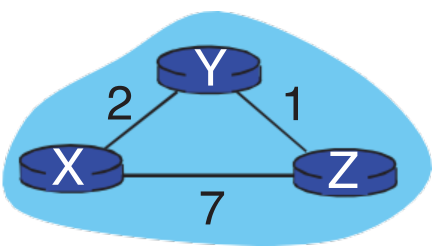
D^X	cost via Y	Z
dest Y	2	∞
dest Z	∞	7

D^Y	cost via X	Z
dest X	2	∞
dest Z	∞	1



Distance Vector Algorithm: example

dest	D^X	cost via Y	cost via Z
	Y	2	∞
	Z	∞	7



dest	D^Y	cost via X	cost via Z
	X	2	∞
	Z	∞	1

dest	D^Z	cost via X	cost via Y
	X	7	∞
	Y	∞	1

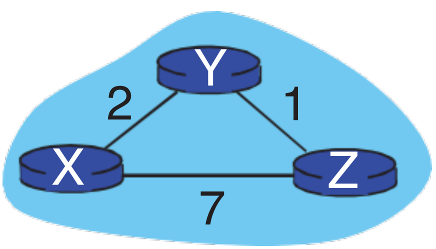
Distance Vector Algorithm: example

D^X	cost via Y	cost via Z
dest Y	2	∞
dest Z	∞	7

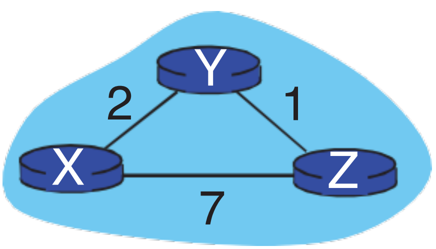
D^Y	cost via X	cost via Z
dest X	2	∞
dest Z	∞	1

D^Z	cost via X	cost via Y
dest X	7	∞
dest Y	∞	1

D^X	cost via Y	cost via Z
dest Y	2	8
dest Z	3	7



Distance Vector Algorithm: example



D^X	cost via Y	cost via Z
dest Y	2	∞
dest Z	∞	7

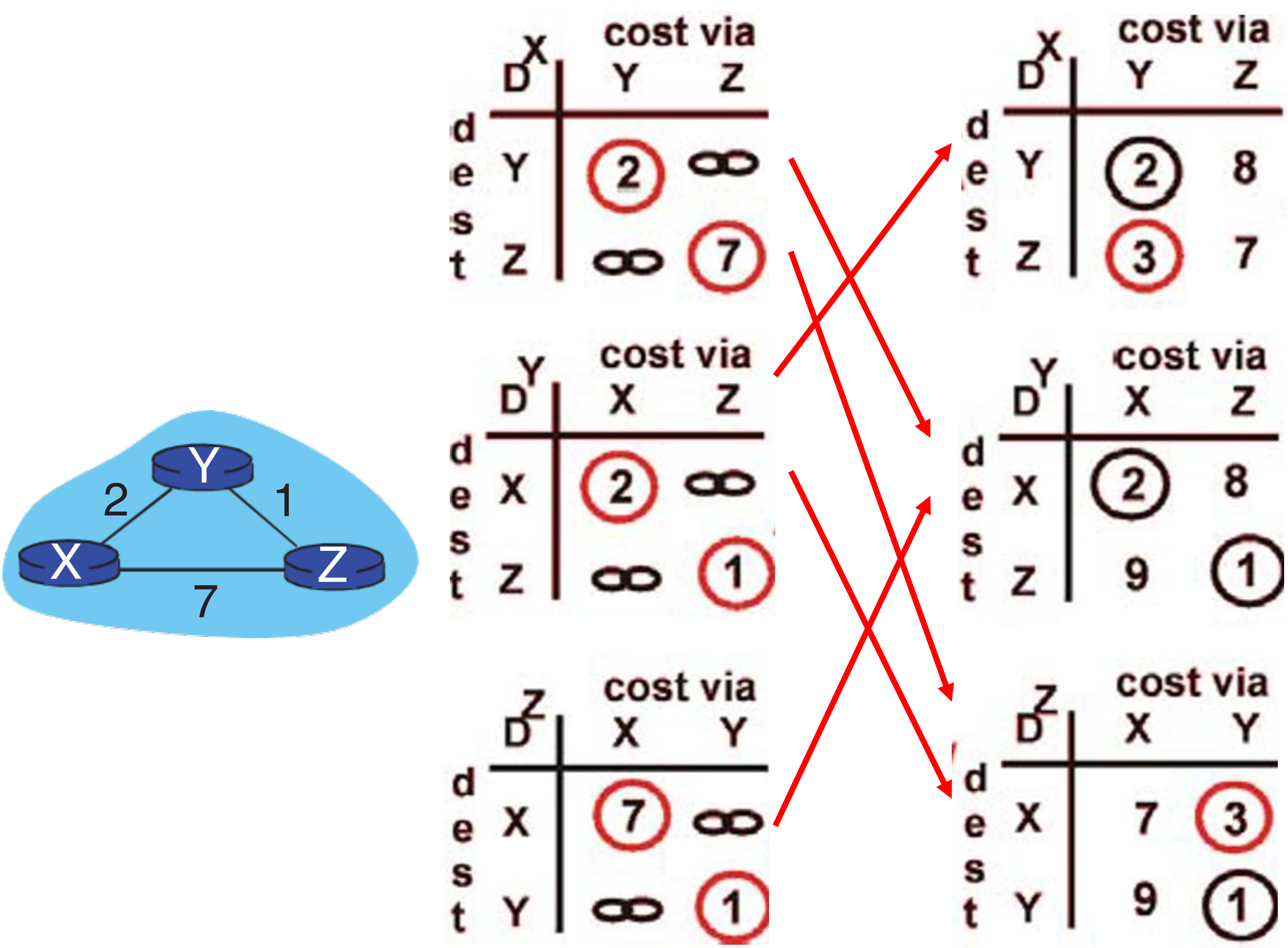
D^Y	cost via X	cost via Z
dest X	2	∞
dest Z	∞	1

D^Z	cost via X	cost via Y
dest X	7	∞
dest Y	∞	1

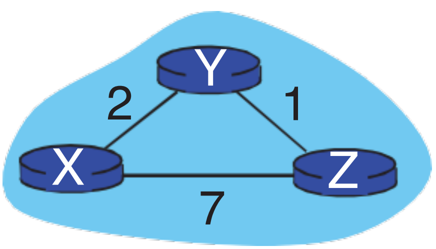
D^X	cost via Y	cost via Z
dest Y	2	8
dest Z	3	7

D^Y	cost via X	cost via Z
dest X	2	8
dest Z	9	1

Distance Vector Algorithm: example



Distance Vector Algorithm: example



D^X (cost via)

dest	Y	Z
Y	2	∞
Z	∞	7

D^Y (cost via)

dest	X	Z
X	2	∞
Z	∞	1

D^Z (cost via)

dest	X	Y
X	7	∞
Y	∞	1

D^X (cost via)

dest	Y	Z
Y	2	8
Z	3	7

D^Y (cost via)

dest	X	Z
X	2	8
Z	9	1

D^Z (cost via)

dest	X	Y
X	7	3
Y	9	1

D^X (cost via)

dest	Y	Z
Y		
Z		

D^Y (cost via)

dest	X	Z
X		
Z		

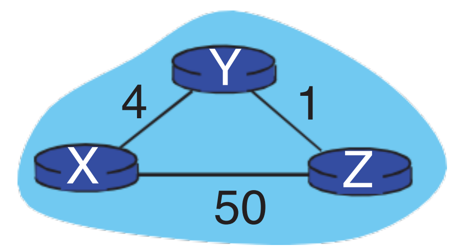
D^Z (cost via)

dest	X	Y
X		
Y		

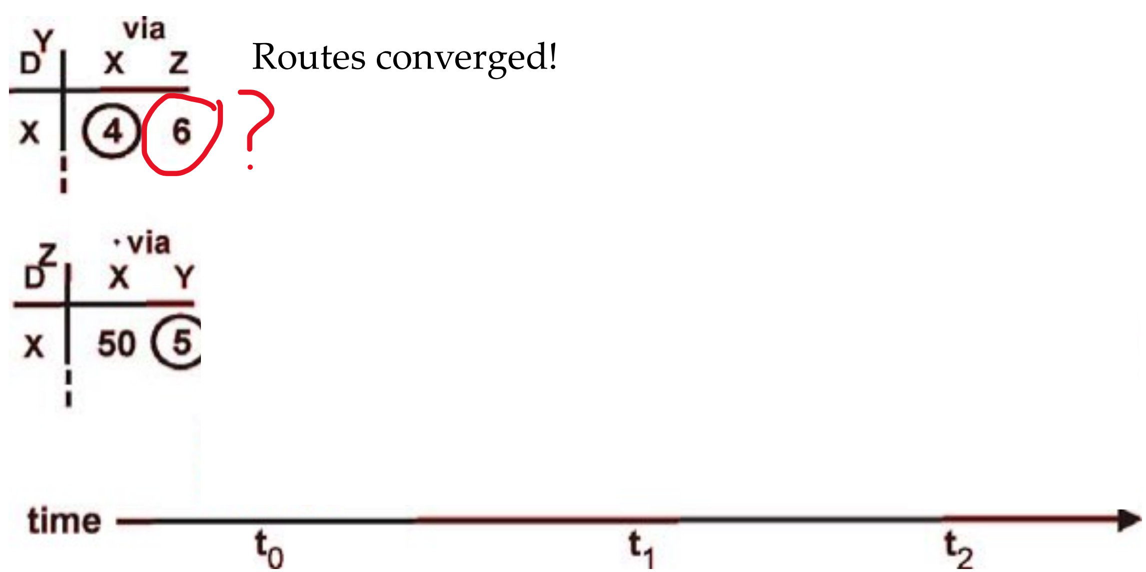
Distance Vector: Link cost changes

Link cost changes:

- Node detects local link cost change
- Updates distance table
- If cost change in least cost path, notify neighbors



Good news travels fast

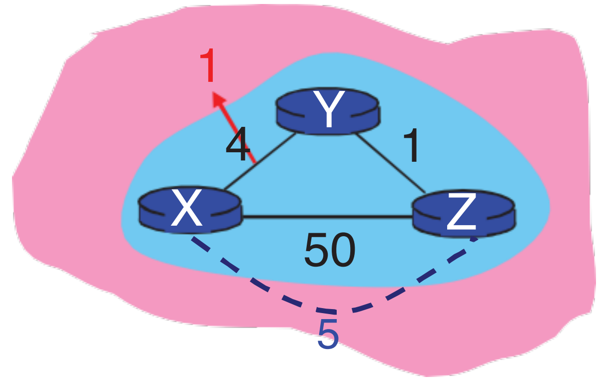


Algorithm terminates

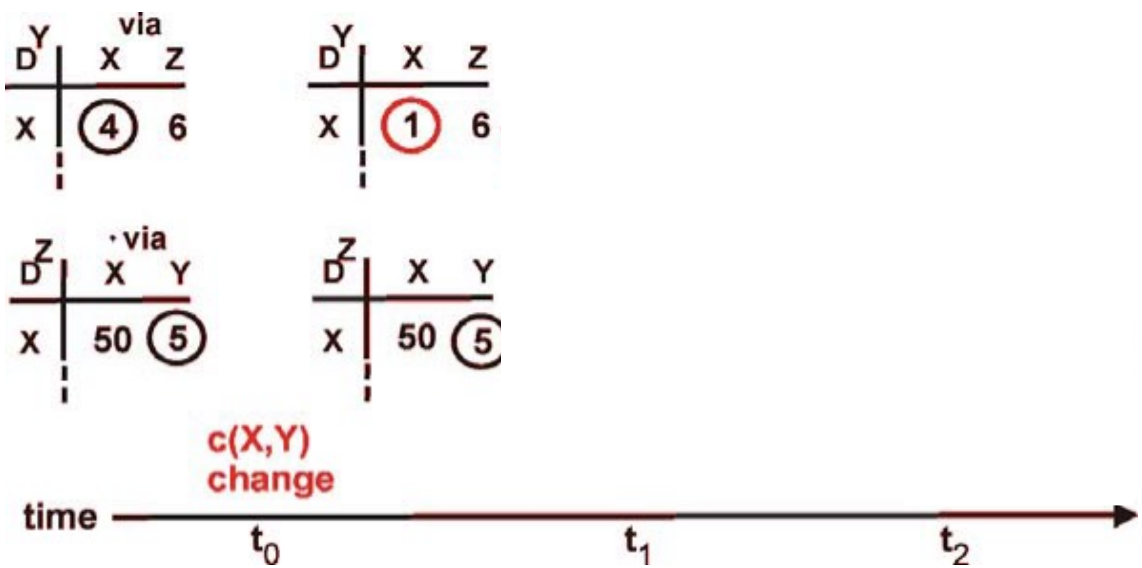
Distance Vector: Link cost changes

Link cost changes:

- Node detects local link cost change
- Updates distance table
- If cost change in least cost path, notify neighbors



Good news travels fast

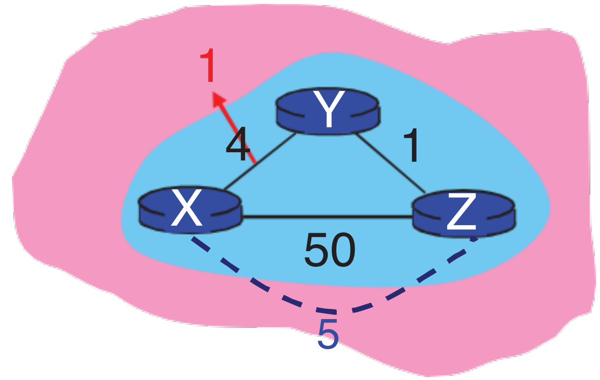


Algorithm terminates

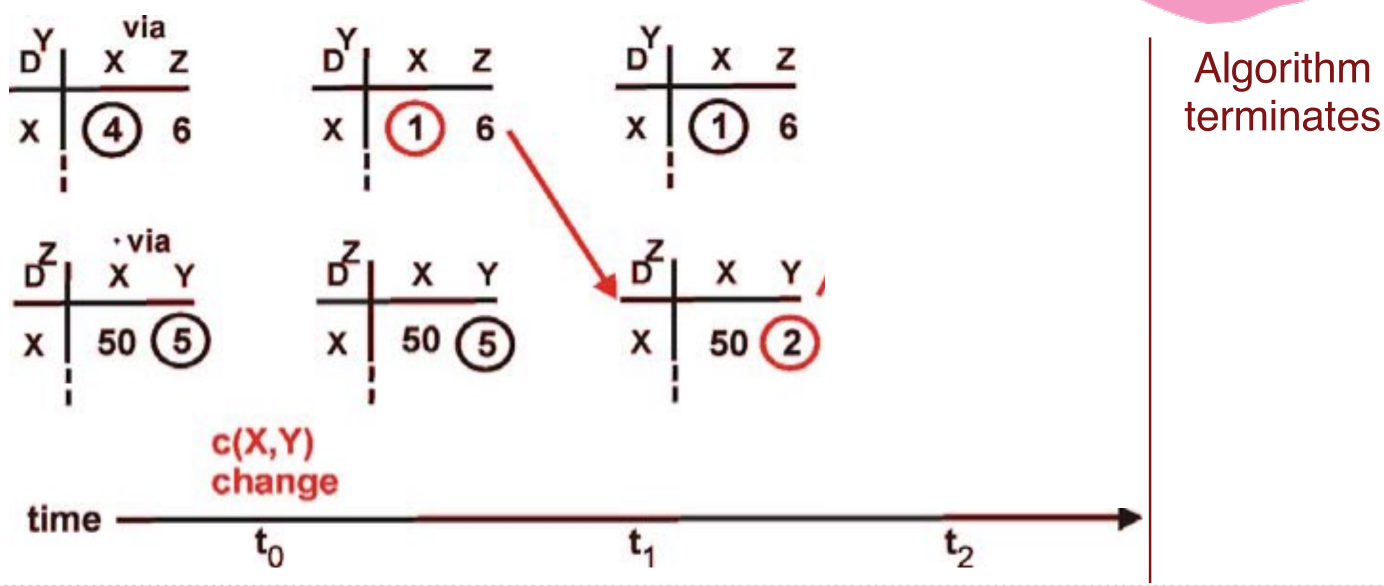
Distance Vector: Link cost changes

Link cost changes:

- Node detects local link cost change
- Updates distance table
- If cost change in least cost path, notify neighbors



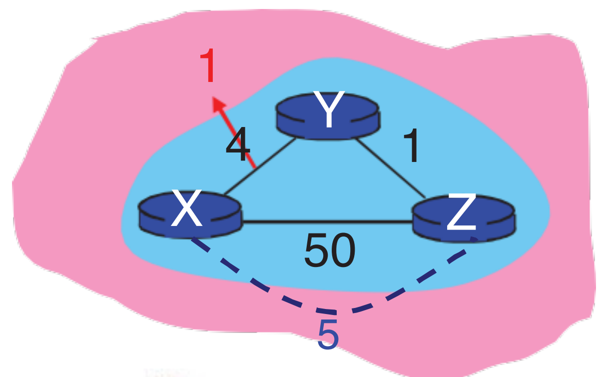
Good news travels fast



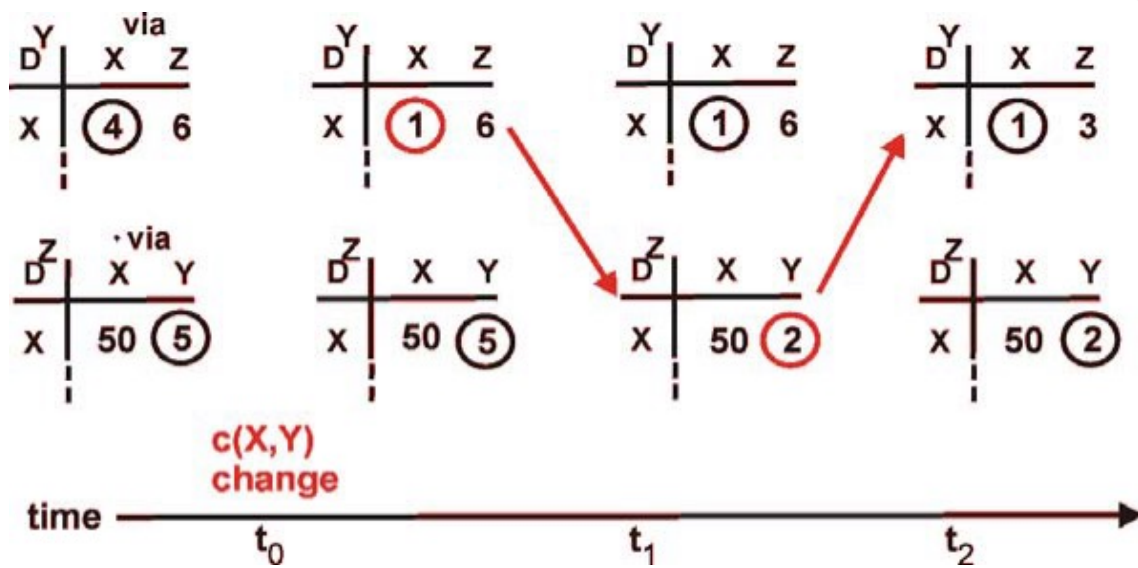
Distance Vector: Link cost changes

Link cost changes:

- Node detects local link cost change
- Updates distance table
- If cost change in least cost path, notify neighbors



Good news travels fast

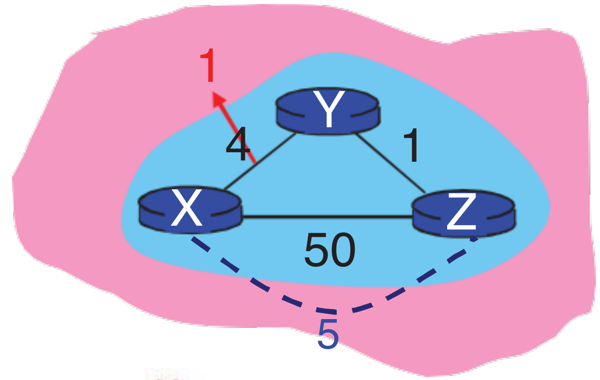


Algorithm terminates

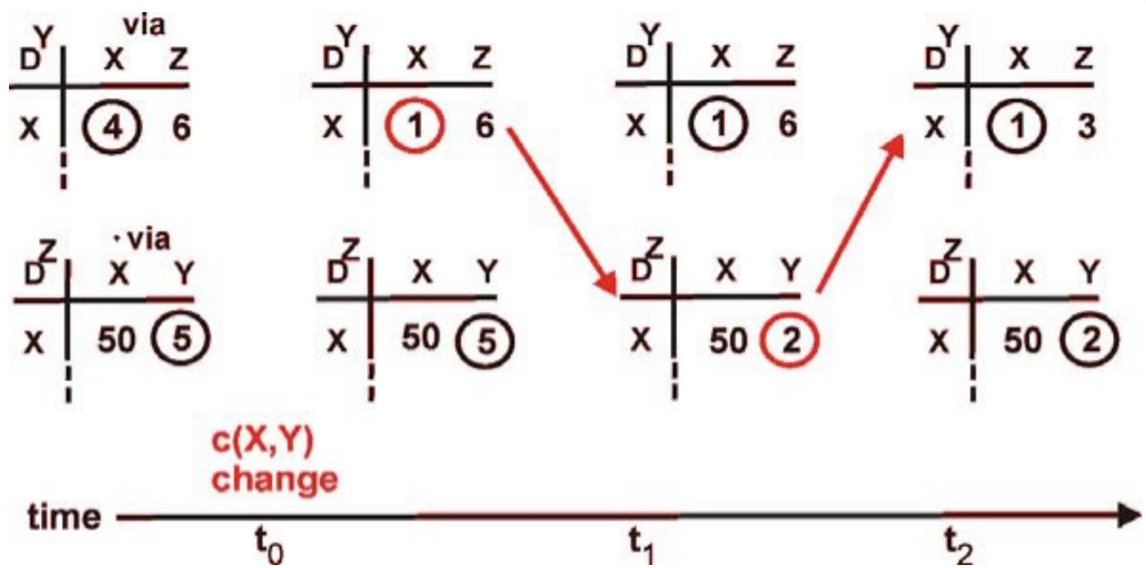
Distance Vector: Link cost changes

Link cost changes:

- Node detects local link cost change
- Updates distance table
- If cost change in least cost path, notify neighbors

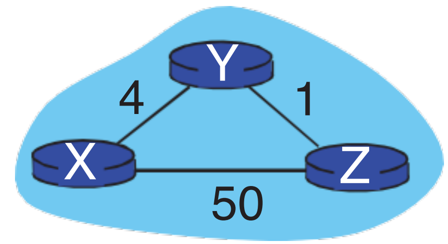


Good news travels fast



Algorithm terminates

Distance Vector: Link cost changes



Link cost changes:

- Good news travels fast
- But not bad news…
- “Count to Infinity” problem!

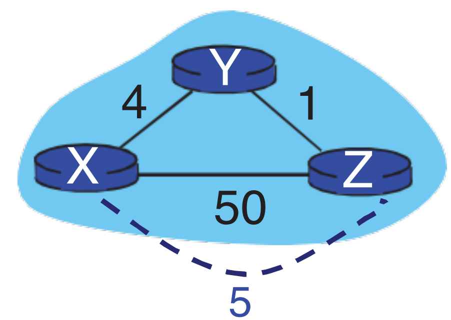
D^Y	via X	via Z
X	4	6

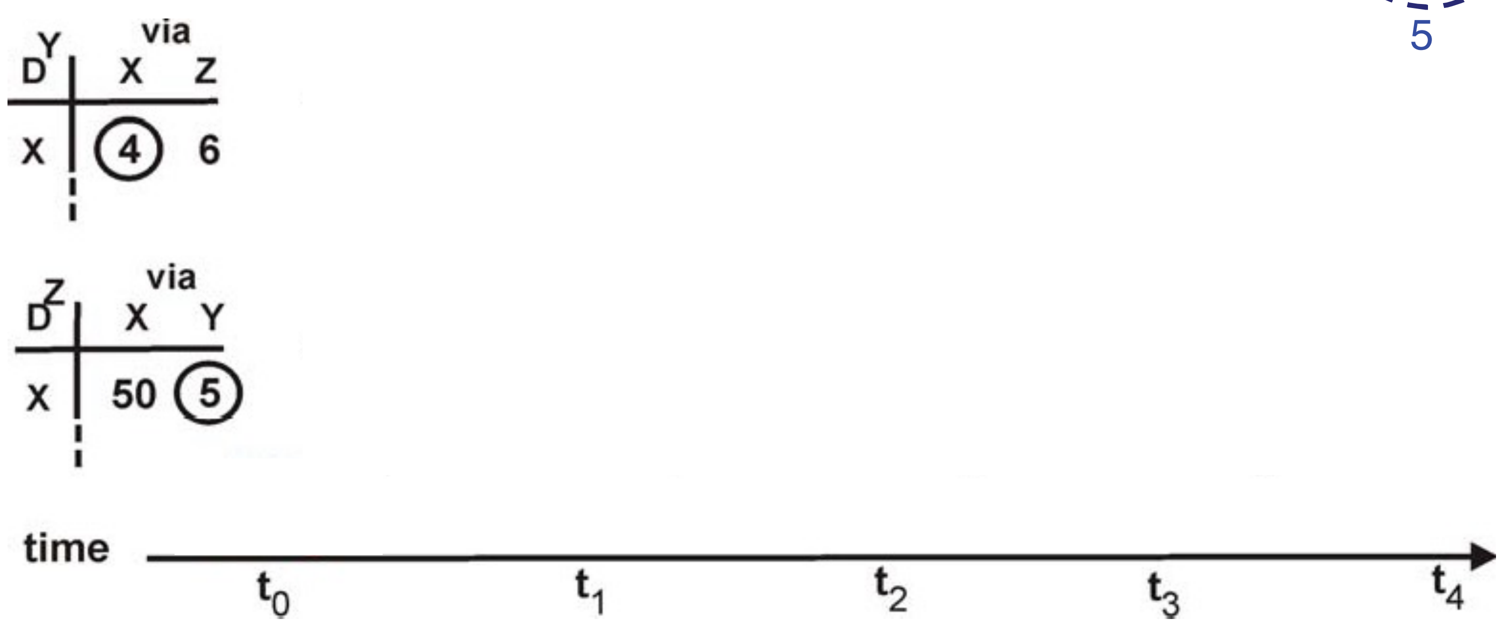
D^Z	via X	via Y
X	50	5

Distance Vector: Link cost changes

Link cost changes:

- Good news travels fast
- But not bad news…
- “Count to Infinity” problem!

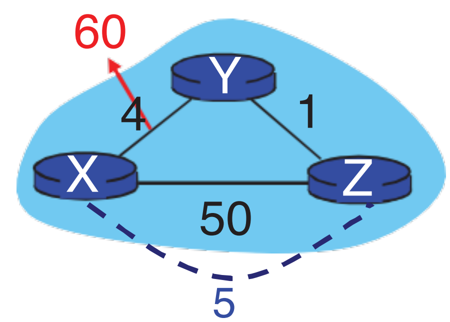


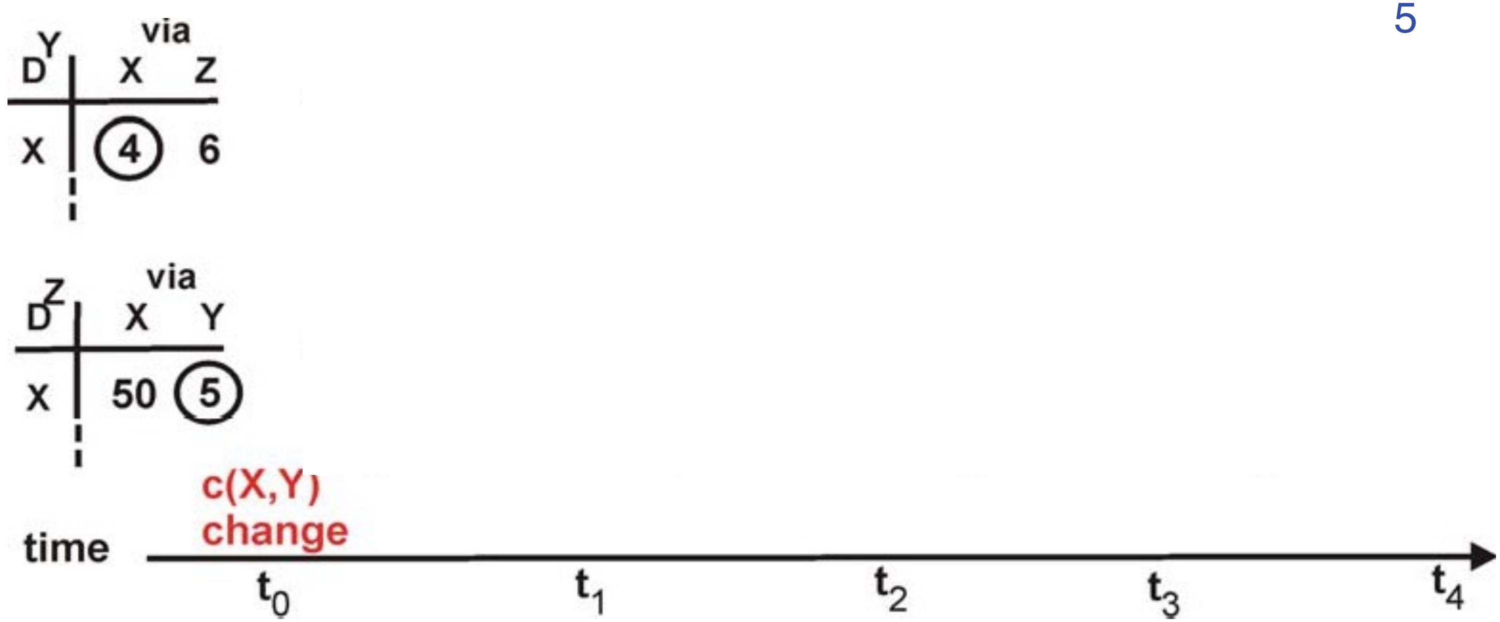


Distance Vector: Link cost changes

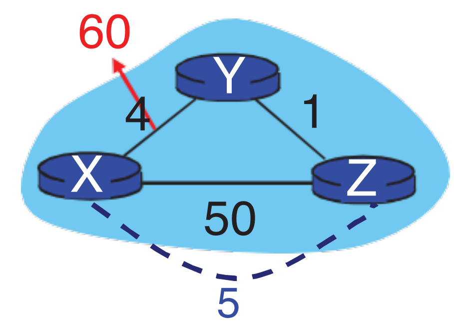
Link cost changes:

- Good news travels fast
- But not bad news…
- “Count to Infinity” problem!



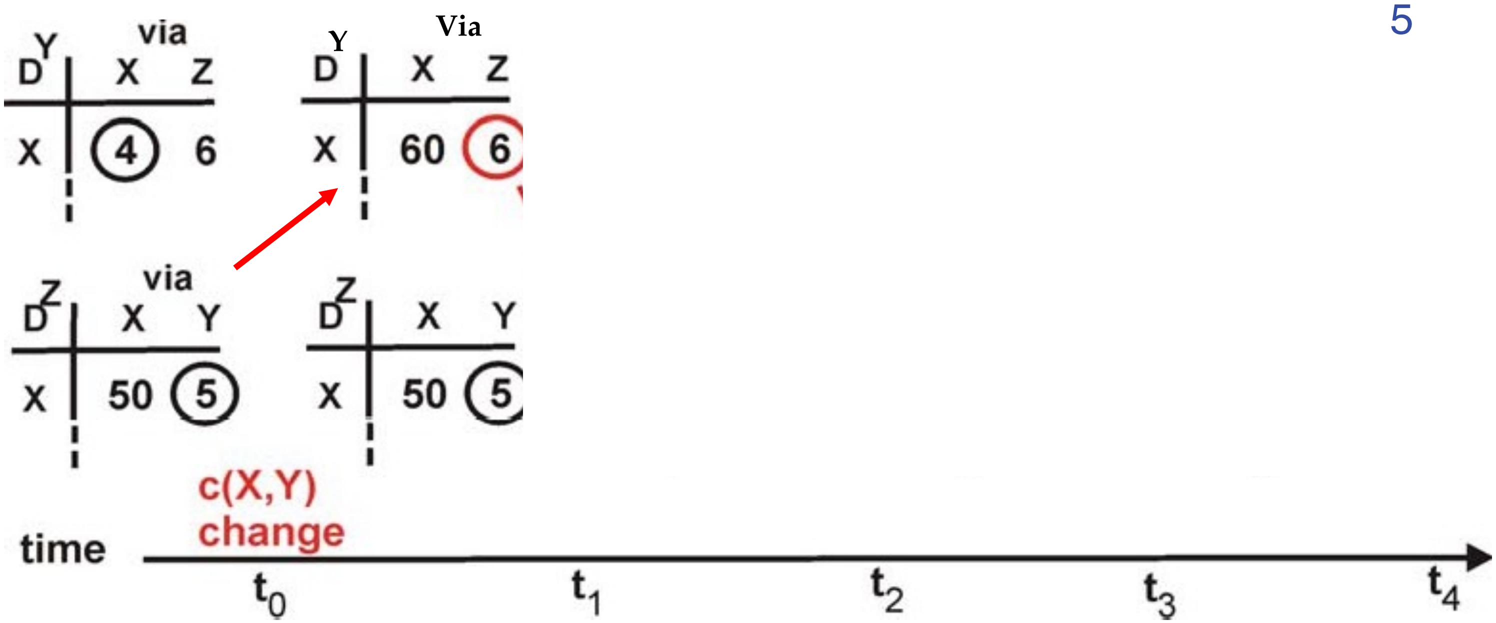


Distance Vector: Link cost changes



Link cost changes:

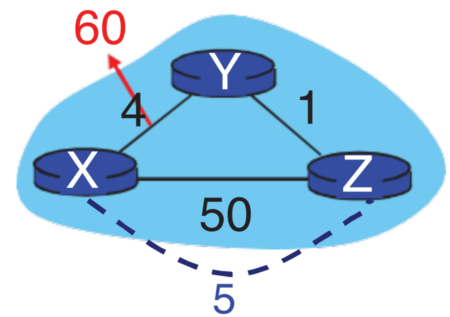
- Good news travels fast
- But not bad news…
- “Count to Infinity” problem!

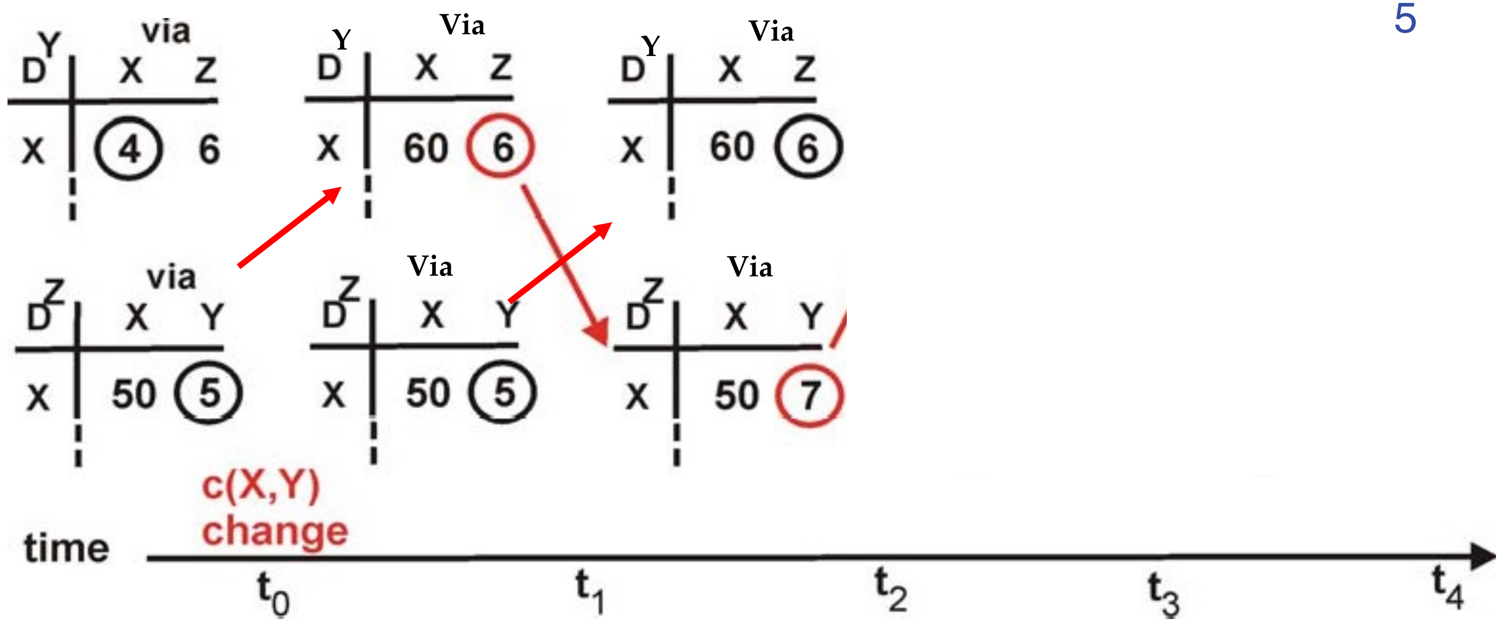


Distance Vector: Link cost changes

Link cost changes:

- Good news travels fast
- But not bad news…
- “Count to Infinity” problem!

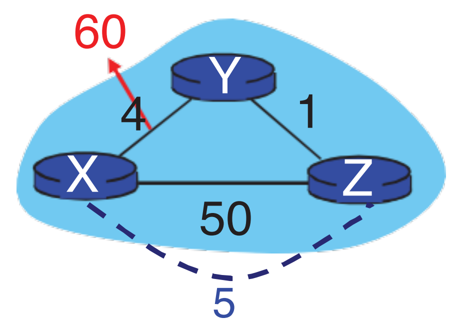


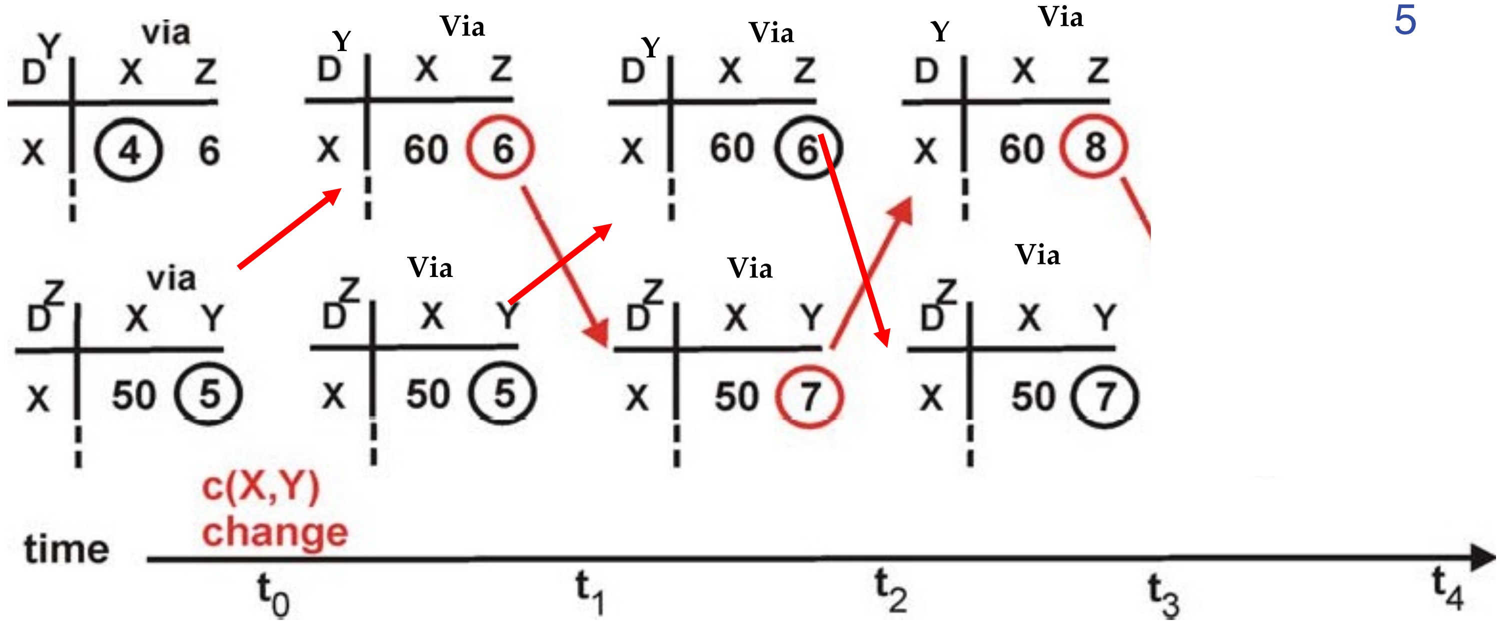


Distance Vector: Link cost changes

Link cost changes:

- Good news travels fast
- But not bad news…
- "Count to Infinity" problem!



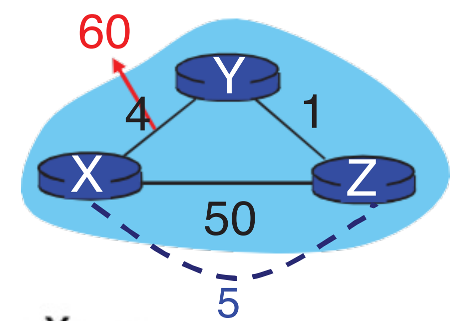


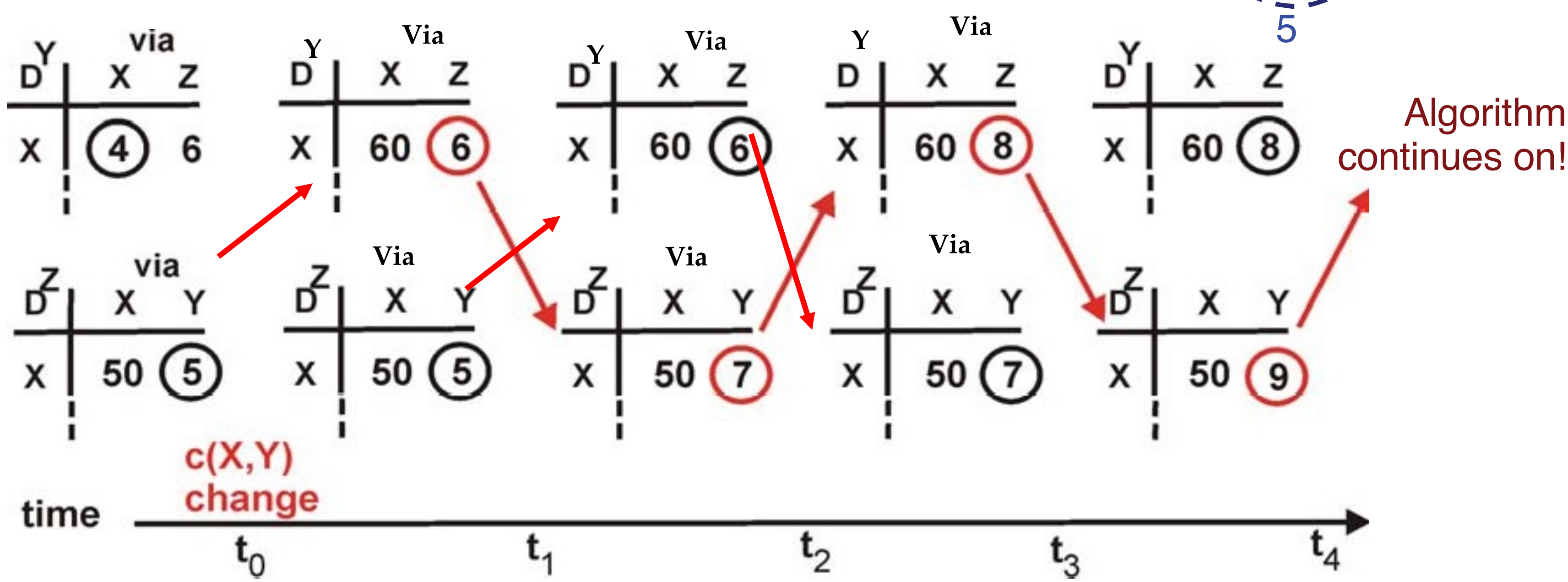
University of BRISTOL

Distance Vector: Link cost changes

Link cost changes:

- Good news travels fast
- But not bad news…
- “Count to Infinity” problem!





University of BRISTOL

Link State algorithms

- Copy of complete network map, regularly updated:
  - Routing table in each router, containing
    - "distance" to neighbours, and
    - which line to use for each neighbour
- Router ‘R’ knows all routers and sends its table to each by flooding.
- Each neighbour periodically sends ‘R’ its own routing table.
- ‘R’ can therefore update its own routing table from this info.
- Used for interior gateway routing in autonomous systems.

Dijkstra's algorithm

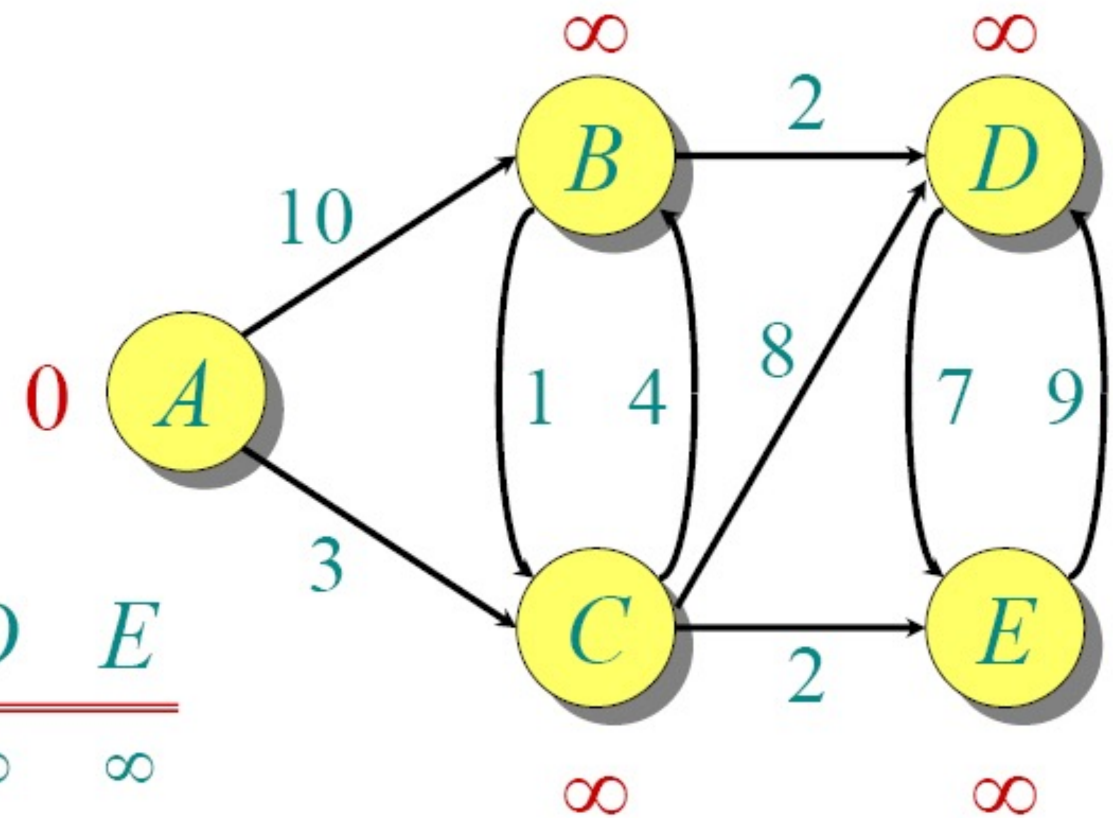
Dijkstra's algorithm - is a solution to the single-source shortest path problem in graph theory.

Input: Weighted graph G={E,V} and source vertex $v \in V$,

Output: Lengths of shortest paths (or the shortest paths themselves) from a given source vertex $v \in V$ to all other vertices

Dijkstra's algorithm

Initialize:

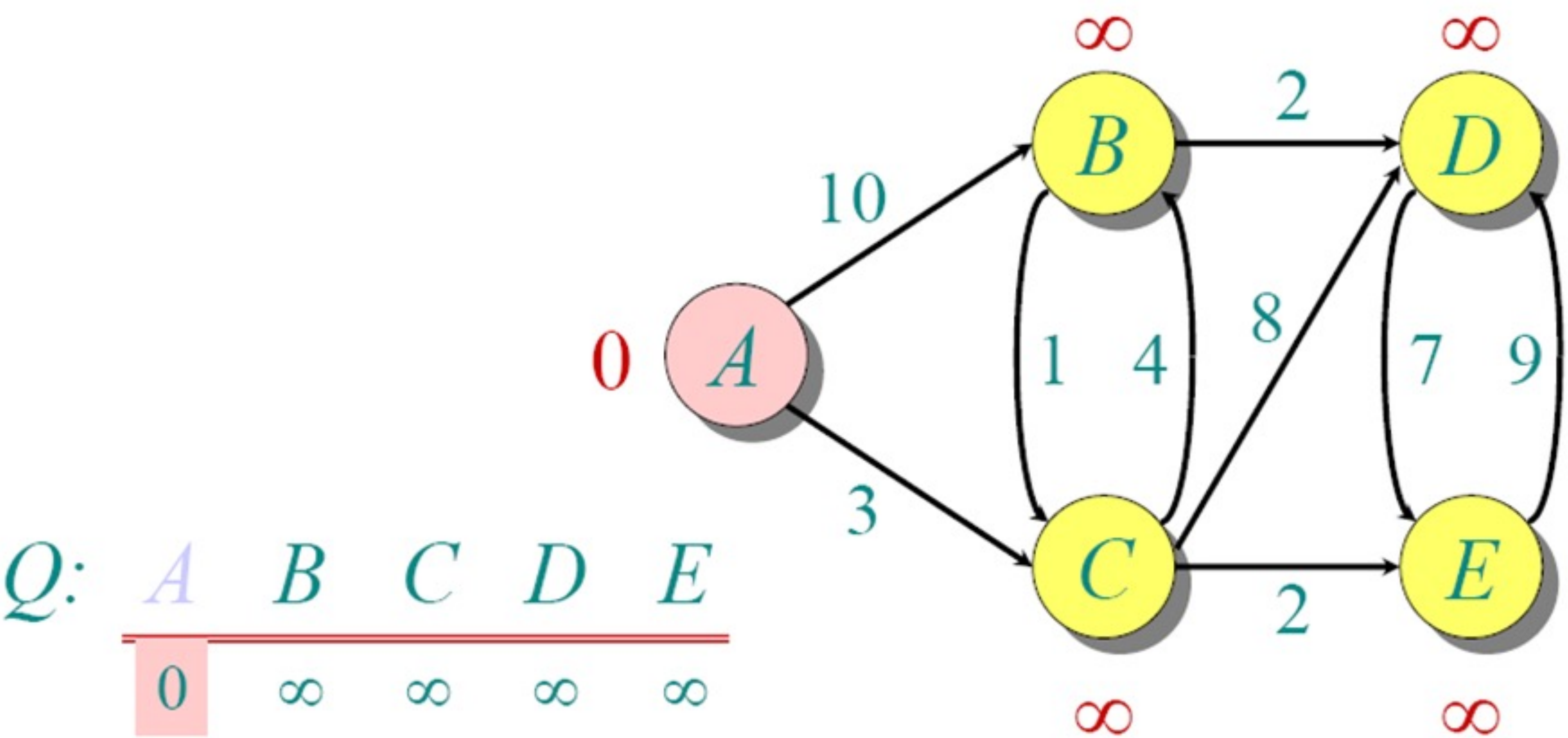


Q:	A	B	C	D	E
	0	∞	∞	∞	∞

S: {}

- By Laksman Veeravagu and Luis Barrera

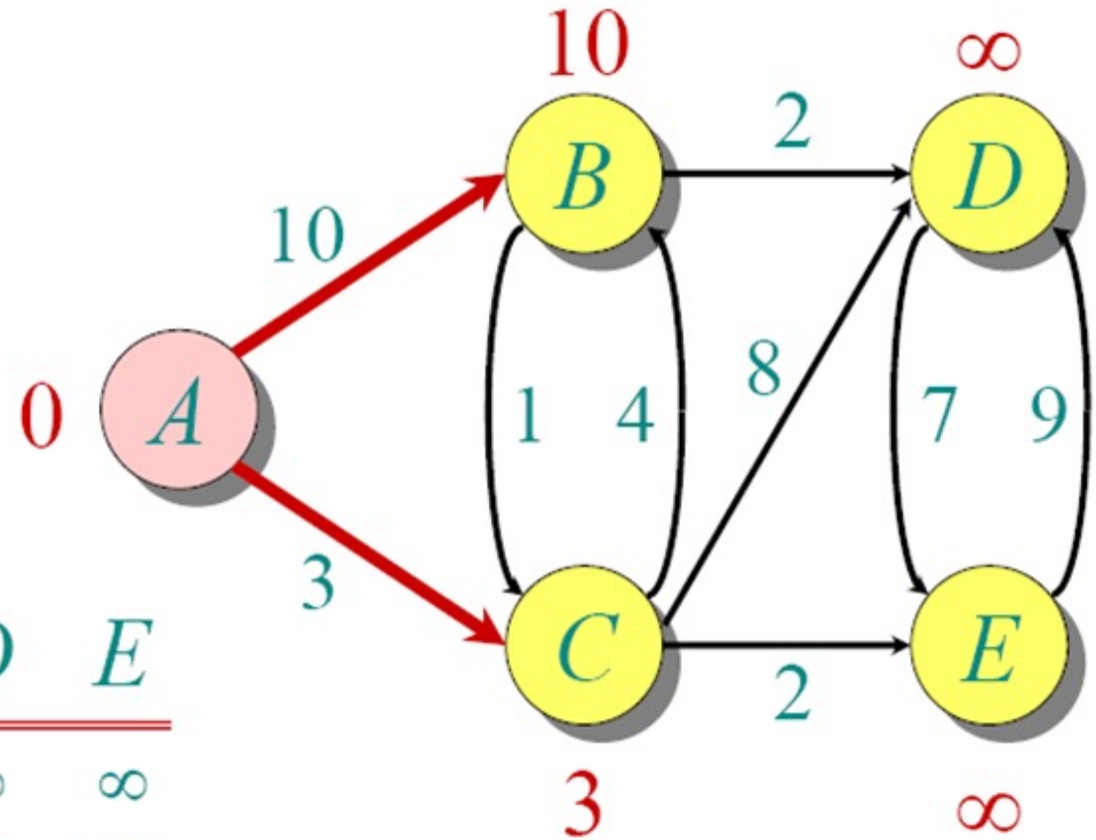
Dijkstra's algorithm



- By Laksman Veeravagu and Luis Barrera

University of BRISTOL

Dijkstra's algorithm

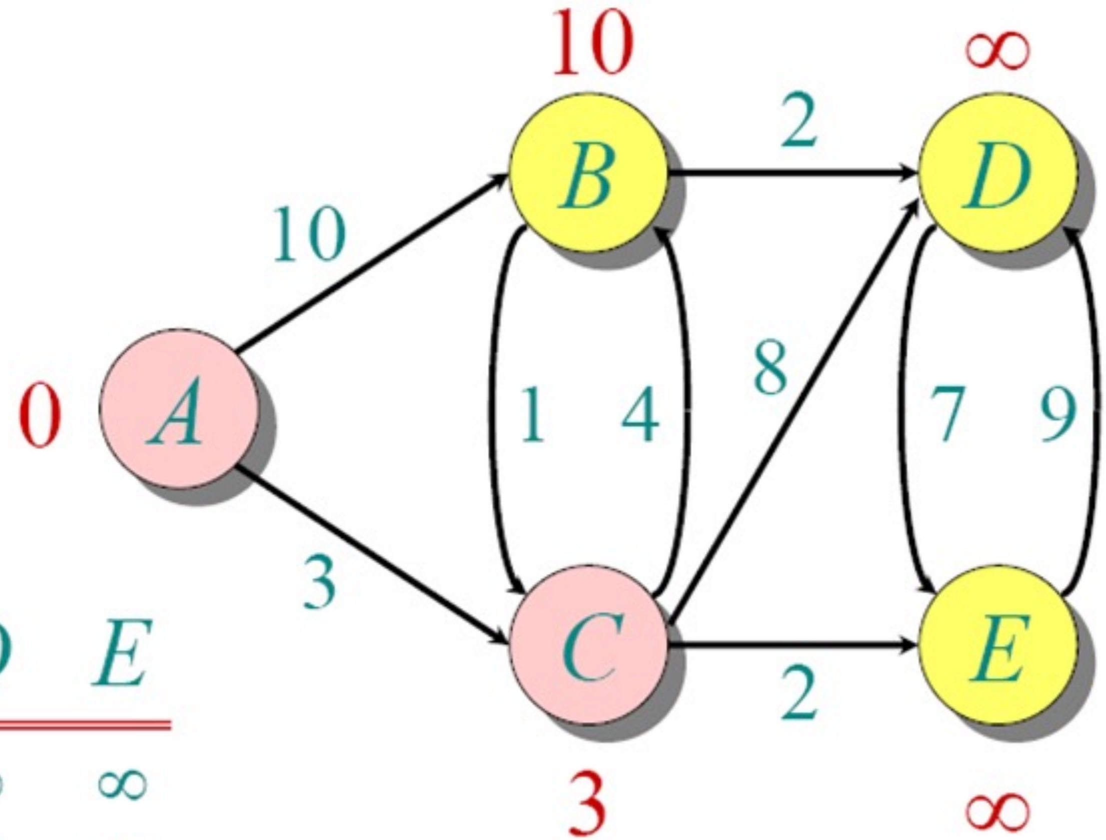


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

$S: \{ A \}$

- By Laksman Veeravagu and Luis Barrera

Dijkstra's algorithm

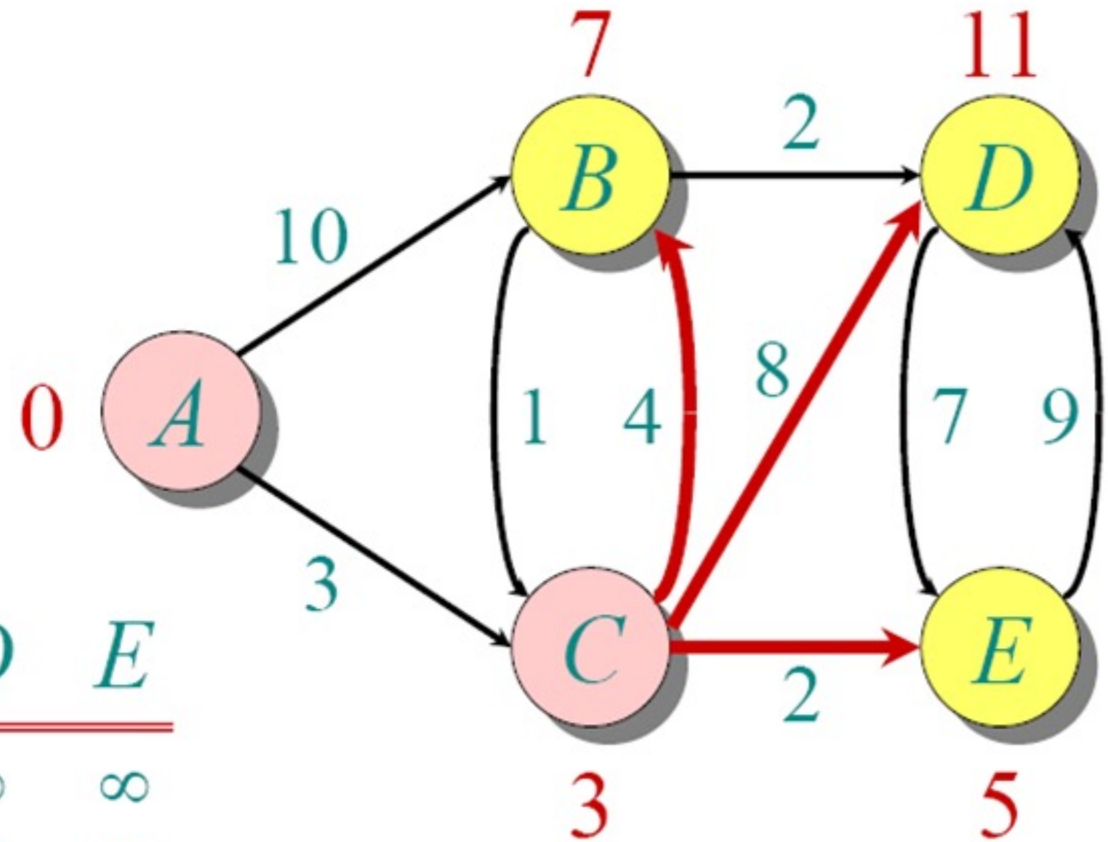


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

$S: \{ A, C \}$

- By Laksman Veeravagu and Luis Barrera

Dijkstra's algorithm

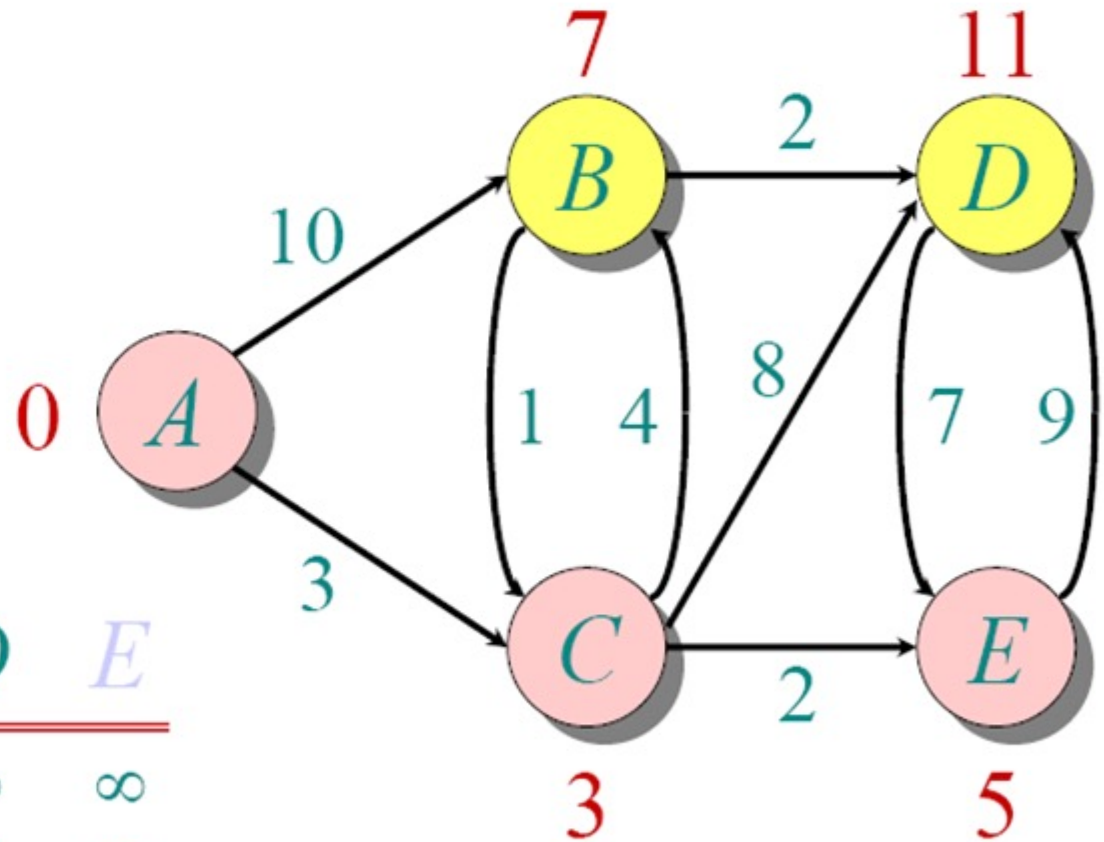


Q:	A	B	C	D	E
	$\boxed{0}$	∞	∞	∞	∞
		10	$\boxed{3}$	∞	∞
		7		11	5

S: { A, C }

- By Laksman Veeravagu and Luis Barrera

Dijkstra's algorithm



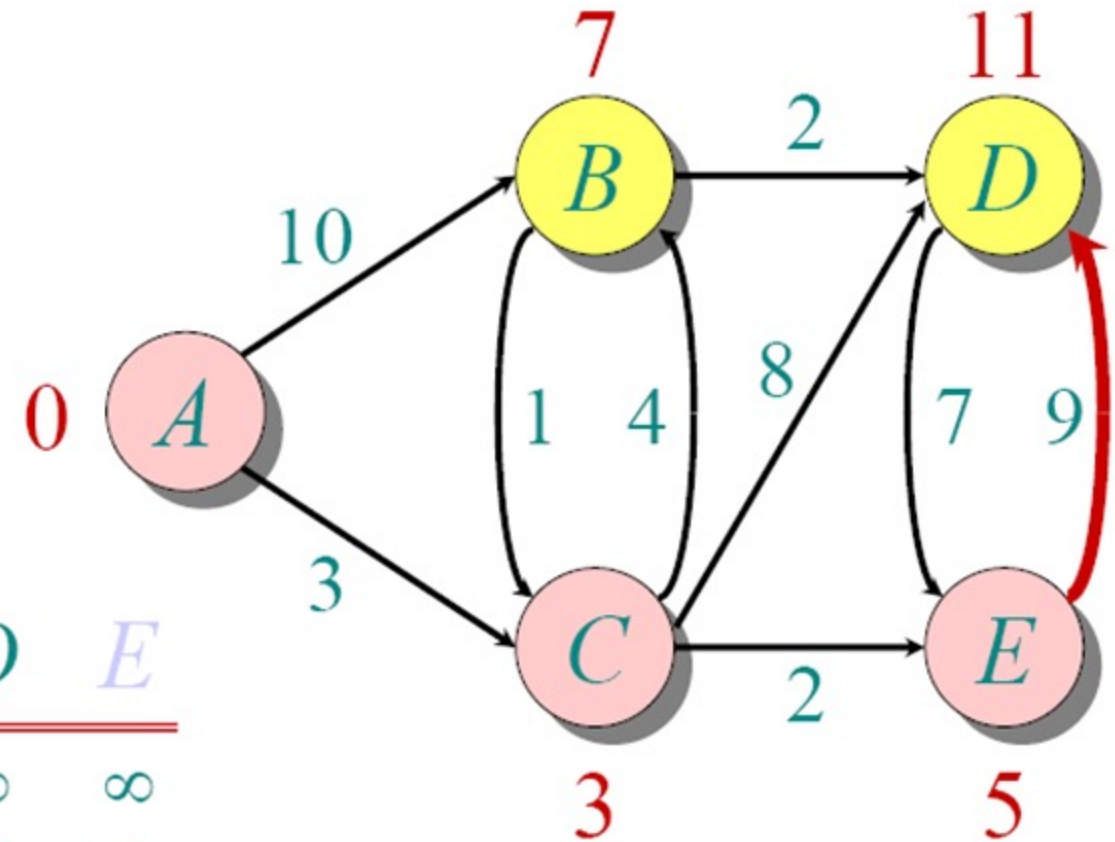
Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5

S: { A, C, E }

bristol.ac.uk

- By Laksman Veeravagu and Luis Barrera

Dijkstra's algorithm

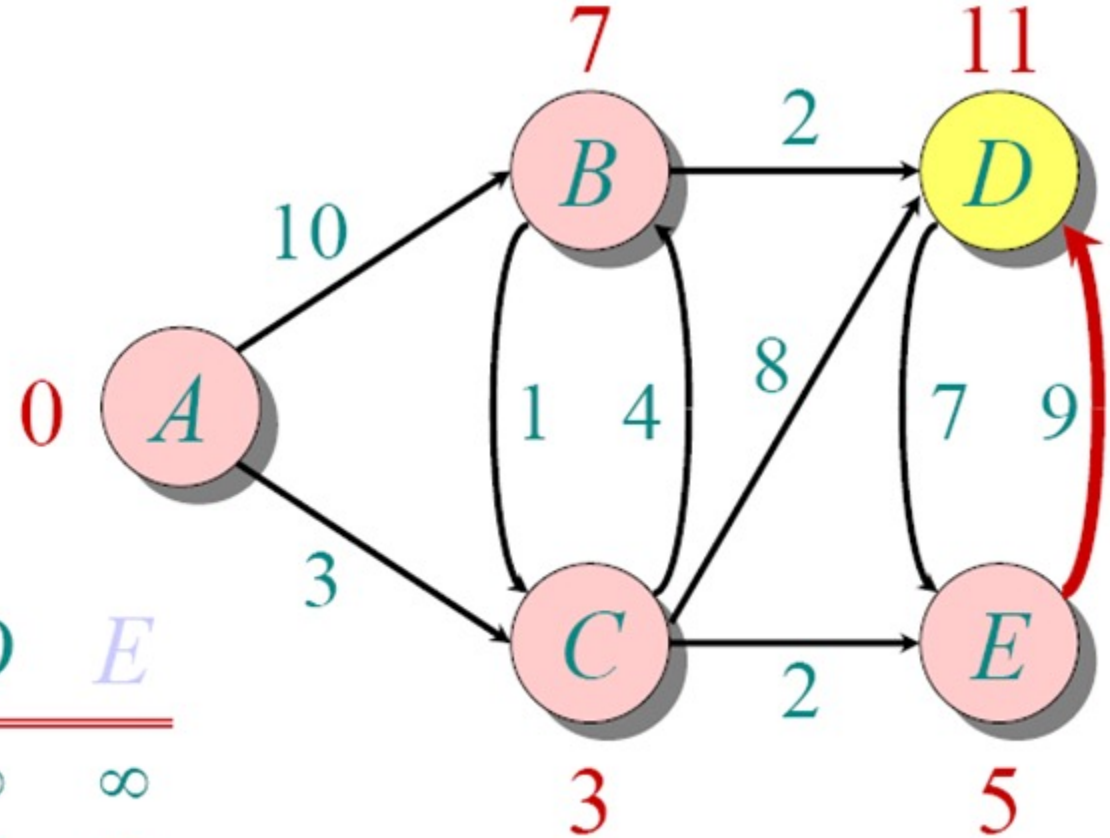


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	

$S: \{ A, C, E \}$

- By Laksman Veeravagu and Luis Barrera

Dijkstra's algorithm

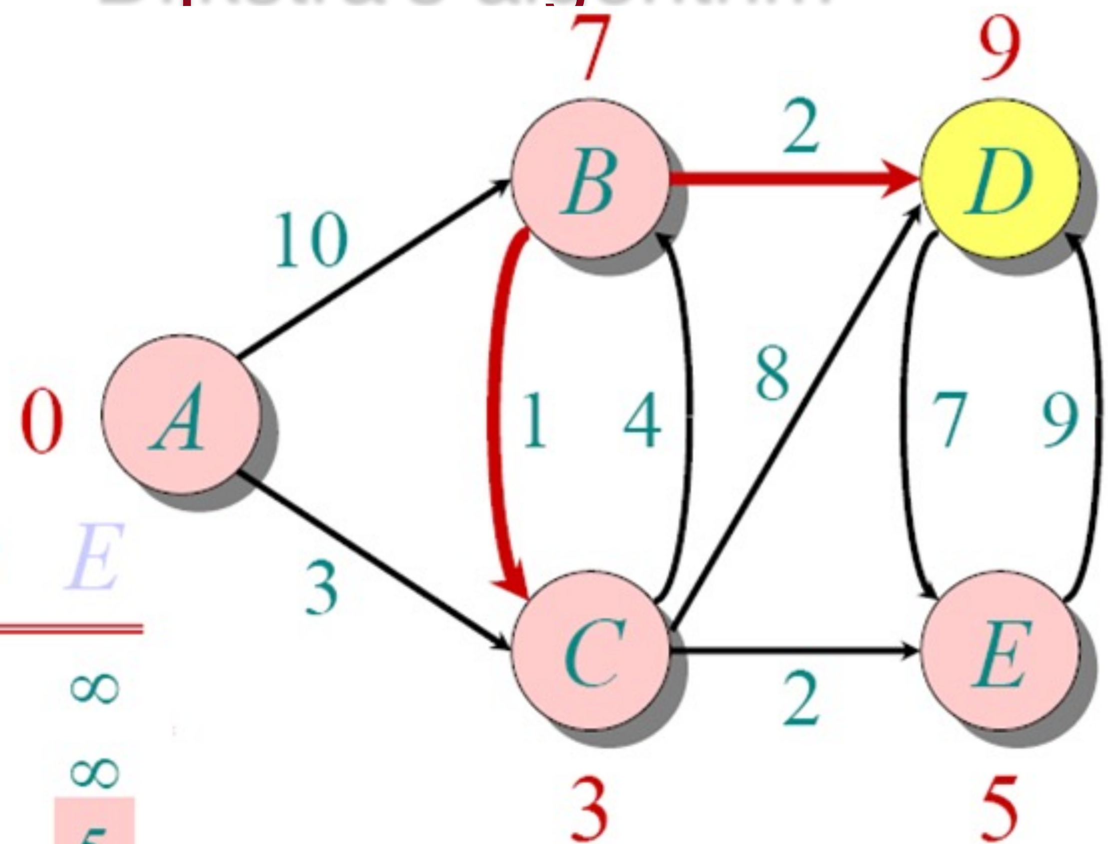


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	

S: { A, C, E, B }

- By Laksman Veeravagu and Luis Barrera

Dijkstra's algorithm



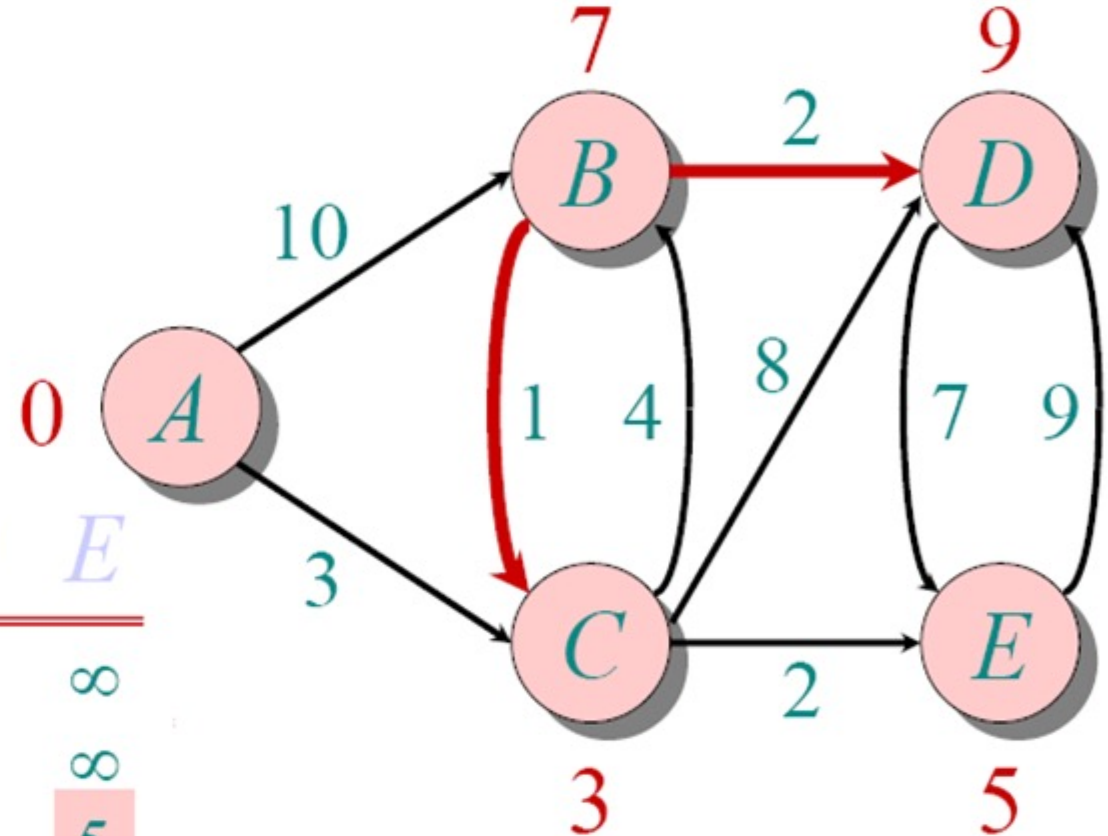
Q:

A	B	C	D	E
$\boxed{0}$	∞	∞	∞	∞
	10	$\boxed{3}$	∞	∞
	7		11	$\boxed{5}$
	$\boxed{7}$		11	
			9	

S: { A, C, E, B }

- By Laksman Veeravagu and Luis Barrera

Dijkstra's algorithm



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	
				9	

S: { A, C, E, B, D }

- By Laksman Veeravagu and Luis Barrera

Dijkstra's algorithm - Pseudocode

```
dist[s] ←0                                  (distance to source vertex is zero)
for  all v ∈ V–{s}
        do  dist[v] ←∞                      (set all other distances to infinity)
S←∅                                         (S, the set of visited vertices is initially empty)
Q←V                                         (Q, the queue initially contains all vertices)
while Q ≠∅                                  (while the queue is not empty)
do   u ← mindistance(Q,dist)                (select the element of Q with the min. distance)
     S←S∪{u}                                (add u to list of visited vertices)
      for all v ∈ neighbors[u]
            do  if   dist[v] > dist[u] + w(u, v)          (if new shortest path found)
                      then      d[v] ←d[u] + w(u, v)   (set new value of shortest path)
                 (if desired, add traceback code)
return dist
```

Dijkstra's algorithm: Discussion

Algorithm complexity: n nodes

- Each iteration: need to check all nodes,
- n*(n+1)/2 comparisons: $O(n^2)$
- More efficient implementations possible: $O(n \cdot \log n)$

Oscillations possible:

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

Comparison: LS vs DV

Complexity

- LS: with n nodes, E links, O(nE) msgs sent each
- DV: exchange between neighbors only

Speed of Convergence

- LS: $O(n^2)$ algorithm requires O(nE) msgs
  - may have oscillations
- DV: convergence time varies
  - may have routing loops
  - count-to-infinity problem

Robustness: what happens if router malfunctions?

LS:

- node can advertise incorrect link cost
- each node computes only its own table

DV:

- Node can advertise incorrect path cost
- Each node's table used by others
- Errors propagate

So far..

- Revised basic Internet routing algorithms
- Gained understanding by example
- What next?
  - Implementations in the Internet

Routing in the Internet

Unicast Route Discovery Protocols

- How to build routing tables
- Network structure - autonomous systems
- Examples
  - Routing Information Protocol (RIP, distance vector)
  - Open Shortest Path First (OSPF, link state)
  - Border Gateway Protocol (BGP)
  - More…

Hierarchical Routing

Our routing study so far: Idealisation

- All routers identical
- The network is "flat”

Not true in practice. Why?

Scale: millions of destinations

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

Administrative autonomy

- Internet = network of networks
- Network admin wants to control routing within own network

Hierarchical Routing

Autonomous System

- Aggregation of routers into regions
- One routing protocol within an AS (called Intra-AS routing protocol)
- routers in different AS can run different intra-AS routing protocol

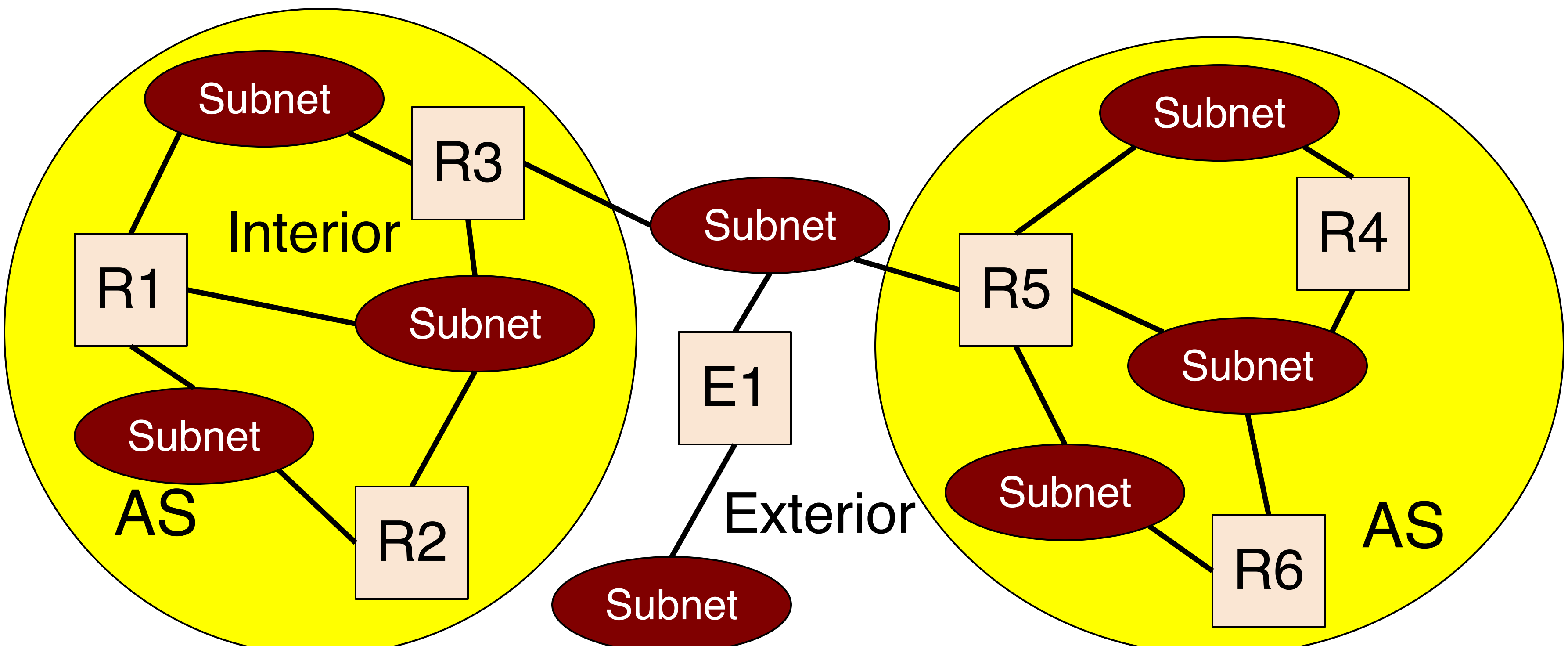
Gateway routers

- Special routers in AS
- Run intra-AS routing protocol with all other routers in AS
- But *also* responsible for routing to destinations outside AS

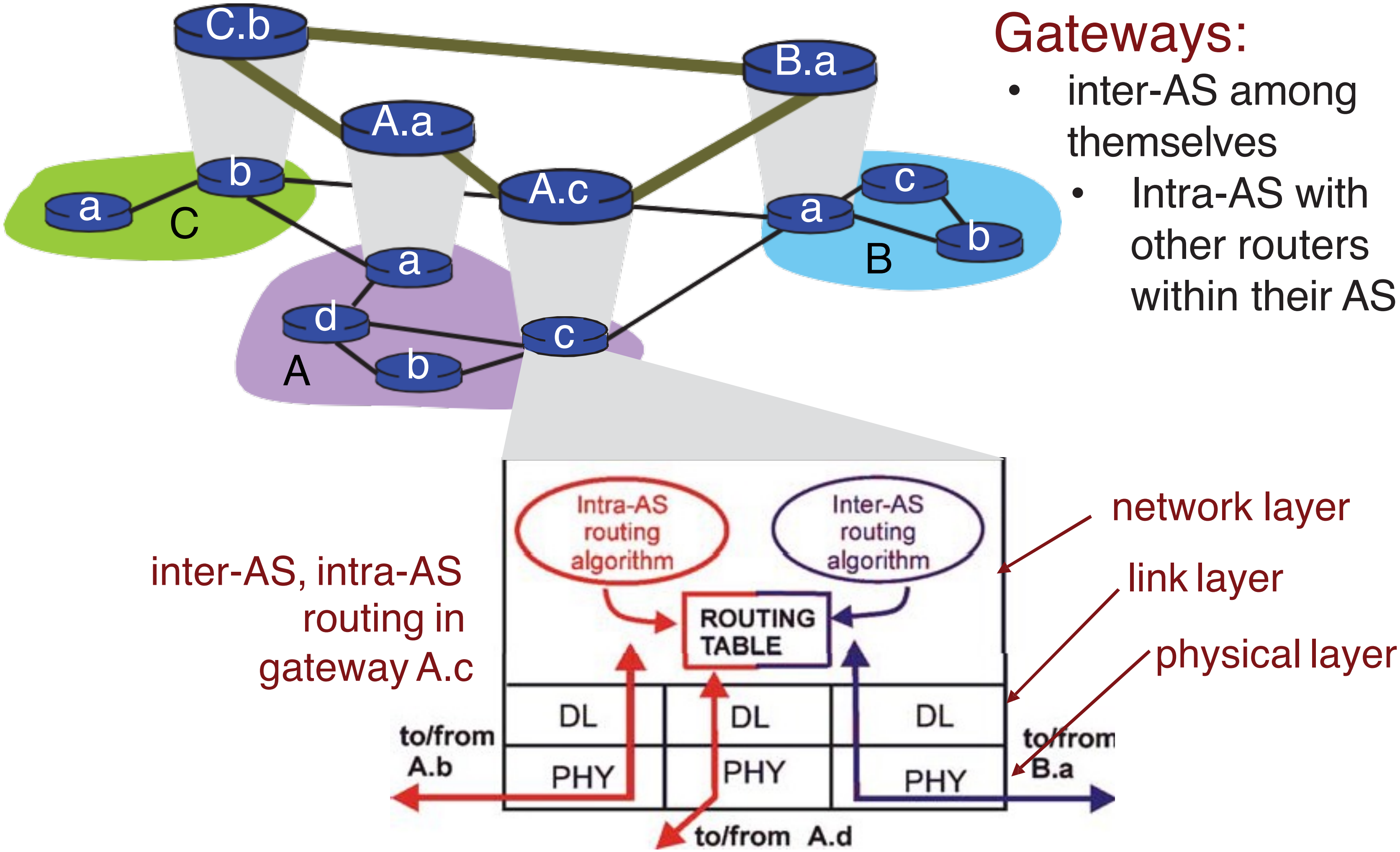
inter-AS routing protocol with other gateway routers

Autonomous System

A collection of Internet subnets connected by routers under a single administration



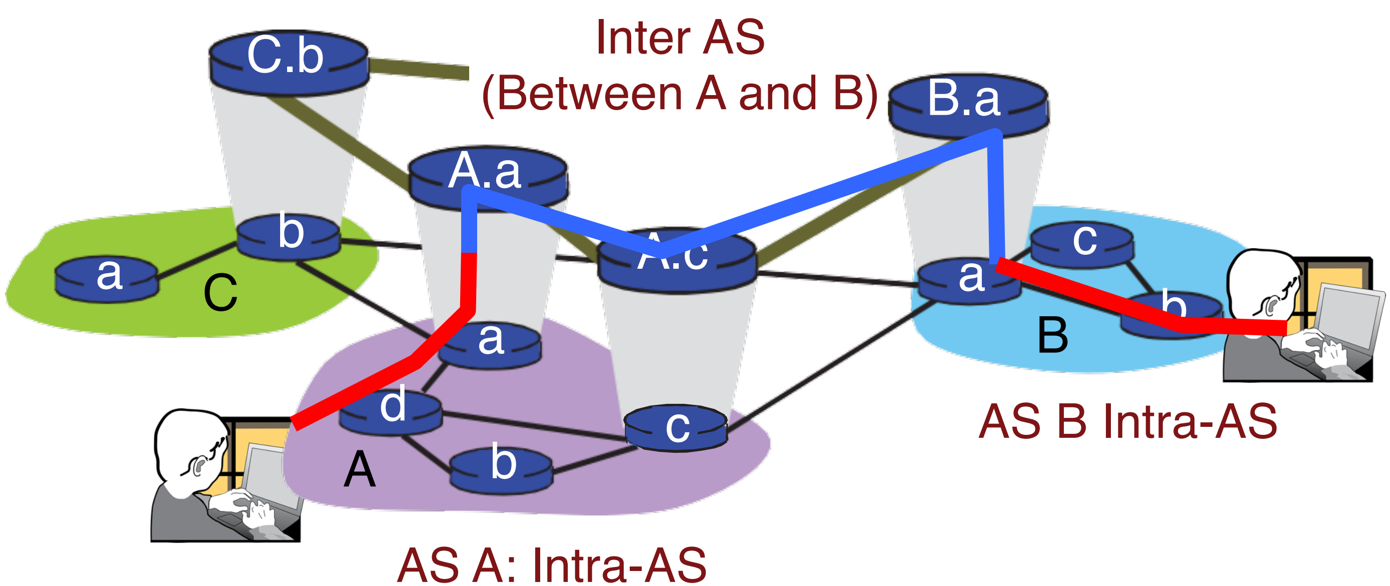
Intra-AS and Inter-AS Routing



Gateways:

- inter-AS among themselves
- Intra-AS with other routers within their AS

Intra-AS and Inter-AS Routing



University of BRISTOL

Routing Protocols

- Interior Gateway Protocol (IGP) between routers within ASs
  - RIP, OSPF, Bellman Ford, Dijkstra
- Exterior Gateway Protocol (EGP) between routers outside ASs
  - Border Gateway Protocol (BGP) between gateways at the AS edge
- …

Routing Information Protocol (RIP)

- Distance Vector (e.g. Bellman Ford)
- Defined in RFC1058
- Intra-AS routing protocol
- Used by hosts (passively) and routers
- Delete old entries that do not respond, update every 30 sec
- Request/response protocol runs over UDP, local to subnetwork

Routing Information Protocol (RIP)

Distance Metric

"The metric of a network is an integer between 1 and 15 inclusive. It is set in some manner not specified in this protocol.

Most existing implementations always use a metric of 1.

New implementations should allow the system administrator to set the cost of each network."

RFC 2453

Routing Information Protocol (RIP)

- Only hop count matters, speed does not!
- Max number of hops is 15
- Bandwidth intensive
(routing tables updated every 30sec routinely, every 1-5 seconds when triggered)
- Long messages, so multiple UDP packets - unreliable?
- Local errors are propagated throughout the routing tables
- Later versions are more efficient but algorithmic problems remain.

Open Shortest Path First (OSPF)

Open Shortest Path First (OSPF)

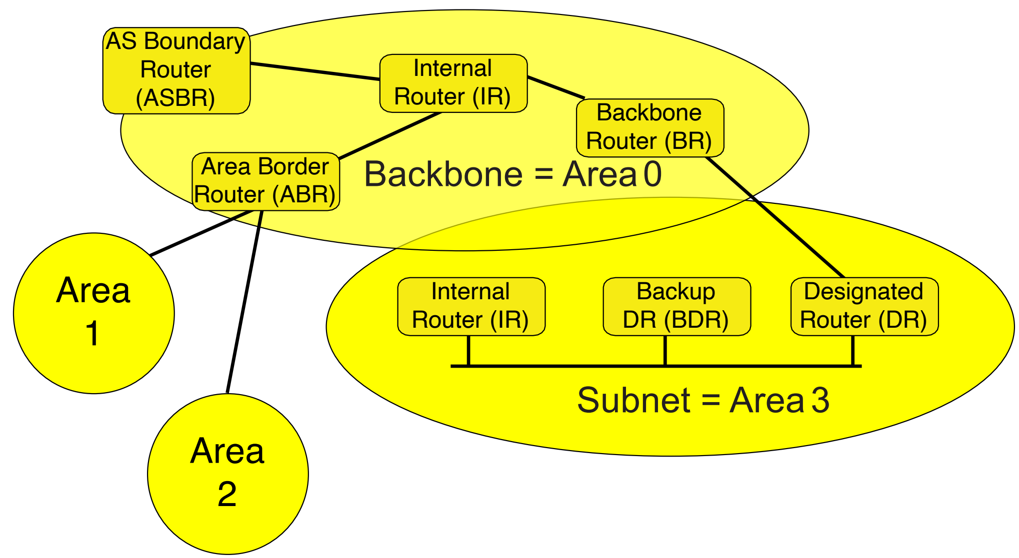
- Link-State Protocol (e.g. Dijkstra)
- Defined in RFC2328
- Intra-AS routing protocol
- Wider range of metrics
- Multiple paths: Load-Balancing
- Understands service quality

Router Types

- Internal Router (IR)
  All interfaces connected to the same area
- Area Border Router (ABR)
  Interfaces between multiple areas
- Backbone Router (BR)
  Connects to the backbone
- Autonomous System Boundary Router (ASBR)
  Exchanges information with other ASBRs
- Designated Router (DR)
  Generates link state information about local subnets
- Backup DR

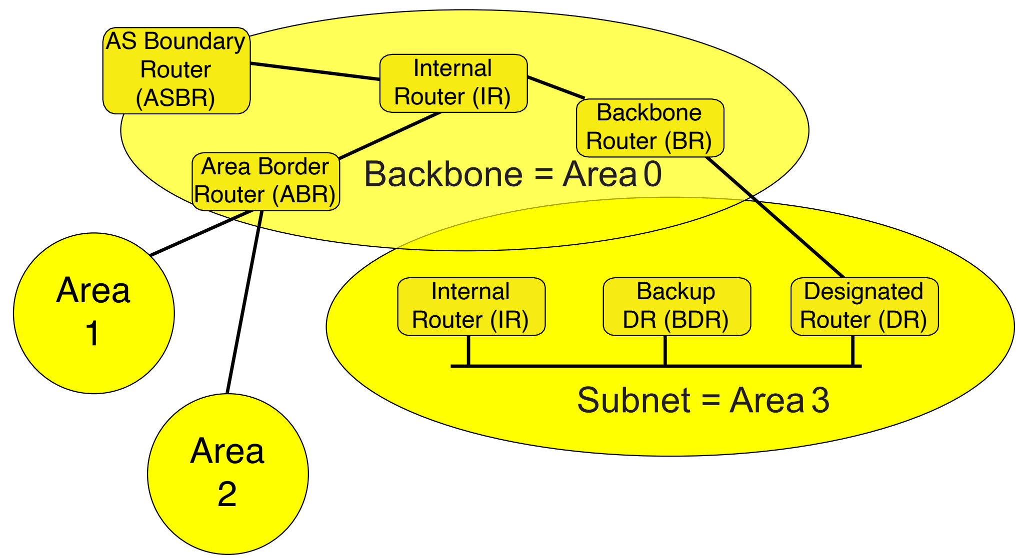
OSPF Areas and Hierarchies

- Large networks may be divided into multiple areas interconnected by backbone routers to reduce routing traffic
- ABRs discover the area topology and advertise it to the BRs who then forward it to other areas
- ABRs execute the OSPF(e.g. Dijkstra) algorithms
- ASBRs use EGP (e.g. BGP)to exchange routing information with other AS



OSPF Table Maintenance

- Routers multicast a HELLO message to all OSPF routers
(apr. every 40 sec)
  - Detect neighbours and elect DRs and BDRs
  - Informs of DR selections, neighbour routers
- Establishing adjacency between
  - Two routers directly connected to one another
  - DR/BDR routers on a LAN (not any others)
- Adjacent routers must synchronise their routing databases
  - Send list of LSAs received from neighbour routers
  - Compare and request missing entries
  - When all the same, execute Dijkstra to choose between paths



Open Shortest Path First (OSPF)

- Can import routes from other ASs
- Authenticates partner routers (security)
- Multicast for announcements
- Lower bandwidth usage
- Compact messages
- Imposes more structure on routers

OSPF Link State Advertisement Protocol

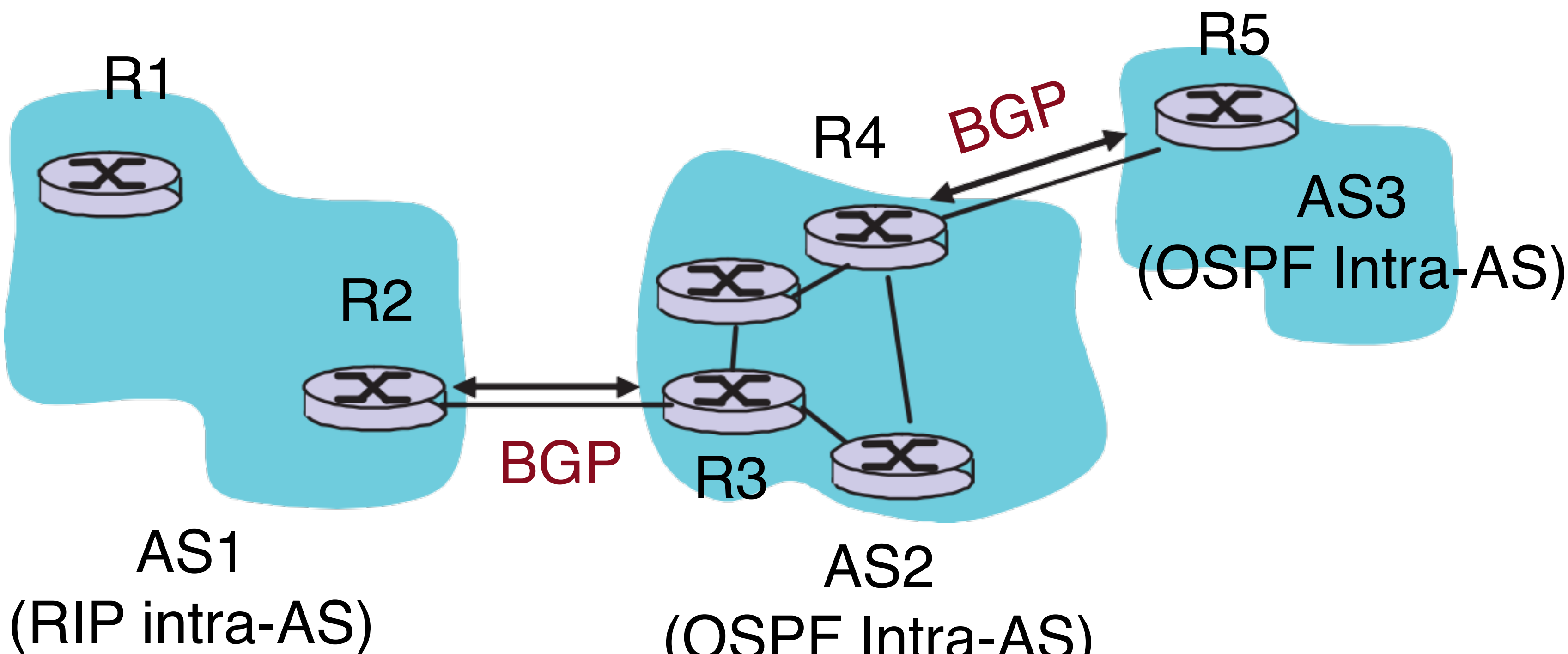
- Link State Advertisements (LSAs) for address and cost
  - Router LSAs flooded by all routers in an area
  - Network LSAs for all routers flooded by the DR
  - Summary LSAs flooded into area by ASBRs, describing reachable networks in other areas
  - More LSA types defined in additional RFCs
- All LSAs have 32 bit sequence numbers for duplicate detection
- All entries expire after a certain time
- Cost metric is related to speed
  - 19.2 Kbps = 5208, 100 Mbps = 1

Border Gateway Protocol (BGP)

Border Gateway Protocol (BGP)

- RFC 1267
- Runs on TCP for reliability
- It is a distance vector protocol (well… kind of…)
  - Keeps a record and advertises entire paths to destination
- Purpose is to maintain inter-AS routes and inform/reroute quickly when there are problems
- Has to cater for different set of criteria ("politics")
- Advertises transit ASs on paths to a destination - if there is more than one path then a router can choose “the best”
- Loop-Free!, no RIP-like problems
- BGP exchanges complete tables followed by update

Inter-AS routing in the Internet: BGP



University of BRISTOL

Internet inter-AS routing: BGP

BGP (Border Gateway Protocol): the de facto standard

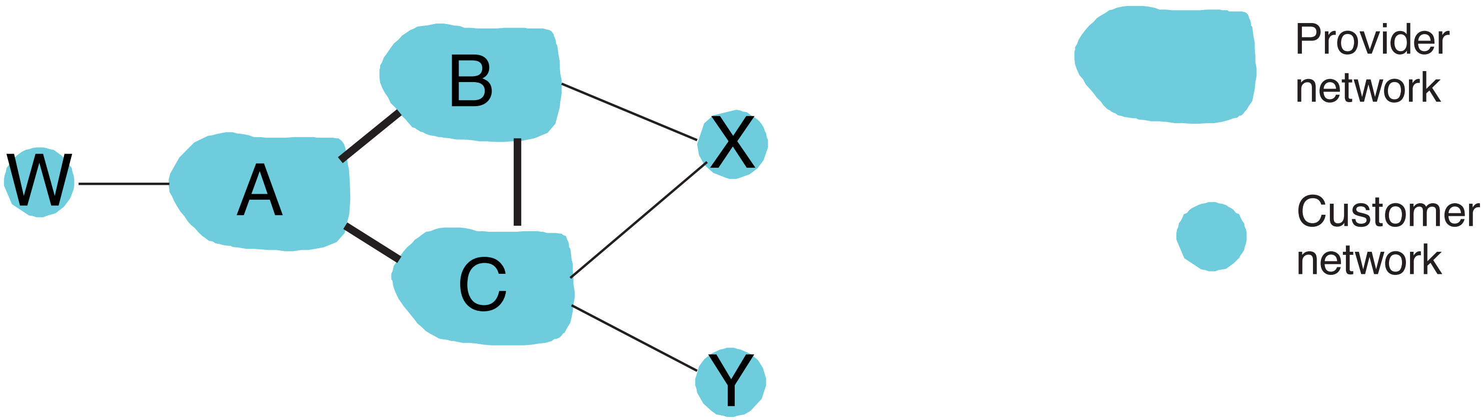
BGP is a Path Vector protocol:

- similar to Distance Vector protocol, but
- each Border Gateway advertises the entire path to its neighbouring peers
  - sequence of AS's to destination
- BGP routes to networks (ASs), not individual hosts

E.g., Gateway X may send its path to dest. Z:

```
Path (X,Z) = X, Y1, Y2, Y3, … , Z
```

BGP
Controlling who routes to you



- A, B, C are provider networks
- X, W, Y are customer (of provider networks)
- X is dual-homed: attached to two networks
  - X does not want to route from B via X to C
  - so X will not advertise to B a route to C

Internet inter-AS routing: BGP

Assume:

- Doman W wants to reach to domain Z

Gateway X (domain X) send its path to Z to peer gateway W

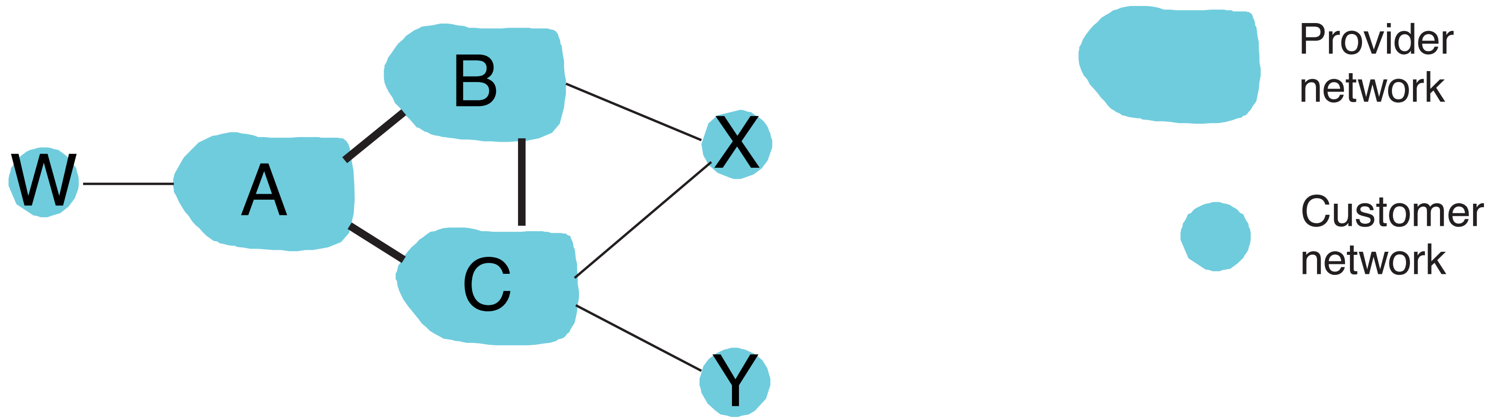
- W may or may not select path offered by X
  - Reasons: cost, policy (don't route via competitors AS), loop prevention
- If W does select path advertised by X, then:

Path (W,Z) = w, Path (X,Z)

Note: X can control incoming traffic by controlling its route advertisements to peers:

- e.g., don't want to route traffic to Z -> don't advertise any routes to Z

BGP: Controlling who routes to you



- A advertises to B the path AW
- B advertises to X the path BAW
- Should B advertise to C the path BAW?
  - No way! B gets no "revenue" for routing CBAW since neither W nor C are B's customers
  - B wants to force C to route to w via A
  - B wants to route only to/from its customers!

BGP operation

- Q: What does a BGP router do?
- Receives and filters route advertisements from directly attached neighbour(s).
- Selects routes
  - To route to destination X, which path (of several advertised) will be taken?
- Sends route advertisements to neighbors.

BGP messages

- BGP messages exchanged using TCP.
- BGP messages:

  `OPEN`: opens TCP connection to peer and authenticates sender

  `UPDATE`: advertises new path (or withdraws old)

  `KEEPALIVE`: keeps connection alive in absence of UPDATES; also ACKs OPEN request

  `NOTIFICATION`: reports errors in previous msg; also used to close connection

Why different Intra- and Inter-AS routing ?

Policy:

- Inter-AS: admin wants control over how its traffic is routed, who routes through its net.
- Intra-AS: single admin, so no policy decisions needed

Scale:

- Hierarchical routing saves table size, reduced update traffic

Performance:

- Intra-AS: can focus on performance
- Inter-AS: policy may dominate over performance

Conclusions

- Study typical Internet routing protocols, discuss structural issues

Outcomes

- Explain difference between algorithm and protocol
  - RIP (Distance vector)
  - OSPF (Link state)
- Demonstrate operation of typical protocols by example in simple networks
- Explain terminology
  - Distance vector, link state, .
- IGP, EGP, BGP, .
- Areas, ASs, .

Reading

- A.S. Tanenbaum, "Computer Networks" 3rd Ed. - Chapter 5.

Or

- L.L. Peterson & B.S. Davies "Computer Networks - A Systems Approach". Chapter 4.

Or

- Kurose, J., "Computer Networks: A Top-Down Approach featuring the Internet", Chapter 4.
- RFC1058 at http://www.ietf.org/rfc.html (original RIP specification - historic status.
- Current RIP standard is described in RFC2453)
- RFC2328 - OSPF version 2 (standard).

University of BRISTOL

Next Up

1. Introduction
2. Internet Routing and Switching
3. IP Multicast
4. Networking for Realtime Applications
5. Routing in Wireless Networks
6. Quality of Service