Computer Networks Advanced Course

The Network Layer

Routing, OSPF

Barak Gonen

Routing Principles

Barak Gonen

Topics

- Router
- Routing table
- Routing protocols
  - Link state vs distance vector

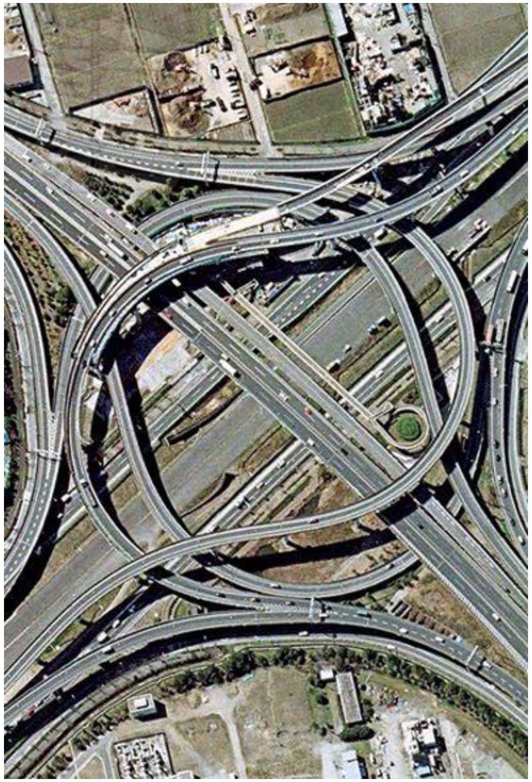
Router

Router used by ISP,
made by Cisco & Fujitso

Home router, TP-Link

Router Tasks

- Routing – find the best path to destination
- Forwarding – connect networks
- Routers are typically filled with more capabilities:
  - DHCP server
  - NAT
  - Local DNS server
  - etc
  - We shall focus only on core router tasks



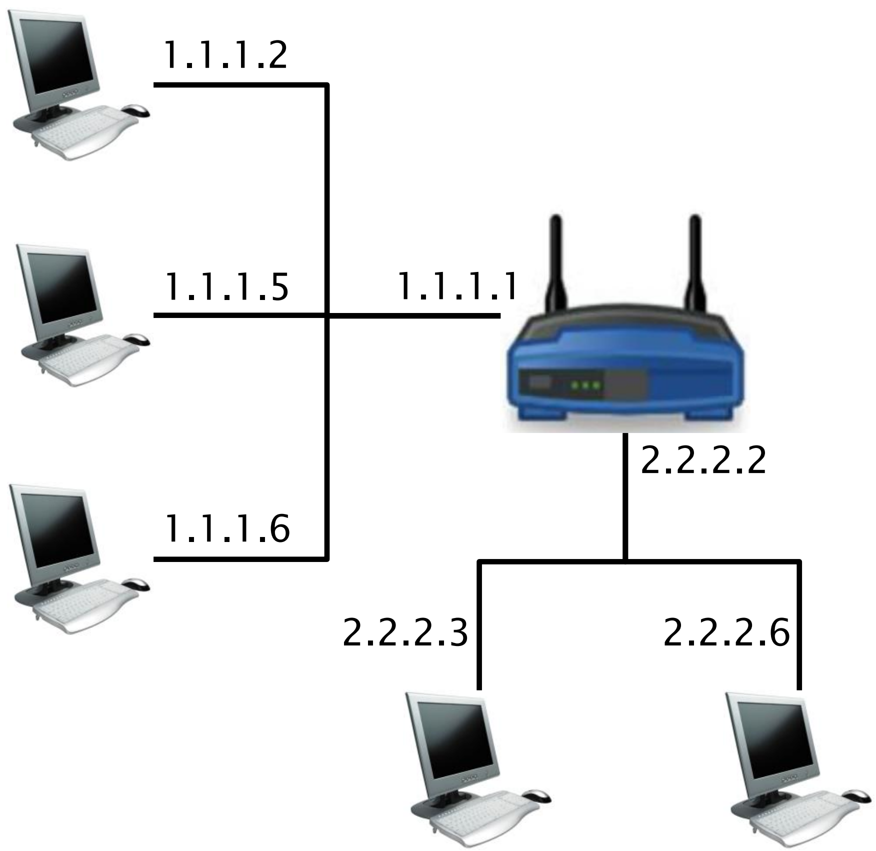
Highway 9. Japan

Network Interface Card

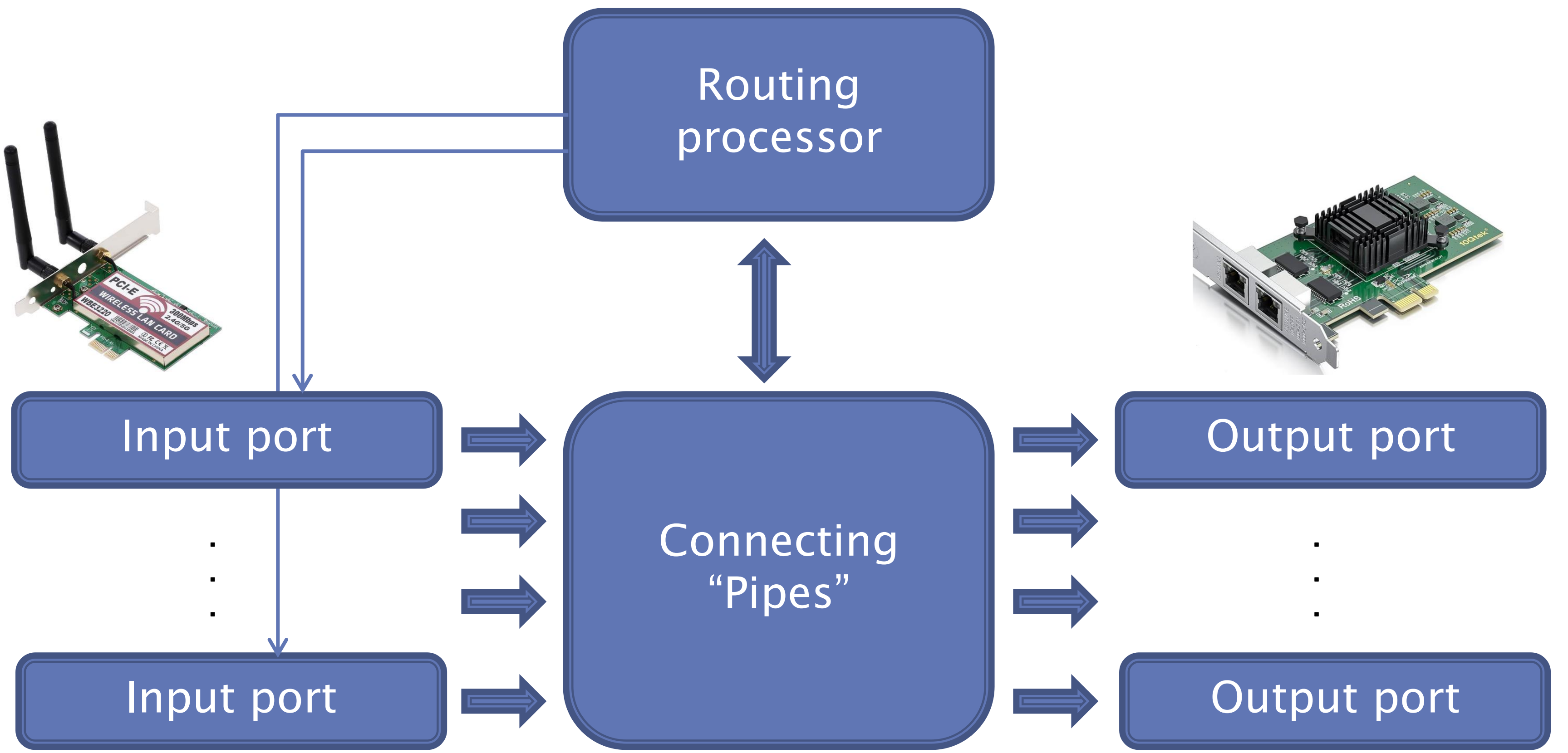
- May have one or more ports

Intra Network Connectivity

- Two networks are connected by a router:
  - 1.1.1.0/24
  - 2.2.2.0/24
- The router has a NIC in each network
  - Each NIC has an IP
    - 1.1.1.1
    - 2.2.2.2
- Since the router is connected to both networks, it can handle connectivity

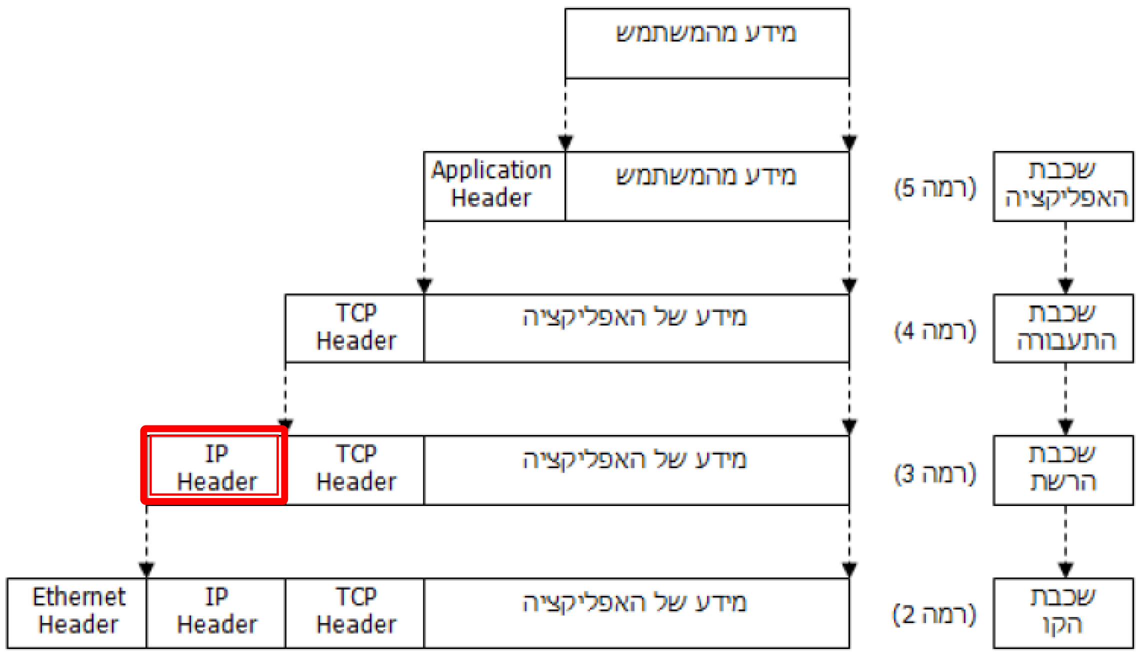


Router Architecture



Input Ports

- Handle PHY layer and encapsulation
- Extract dest. IP

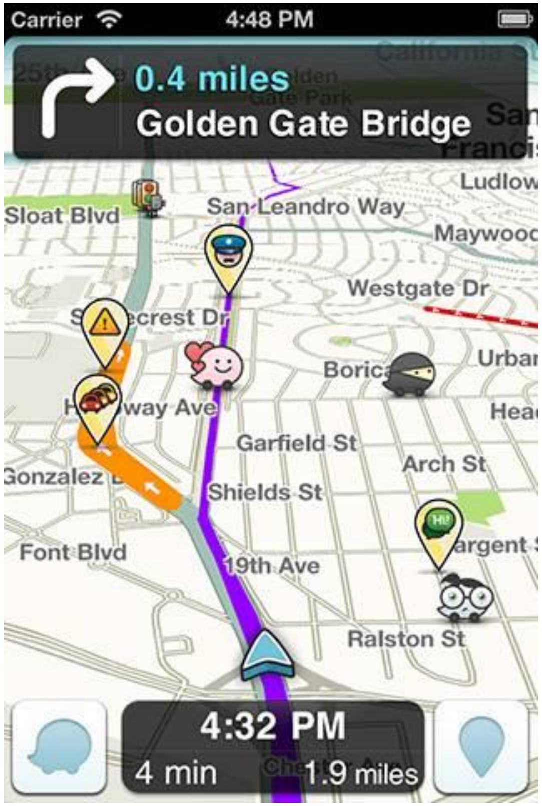


Output ports

- Opposite tasks of input ports
- The input is layer 3 and above
  - Add data link layer
  - Transmit using PHY layer

Routing Processor

- Sets the route for each packet
  - “Packets destined to 1.2.3.4 forward to output port #6”
- Route is deduced from routing table
- Routing table is created by routing algorithms
  - We shall review an example



Routing Table

- Assume network:
  - Router R1 knows 192.168.0.0/16 and R2
  - Router R3 knows 100.200.0.0/16 and R2
  - R2 is connected to 3 networks, has 3 IP addresses
- How will R2 forward a packet who's dest. IP is 100.200.5.8?

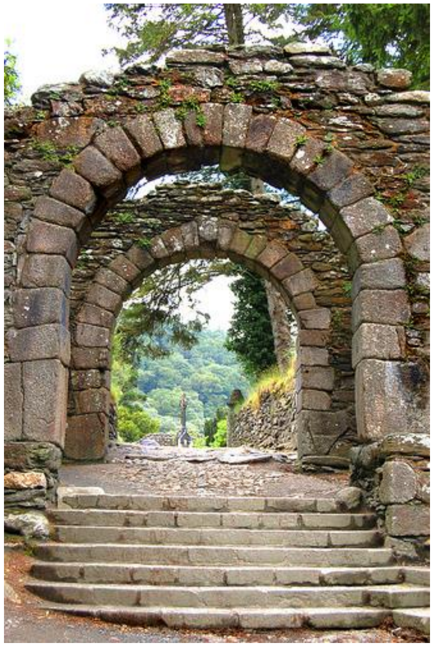
Routing Table

- Assume R2 routing table is

מספר שורה	Network) יעד (Destination	Network) מסכת רשת (Mask	(Interface) ממשק
1	0.0.0.0	0.0.0.0	3.3.3.3
2	192.168.0.0	255.255.0.0	1.1.1.1
3	100.200.0.0	255.255.0.0	2.2.2.2

- Router will scroll from *bottom up* and look for matches
- IP dest 100.200.5.8 will match line #3
  - Interface 2.2.2.2

Routing Table

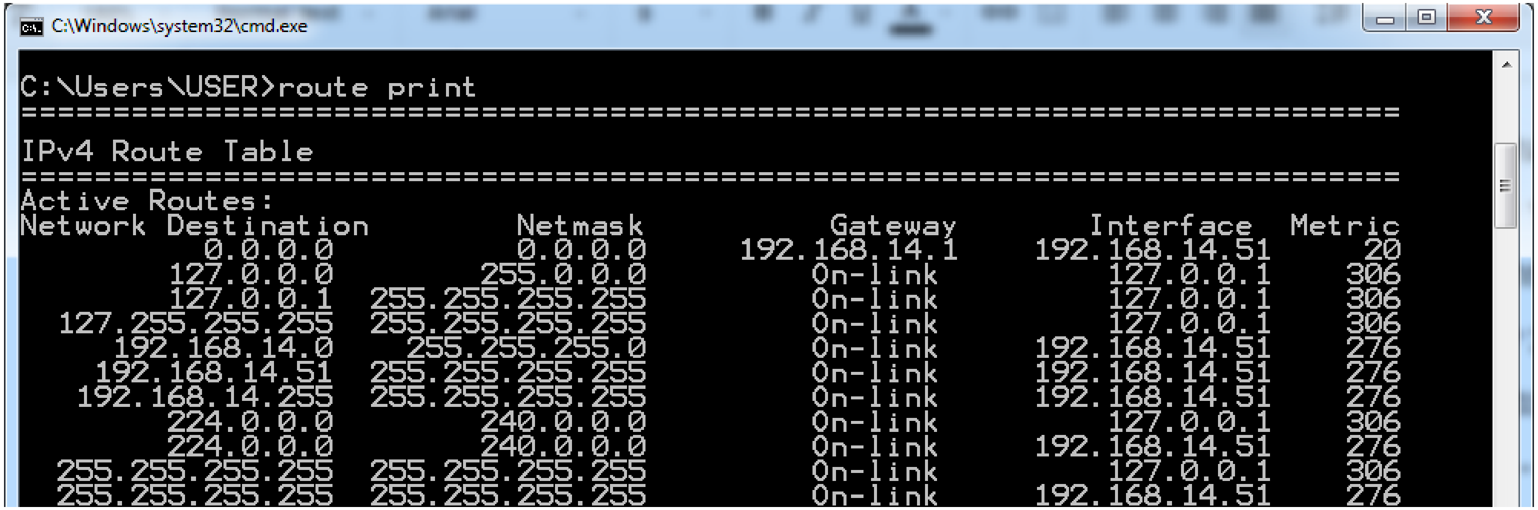


- What will be the interface for 192.168.6.6?
- For 5.5.5.5?
- The rule in line #1 captures and packet
  - **Default Gateway**

מספר שורה	יעד (Destination Network)	מסכת רשת (Network Mask)	ממשק (Interface)
1	0.0.0.0	0.0.0.0	3.3.3.3
2	192.168.0.0	255.255.0.0	1.1.1.1
3	100.200.0.0	255.255.0.0	2.2.2.2

Route Print

- Cmd -> route print

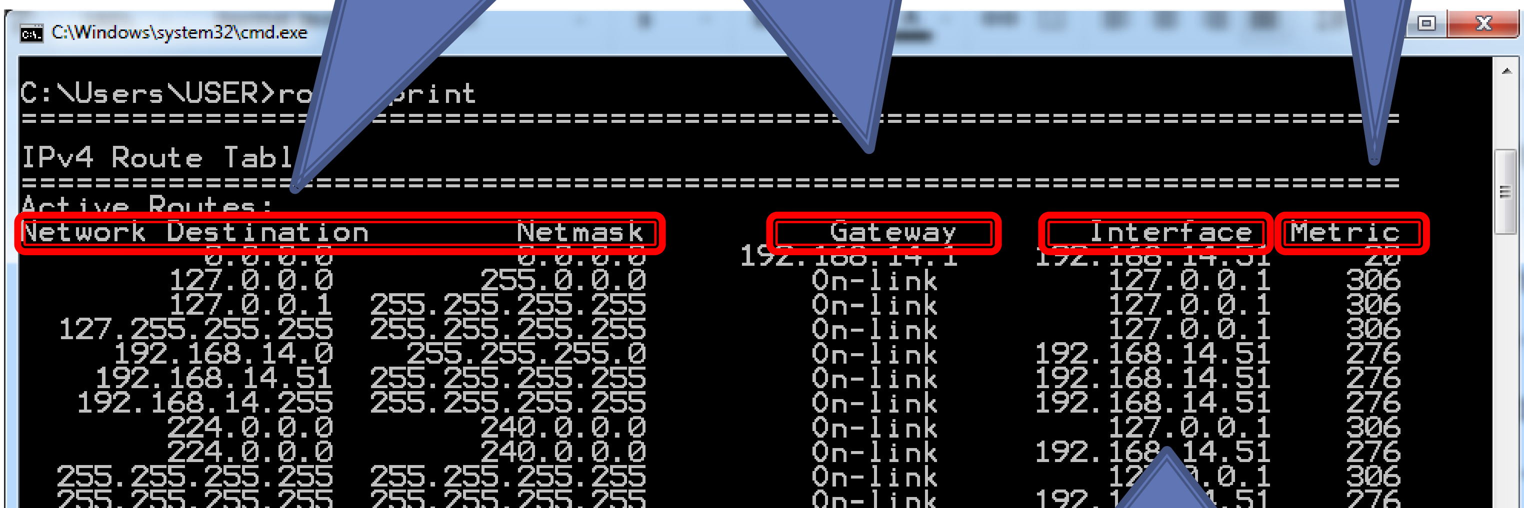


Route Print

Network dest + Netmask are the condition checked

IP address of router used to access the network

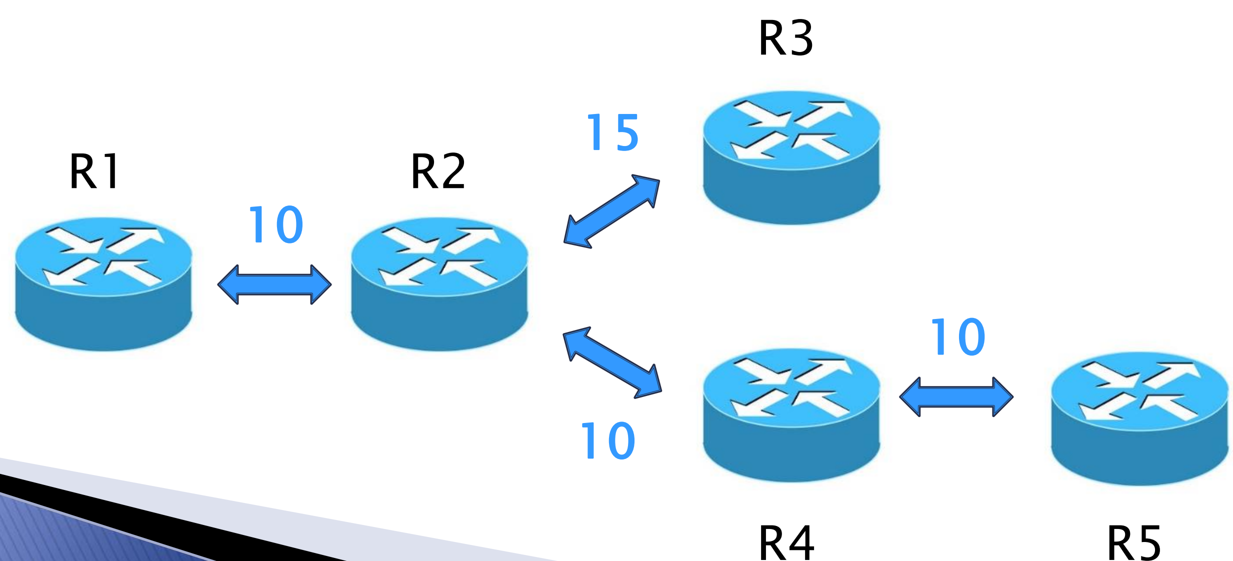
Cost to dest



IP address of NIC for outgoing packet

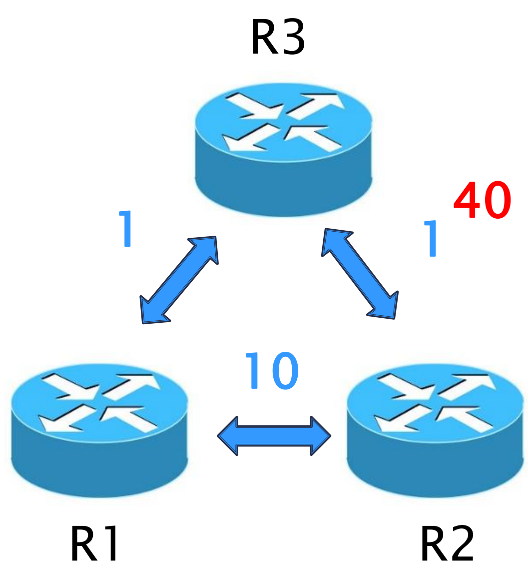
Routing Protocols

- Distance Vector (DV) – each node has information only about it's neighbors
  - R1 knows "R2 can get to R5 cost 20"
- Link State (LS) – each node has full network link data
  - R1 know "R2 can get to R4 cost 10, R4 can get to R5 cost 10"



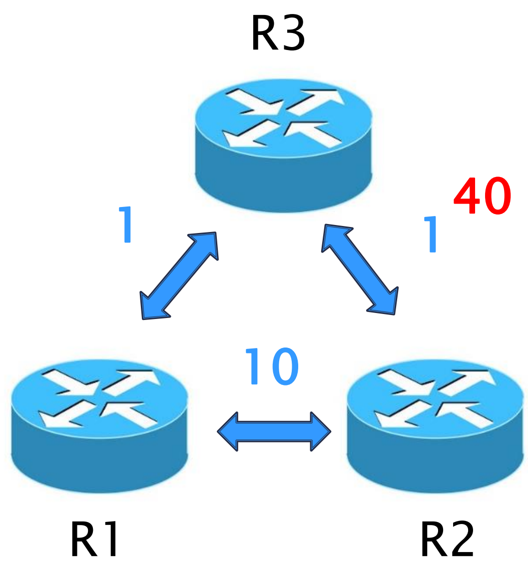
Routing Protocols

- Example to clarify the difference
- Suppose R1-R2 link changes to 40, what will change in LS? DV?



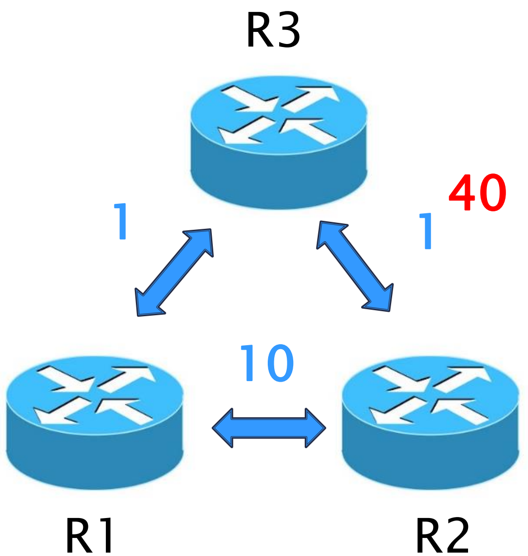
Routing Protocols – LS

- R3 to R1: "Update: my direct connection to R2 costs 40"
- R1 recalculates best route to R2
  - No longer cost 2



Routing Protocols – DV

- R3 to R1: “Update: I can get to R2, cost 3”
  - Why? R1 told R3 it can get to R2 cost 2…
  - Plus 1 for R3->R1
- R1 to R3: “Update: I can get to R2 cost 4”
  - Why? R1->R3 is 1, R3->R2 is “3”
- …And so on until finally R1->R2 is better



Routing Protocols

- DV – updates only from neighboring nodes
  - Relatively few updates
  - Changes propagate slowly
  - Efficient for large networks
- LS – all nodes share the same link information
  - Relatively many updates
  - Changes propagate fast
  - Not efficient in large networks

OSPF Routing Protocol

Open Shortest Path First

Barak Gonen

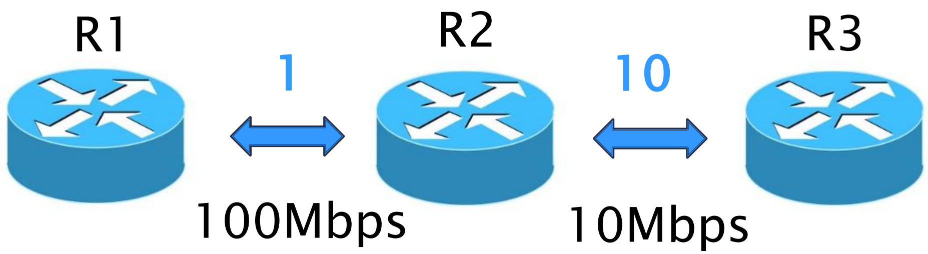
Topics

- Metric
- Link State Advertisements
- Areas
- Minimal cost algorithm – Dijkstra

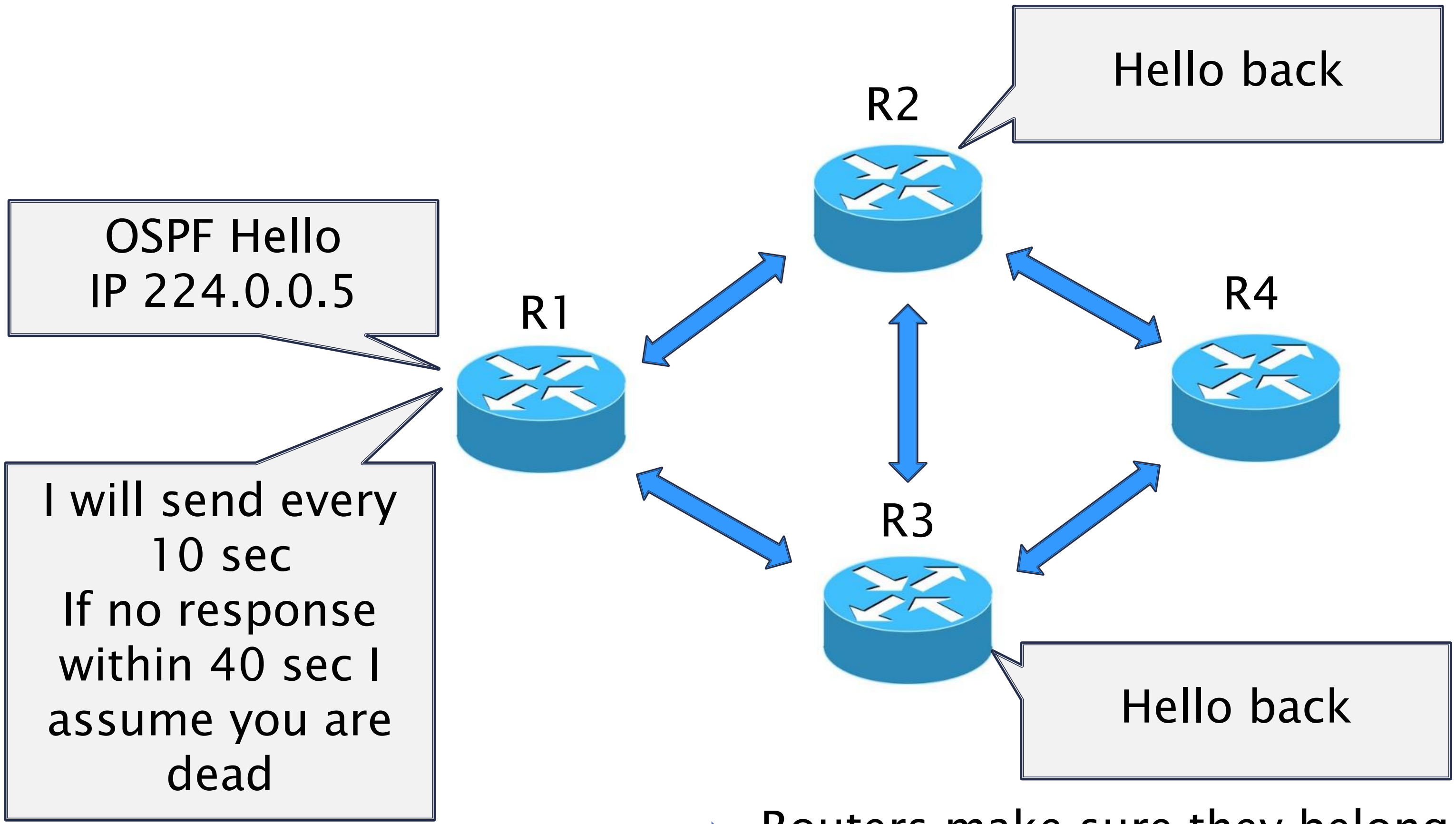
- Note: this is not a full OSPF review, goal is only to get familiar with routing protocol concepts
  - OSPF Areas not fully discussed
  - OSPF Hello simplified
    - “Down” to “Full” not discussed

Metric

- Routers' PHY operate at max possible speed that link / other side allows
  - R1: "My link with R2 is 100Mbps"
- Cost: 100 / link speed
  - 100 Mbps -> cost 1
  - 10 Mbps -> cost 10
  - 5 Mbps -> cost 20
  - etc
  - The default 100 can be changed
- Cost is summed up
  - R1: "My cost to reach R3 is 11"

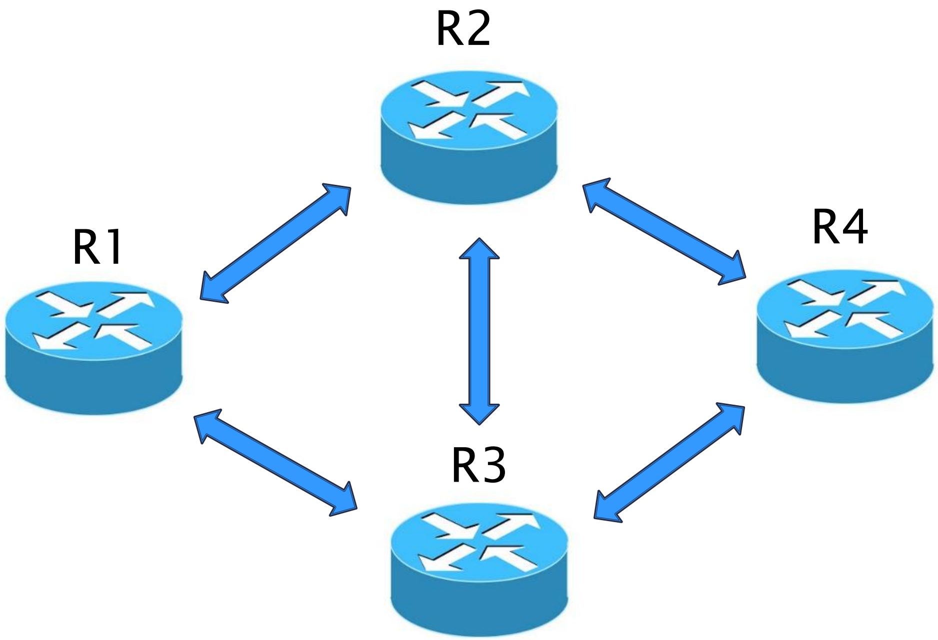


Neighbors Table



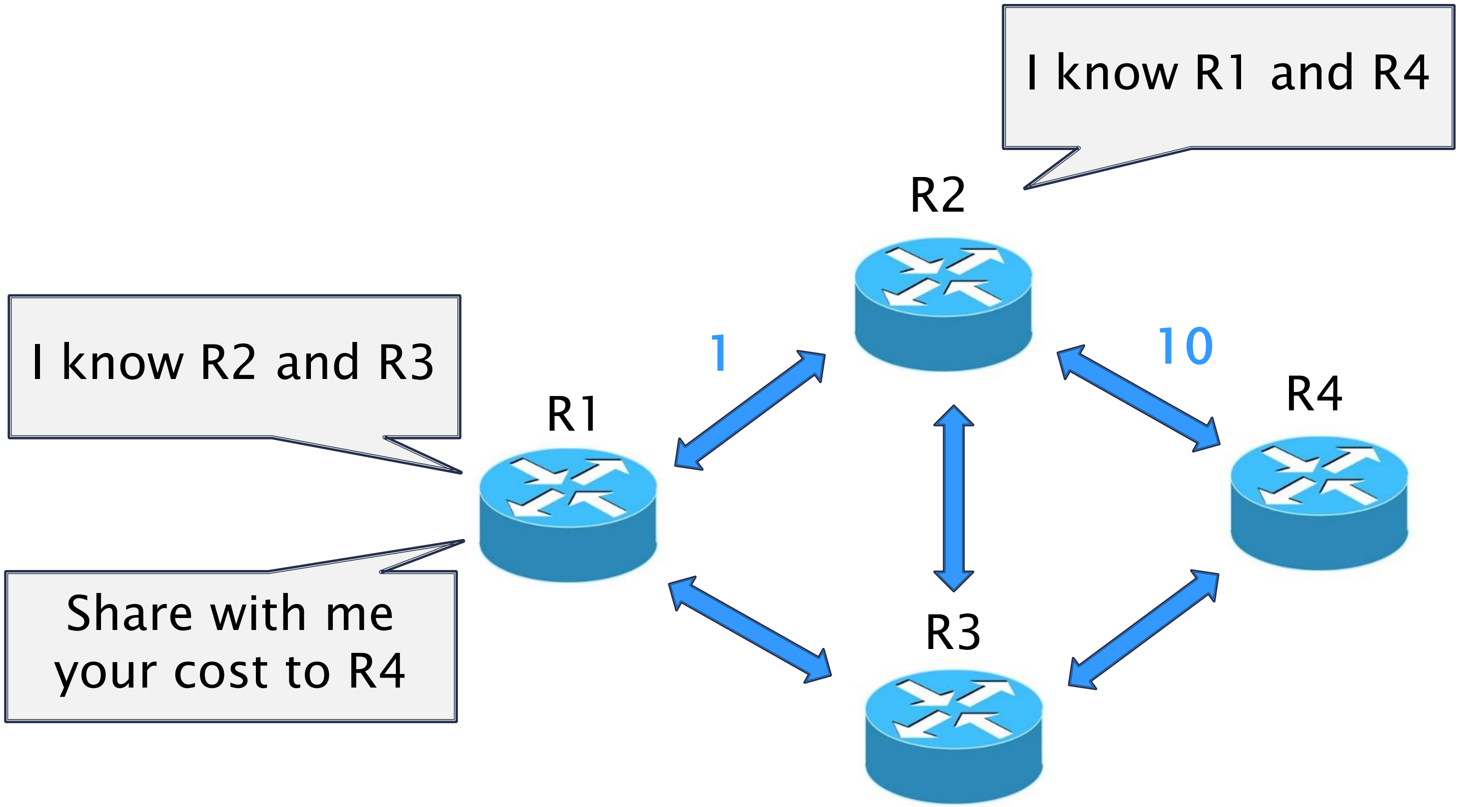
- Routers make sure they belong to same network: Network mask, password, time intervals, area ID and type

Neighbors Table

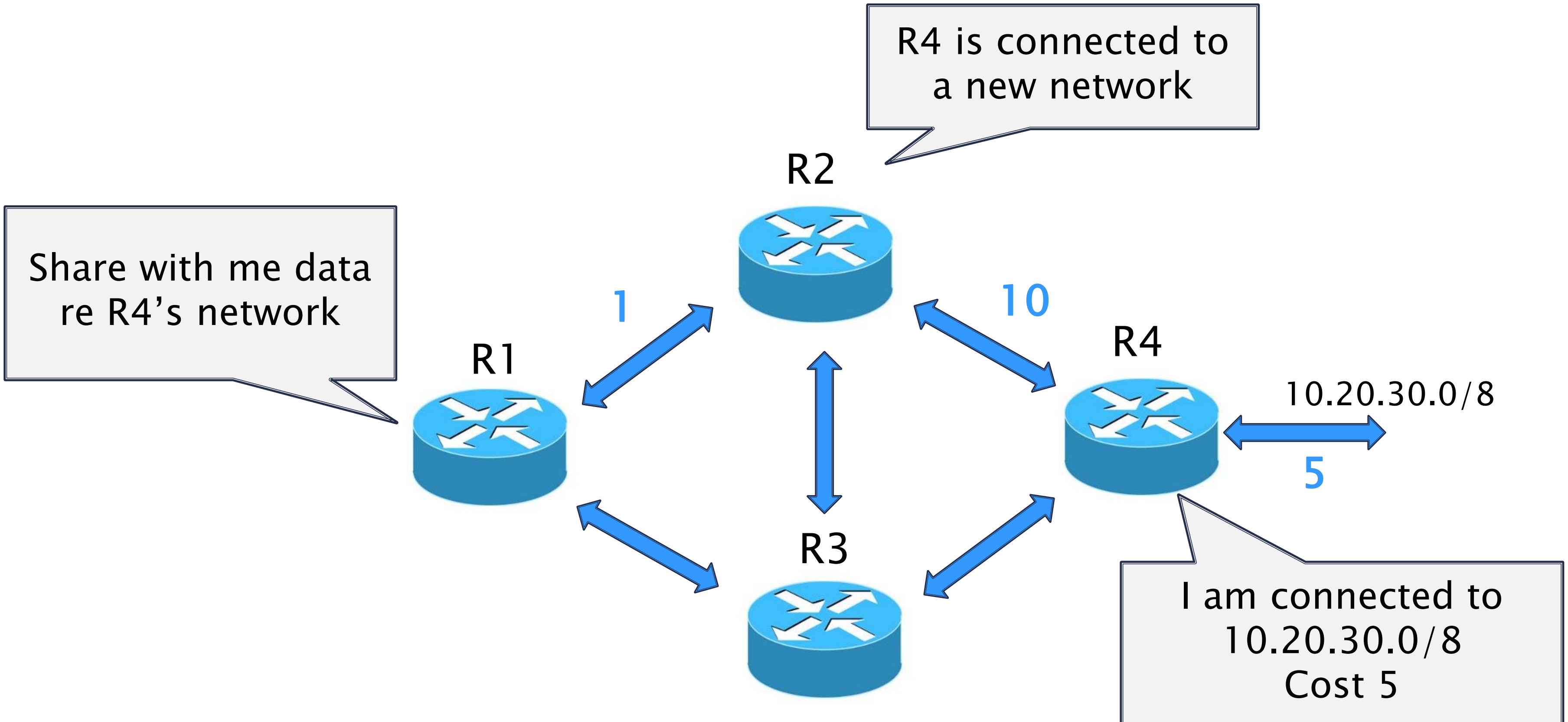


Neighbor ID	State	Dead Time	Address	Interface
2.2.2.2 (R2)	Full	00:15	10.20.0.1	GigaEther 1
3.3.3.3 (R3)	Full	00:07	10.20.10.1	GigaEther 2

Link State Advertisement

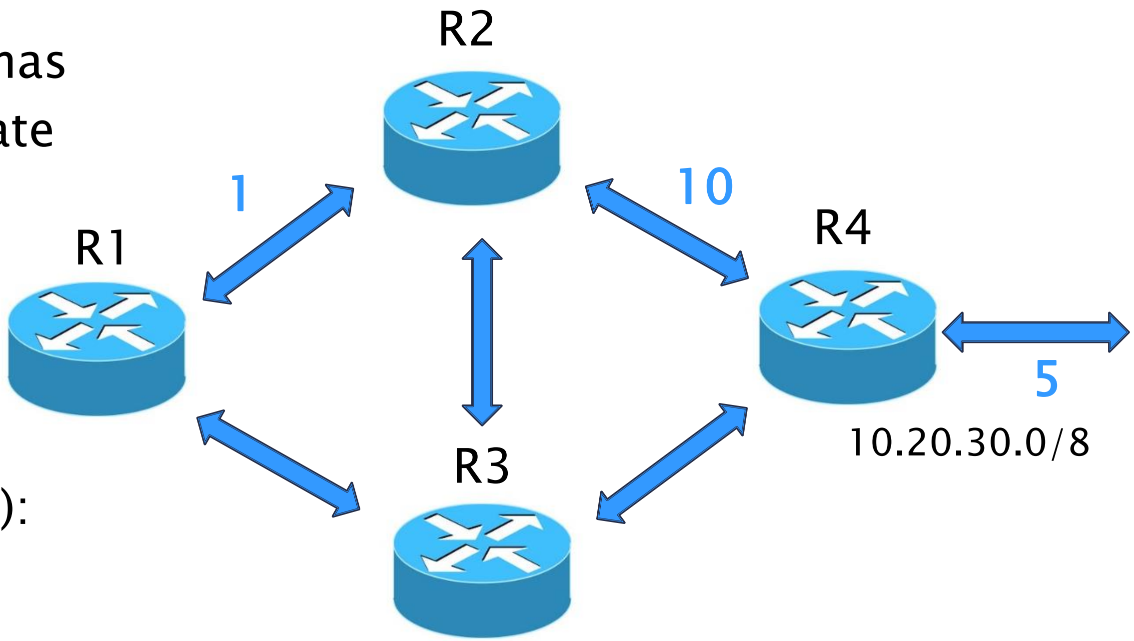


Link State Advertisement



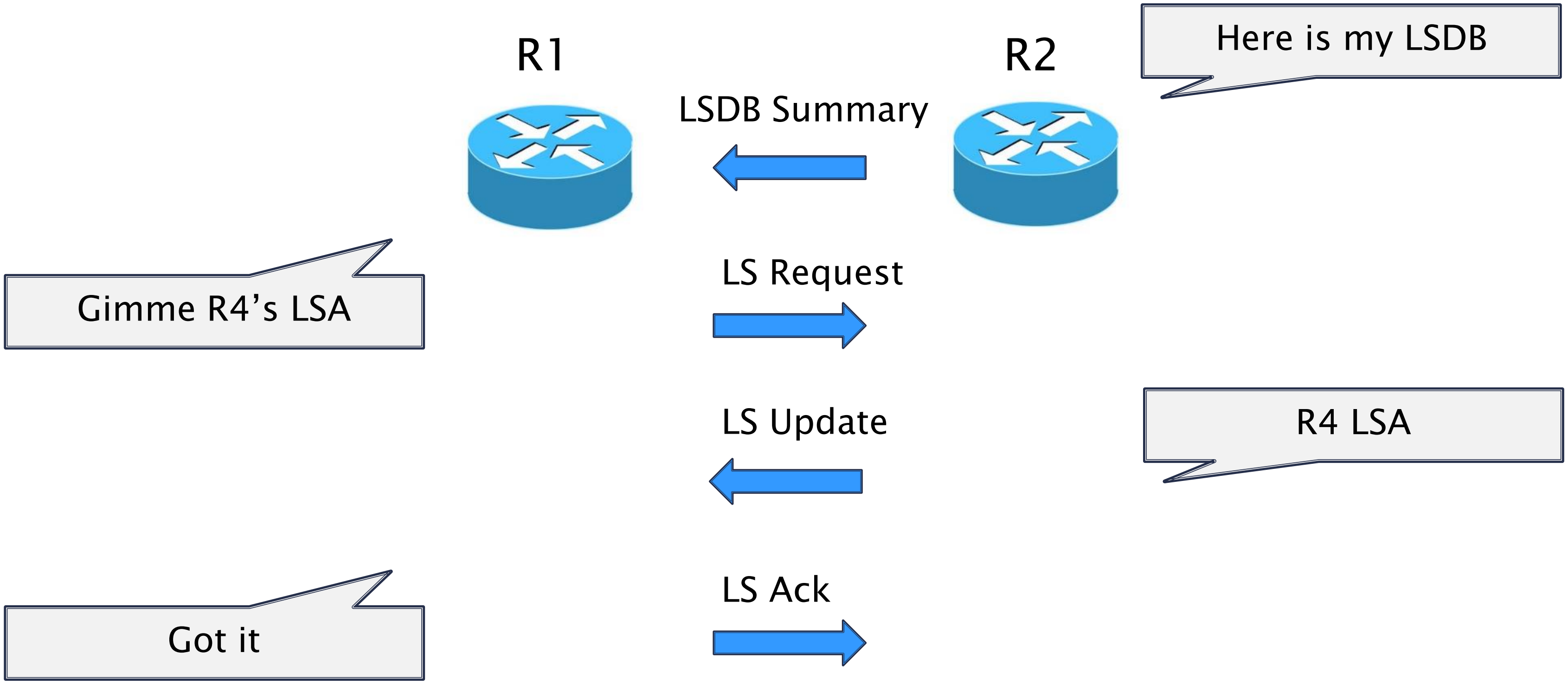
Topology Table

- All of the LSA's a router has
- Known as LSDB – Link State Data Base
- Network converged = Identical LSDB's
- The LSDB will be used to create routing table
- Example (missing 3 links):



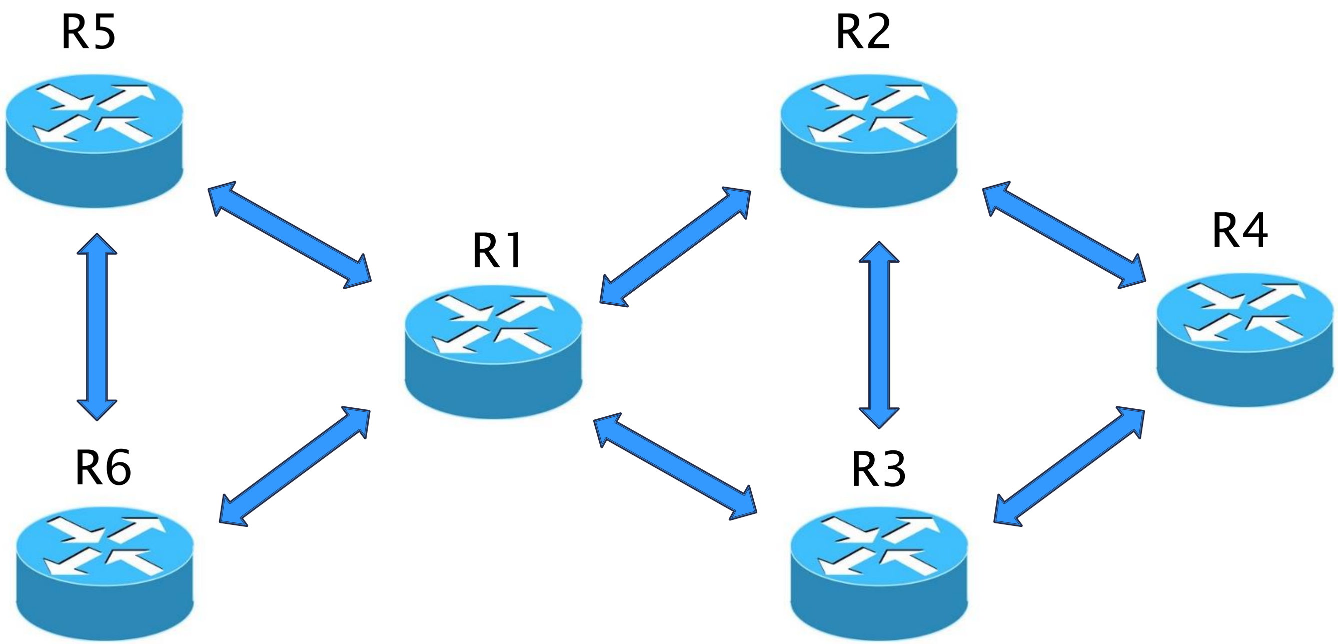
Link ID	Advertising Router	Connected to	Address	Cost
1.1.1.1	R1	R2	10.20.0.1	1
2.2.2.2	R2	R4	10.20.10.1	10
...	...	...	...	...
4.4.4.4	R4	Subnet	10.20.30.0/8	5

Link State Packet Types



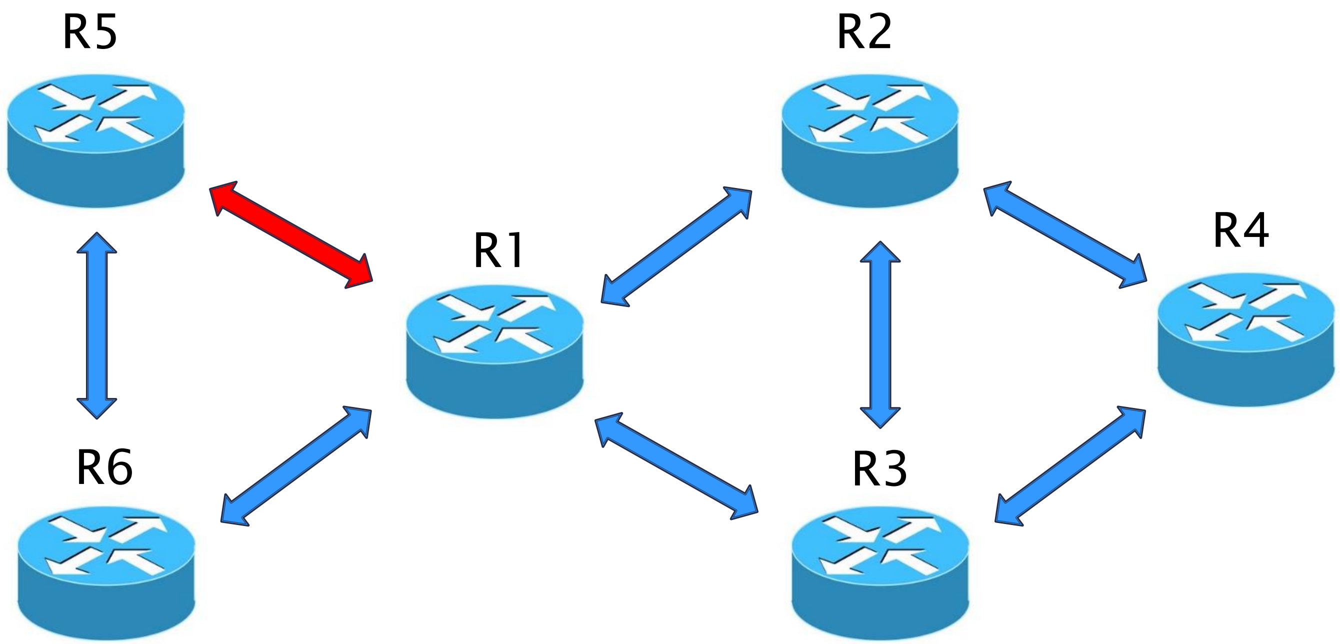
Areas

- Do *everyone* need to know about R5, R6?



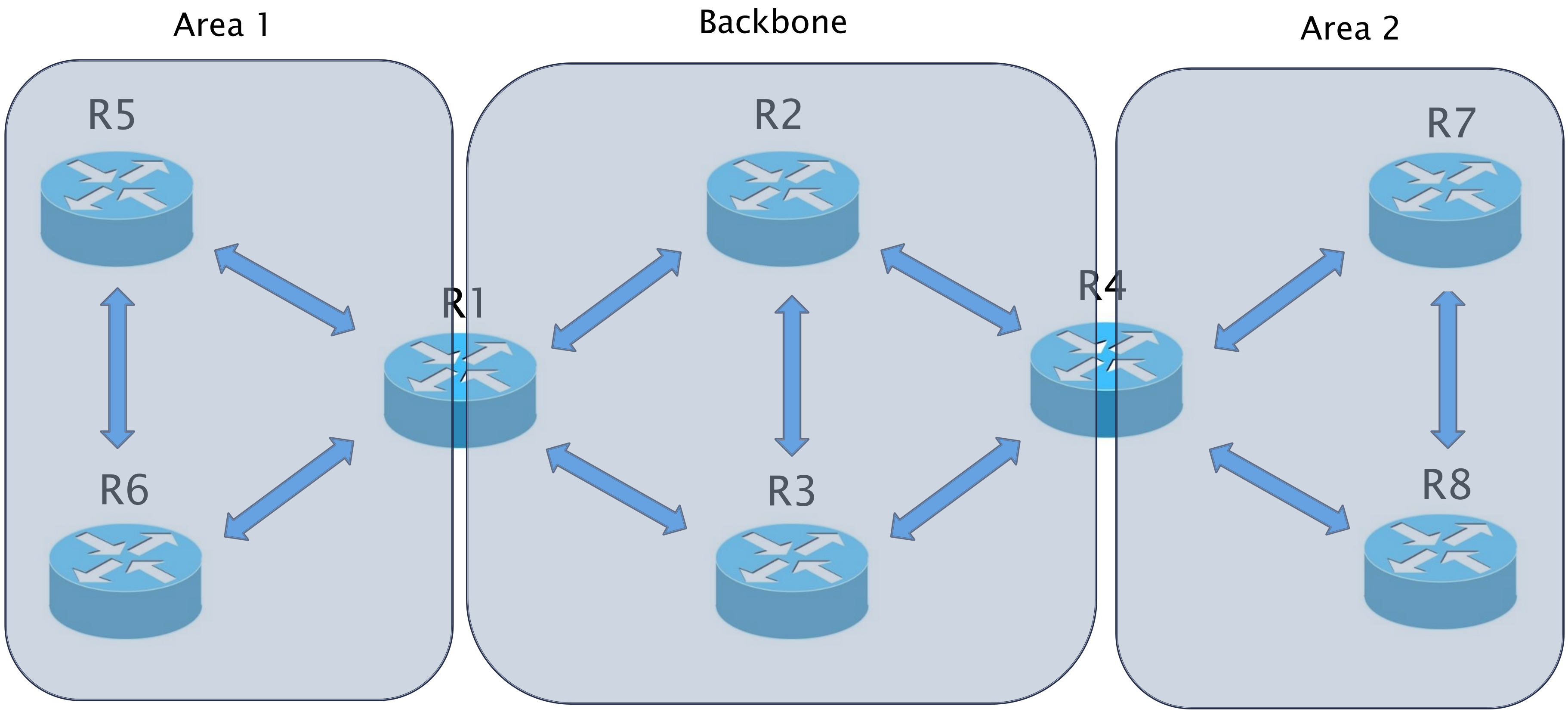
Areas

- Assume the red connection is dead. Who needs to know about it?



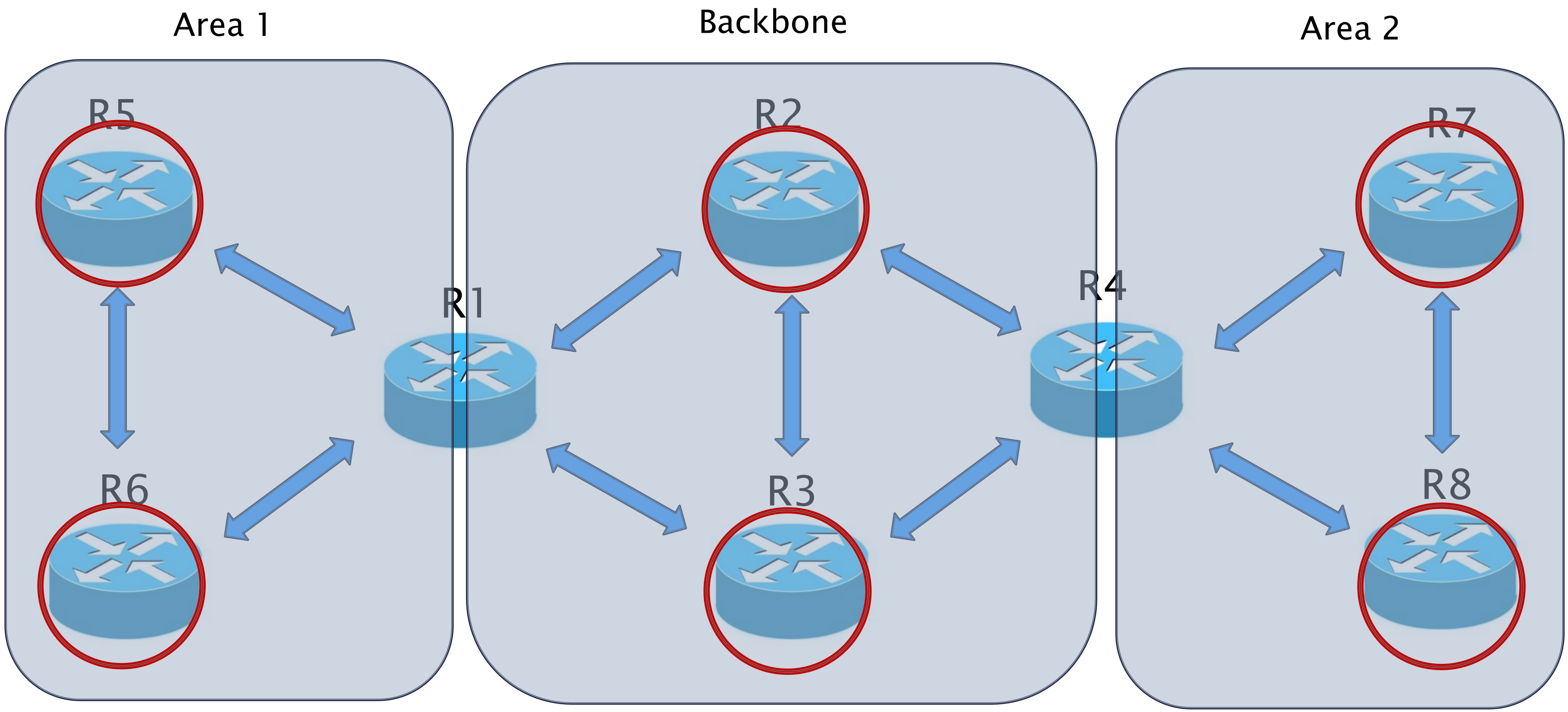
- R1does not need to propagate changes to R2, R3, R4

Areas



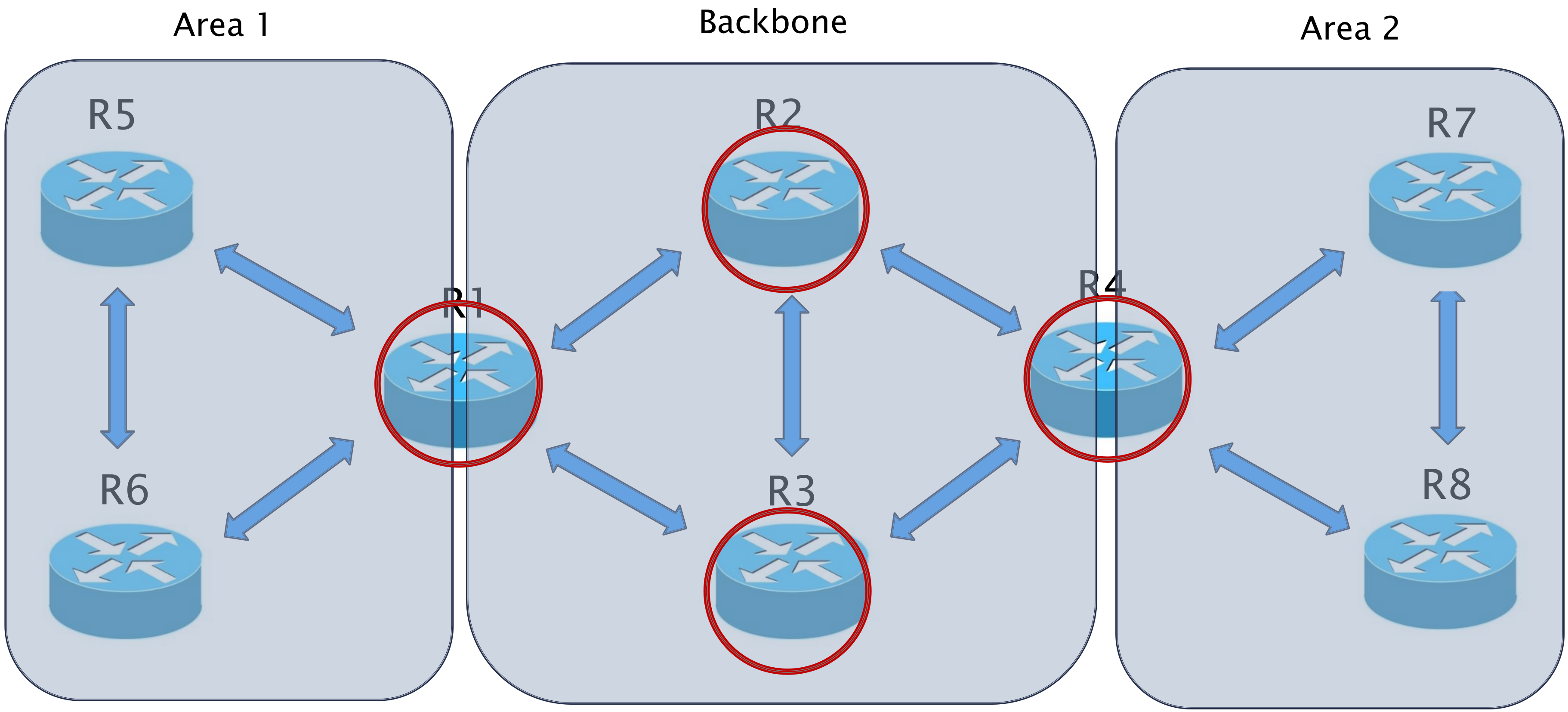
- OSPF is a star topology
- Areas can be connected only to the backbone, no loops

Types of Routers



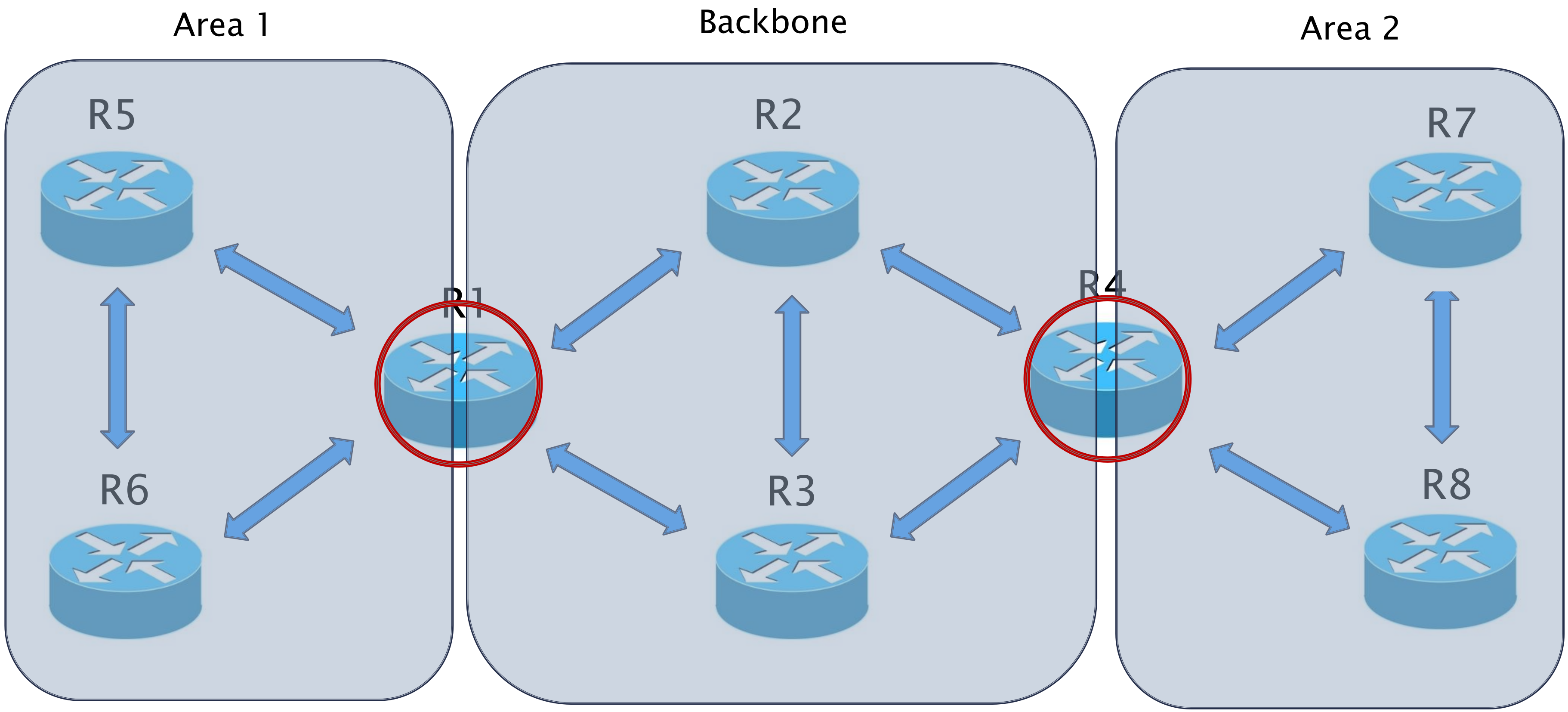
- Internal routers – all interfaces are only in one area

Types of Routers



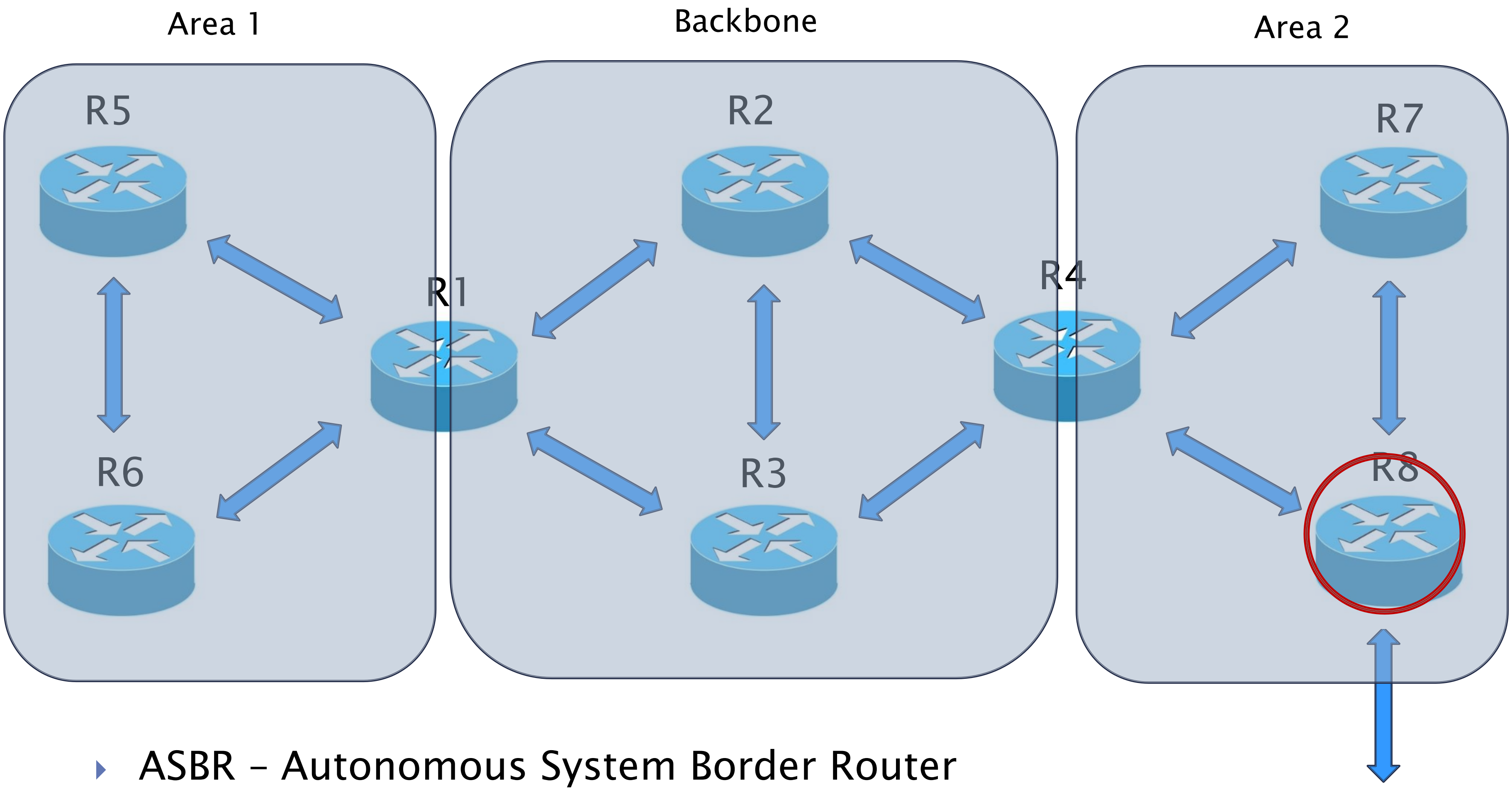
- Backbone routers – at least one interface in the backbone

Types of Routers



- Area Border Routers –interfaces in backbone and another area
  - Maintain LSDB for each area
  - Summarize LSA's and distributes

Types of Routers

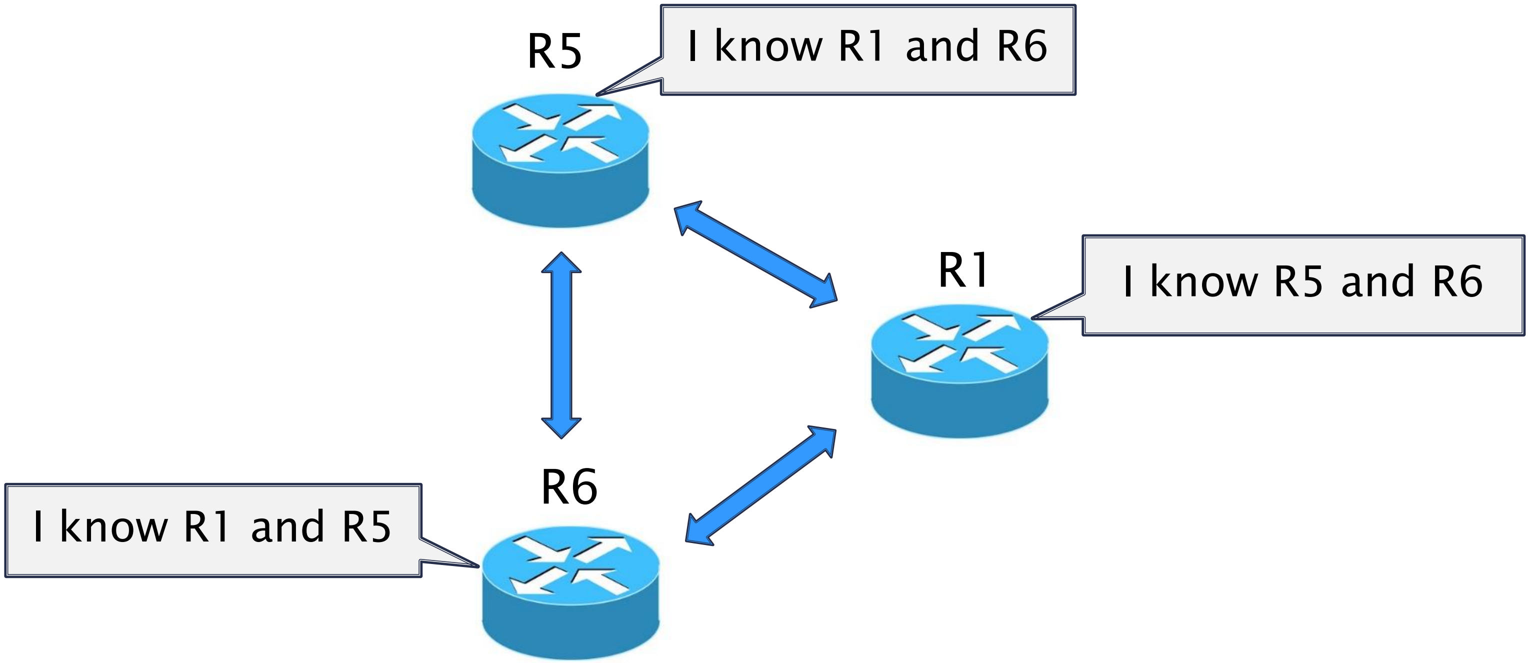


- ASBR – Autonomous System Border Router
  - Share foreign networks routes

External network (OSPF, BGP, EIGRP, RIR etc)

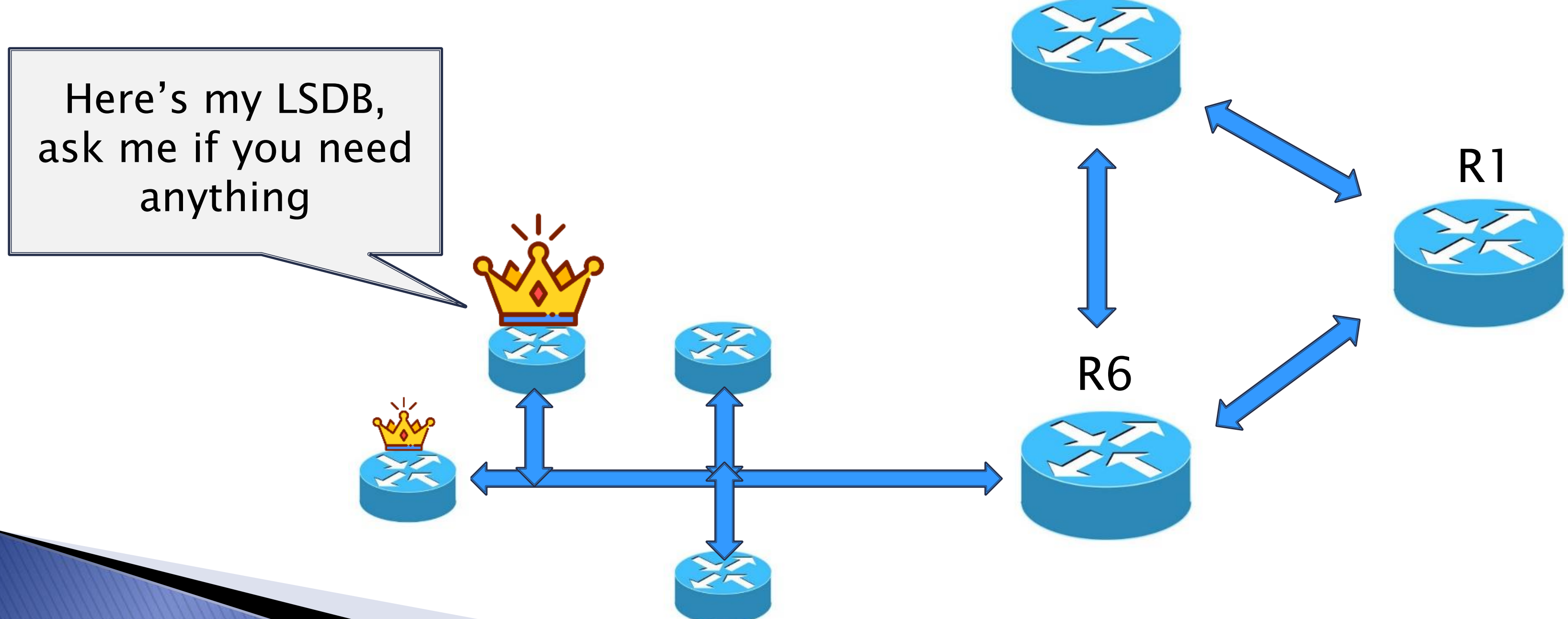
Types of LSAs

- Type 1 – Router LSA
  - Router shares links info
  - Used to create full LSDB among routers
  - Intra area – Confined to specific area



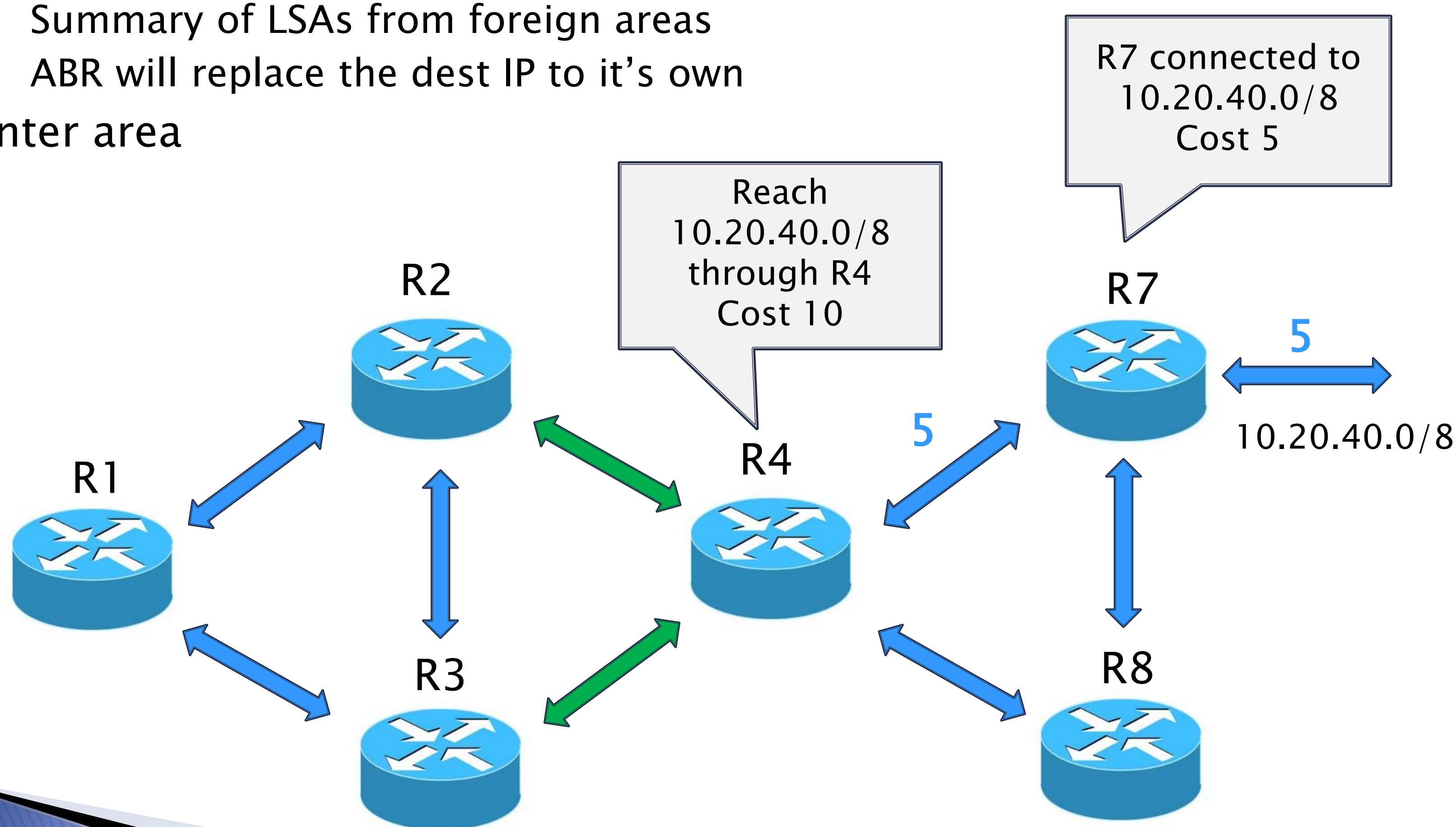
Types of LSAs

- Type 2 – Network LSA
  - If several routers share the same link (layer 2)
  - DR – Designated “king” Router
    - ADR – Alternative DR, in case DR fails
  - Only the DR will send updates
  - Intra area



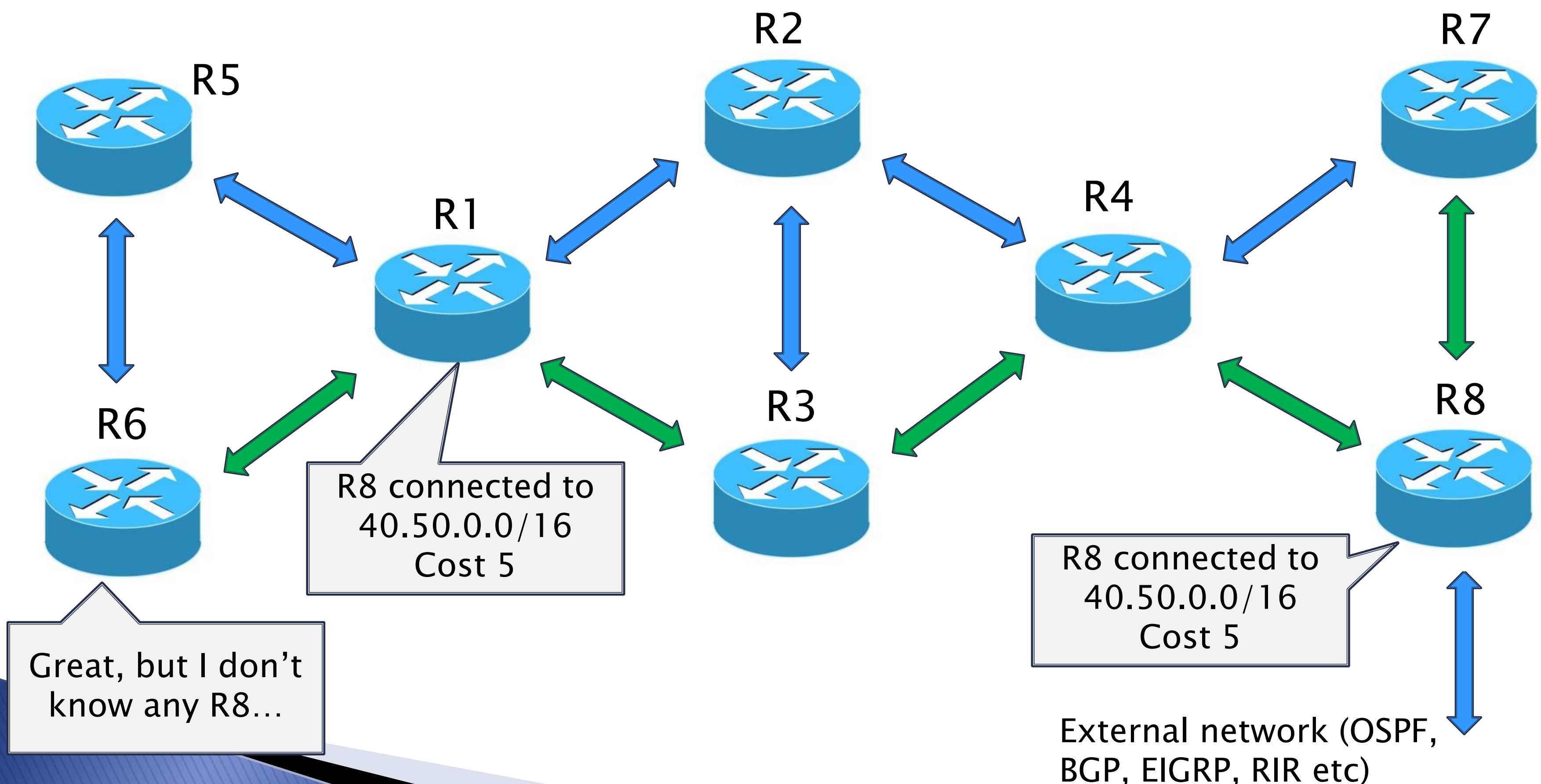
Types of LSAs

- Type 3 – Summary LSA
  - Sent by the ABR
  - Summary of LSAs from foreign areas
  - ABR will replace the dest IP to it's own
- Inter area



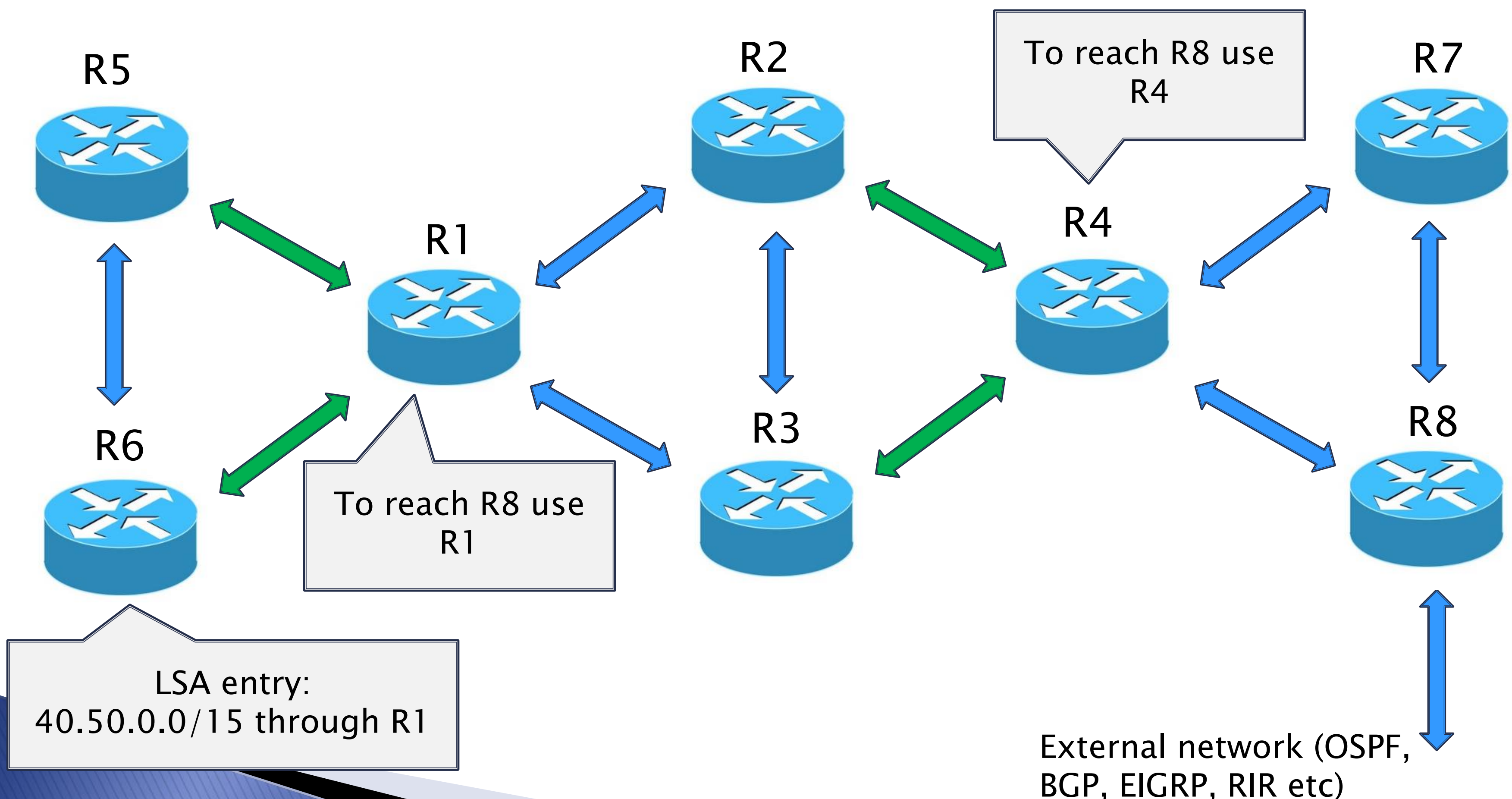
Types of LSAs

- Type 5 – ASBR External LSA
  - Propagates without change to all areas
- External area



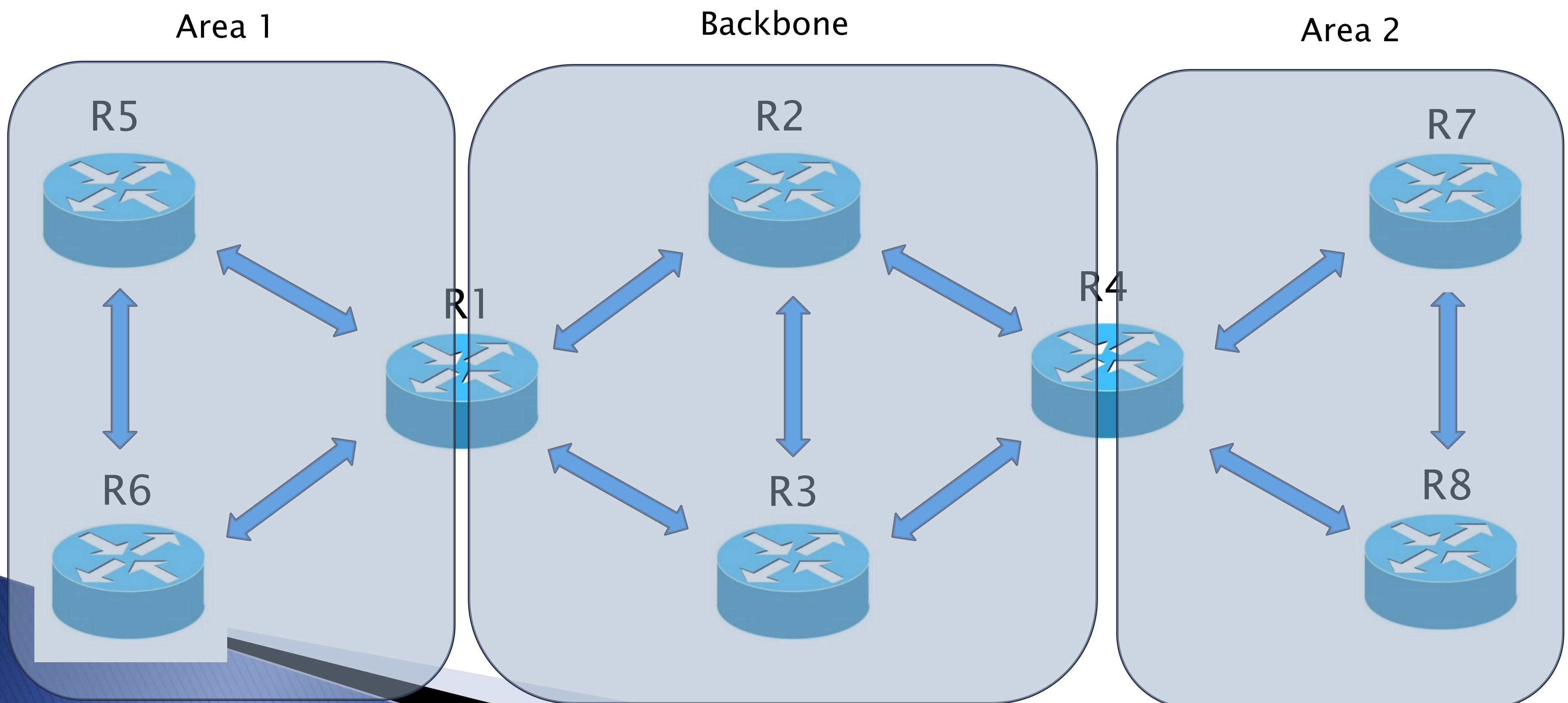
Types of LSAs

- Type 4 – ASBR Summary LSA
  - Instructions how to reach ASBR



OSPF LSA Types – Summary

- Type 1 + 2 – intra area
- Type 3 – inter area
- Type 4 + 5 – foreign area



Dijkstra's link-state routing algorithm

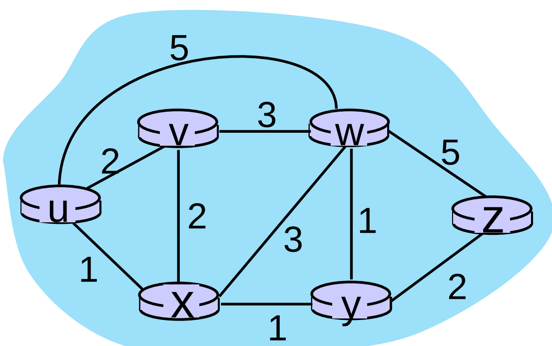
Initialization:
  $N' = \{u\}$
  for all nodes v
    if v adjacent to u
      then $D(v) = c_u$
    else $D(v) = \infty$

Loop
  find w not in N' such that $D(w)$ is a minimum
  add w to N'
  update $D(v)$ for all v adjacent to w and not in N':
    $D(v) = \min(D(v),\ D(w) + c_{w,v})$
  /* new least-path-cost to v is either old least-cost-path to v or known
  least-cost-path to w plus direct-cost from w to v */
until all nodes in N'

Source: Kurose and Ross, Computer Networking

Dijkstra's algorithm: an example

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

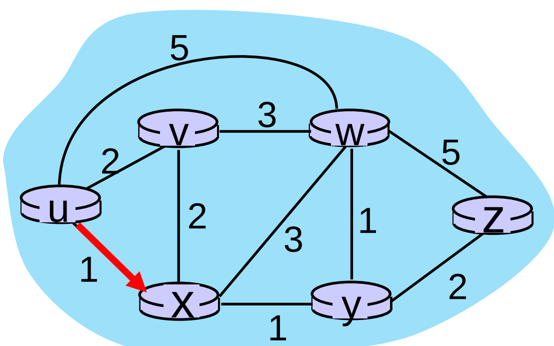


Initialization (step 0):
For all *a*: if *a* adjacent to *u* then $D(a) = c_{u,a}$

Source: Kurose and Ross, Computer Networking

Dijkstra's algorithm: an example

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



Loop

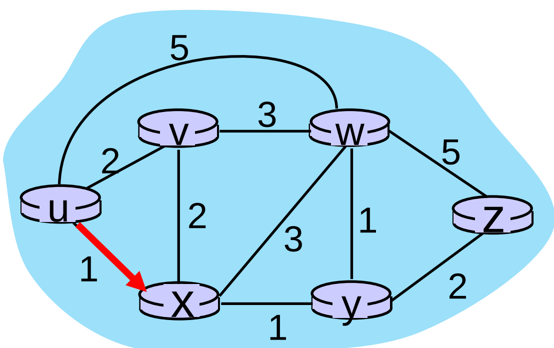
find *a* not in *N'* such that *D(a)* is a minimum

add *a* to *N'*

Source: Kurose and Ross, Computer Networking

Dijkstra's algorithm: an example

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



Loop

find *a* not in *N'* such that *D(a)* is a minimum

add *a* to *N'*

update *D(b)* for all *b* adjacent to *a* and not in *N'* :

$D(b) = \min(D(b), D(a) + c_{a,b})$

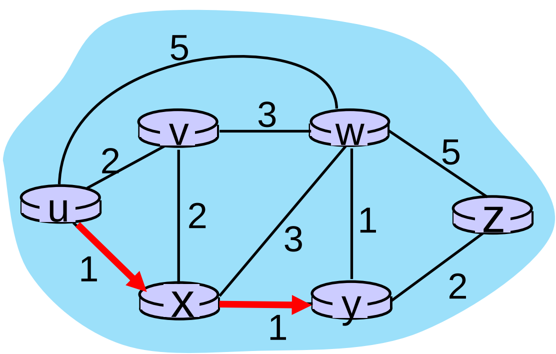
$D(v) = \min(D(v), D(x) + c_{x,v}) = \min(2, 1+2) = 2$

$D(w) = \min(D(w), D(x) + c_{x,w}) = \min(5, 1+3) = 4$ NEW!

$D(y) = \min(D(y), D(x) + c_{x,y}) = \min(inf, 1+1) = 2$ NEW!

Dijkstra’s algorithm: an example

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



Loop

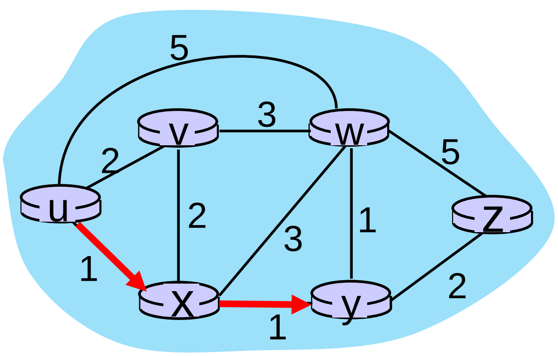
find *a* not in *N'* such that *D(a)* is a minimum

add *a* to *N'*

Source: Kurose and Ross, Computer Networking

Dijkstra's algorithm: an example

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



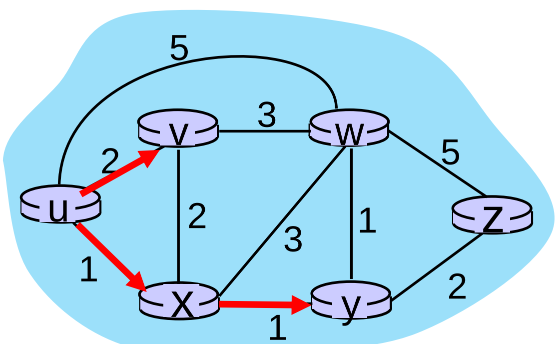
8 *Loop*
9 find *a* not in *N'* such that *D(a)* is a minimum
10 add *a* to *N'*
11 update *D(b)* for all *b* adjacent to *a* and not in *N'* :
$D(b) = \min (D(b), D(a) + c_{a,b})$

$D(w) = \min (D(w), D(y) + c_{y,w}) = \min (4, 2+1) = 3$ NEW!
$D(z) = \min (D(z), D(y) + c_{y,z}) = \min(inf,2+2) = 4$ NEW!

Source: Kurose and Ross, Computer Networking

Dijkstra's algorithm: an example

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

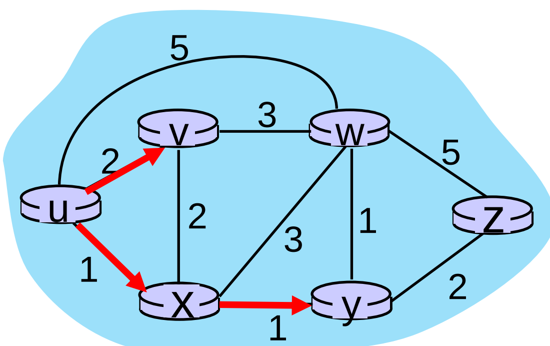


8 *Loop*
9 find *a* not in *N'* such that *D(a)* is a minimum
10 add *a* to *N'*

Source: Kurose and Ross, Computer Networking

Dijkstra's algorithm: an example

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



Loop

find *a* not in *N'* such that *D(a)* is a minimum

add *a* to *N'*

update *D(b)* for all *b* adjacent to *a* and not in *N'* :

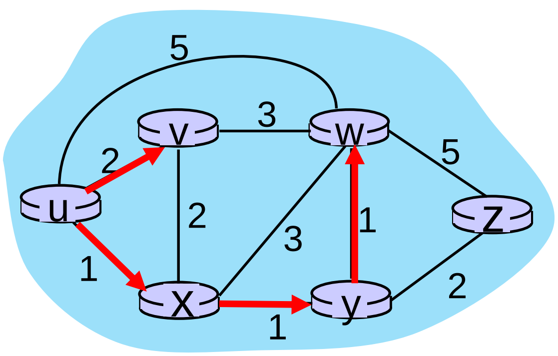
$D(b) = \min(D(b), D(a) + c_{a,b})$

$D(w) = \min(D(w), D(v) + c_{v,w}) = \min(3, 2+3) = 3$

Source: Kurose and Ross, Computer Networking

Dijkstra's algorithm: an example

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



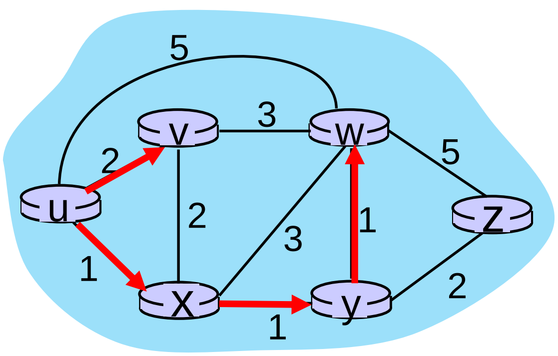
8 *Loop*

9 find *a* not in *N'* such that *D(a)* is a minimum

10 add *a* to *N'*

Dijkstra's algorithm: an example

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



8 *Loop*

9 find a not in N' such that $D(a)$ is a minimum

10 add a to N'

11 update $D(b)$ for all b adjacent to a and not in N' :

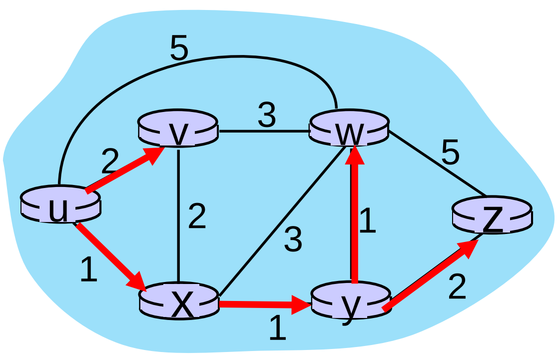
$$D(b) = \min (D(b), D(a) + c_{a,b})$$

$$D(z) = \min (D(z), D(w) + c_{w,z}) = \min (4, 3+5) = 4$$

Source: Kurose and Ross, Computer Networking

Dijkstra's algorithm: an example

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



8 *Loop*

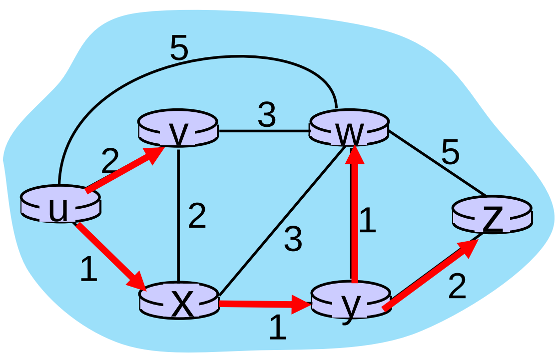
9 find *a* not in *N'* such that *D(a)* is a minimum

10 add *a* to *N'*

Source: Kurose and Ross, Computer Networking

Dijkstra's algorithm: an example

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



8 *Loop*

9 find *a* not in *N'* such that *D(a)* is a minimum

10 add *a* to *N'*

11 update *D(b)* for all *b* adjacent to *a* and not in *N'* :

$D(b) = \min (D(b), D(a) + c_{a,b})$

Source: Kurose and Ross, Computer Networking

Summary

- Metric
- Link State Advertisements
- Areas
- Minimal cost algorithm – Dijkstra