COMP2221 Networks

David Head

University of Leeds

Lecture 18

Previous lectures

In the last lecture we saw how routers redirect packets using as **forwarding table**:

- Traditionally just use the destination IP address, but increasingly can use other information and perform other actions, *i.e.* the **match-action** pattern.
- Known as **generalised forwarding**, and implemented in SDNs = Software Defined Networks.
- Also looked at ICMP, the Internet Control Message Protocol.
- IPv6 and how it co-exists with IPv4 using **tunnelling**.

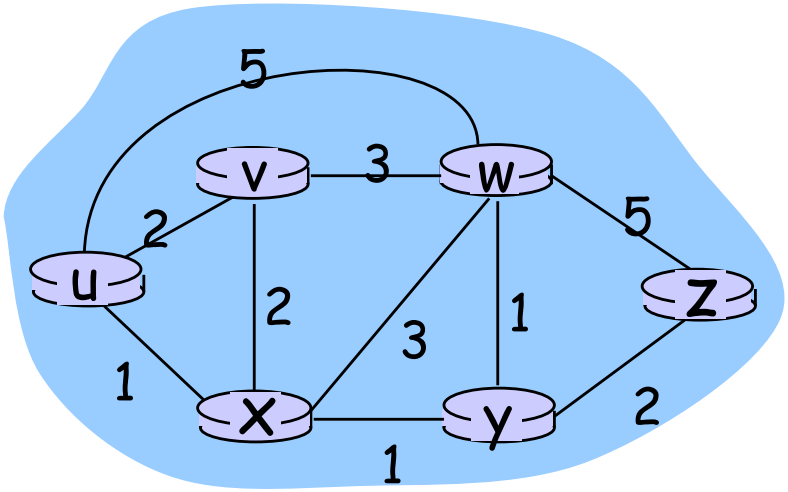
Today's lecture

Todays lecture is the second of two on the Network layer and we will look at how the outgoing path for each packet is actually determined, *i.e.* **routing algorithms**:

- **Dijkstra's algorithm**, a link state algorithm.
- **RIP**, a distance vector algorithm.
- **OSPF** and the hierarchical version, that work within subnetworks.
- **BGP**, the *de facto* standard for routing between subnetworks.

We will only consider destination IP addresses, and assume the goal is to be transported 'efficiently.'

Routing: Graph abstraction[1]



Graph $G = (N, E)$, where:

- N (nodes) is a set of **routers**: $\{u, v, w, x, y, z\}$
- E (edges) is a set of **links**:
  $\{(u, v), (u, x), (u, w), (v, x), (v, w), (x, w), (x, y), (w, y), (w, z), (y, z)\}$

[1] Graph abstraction also useful in peer-to-peer (P2P) communication, with N the set of peers and E the set of TCP connections.

Link costs

Let $c(x, y)$ be the **cost** of link (x, y).

- *e.g.* $c(w, z) = 5$ in this example.
- Could be inversely related to bandwidth, or could ignore (*i.e.* set all costs to the same value, *e.g.* 1).

Cost of **path** $(x_1, x_2, x_3, \ldots x_p)$ is

$$c(x_1, x_2) + c(x_2, x_3) + \ldots + c(x_{p-1}, x_p)$$

Question: What is the least cost path between u and z?

Routing algorithm: Algorithm that finds this least-cost path

Routing algorithm classification

Global or **decentralised** information?

- If **global**, all routers have complete knowledge of topology and link costs. Leads to **link state** algorithms.
- If **decentralised**, each router knows only physically-connected routers and link costs to them.
  - **Iterative** process of computation and information exchange with neighbours (**distance vector** algorithms).

Can also be **static** or **dynamic**:

- **Static** suitable if routes change slowly.
- **Dynamic** if routes change more quickly, including link cost changes.

Dijkstra's link-state routing algorithm

Assume topology and costs of entire (sub-)network known to all hosts, achieved by *i.e.* link state broadcast.

- Computes least cost paths from **source** node to all others.
- Gives **forwarding table** for that node.
- **Iterative**: After k iterations, knows least cost to k destinations.

Notation:

- $c(x, y)$ is link cost for $x \to y$, or ∞ if no direct link.
- $D(v)$ is current path cost from source to v.
- $p(v)$ is the predecessor node along path to v.
- N' is set of nodes whose least cost path is known.

Dijkstra's algorithm

Initialisation:

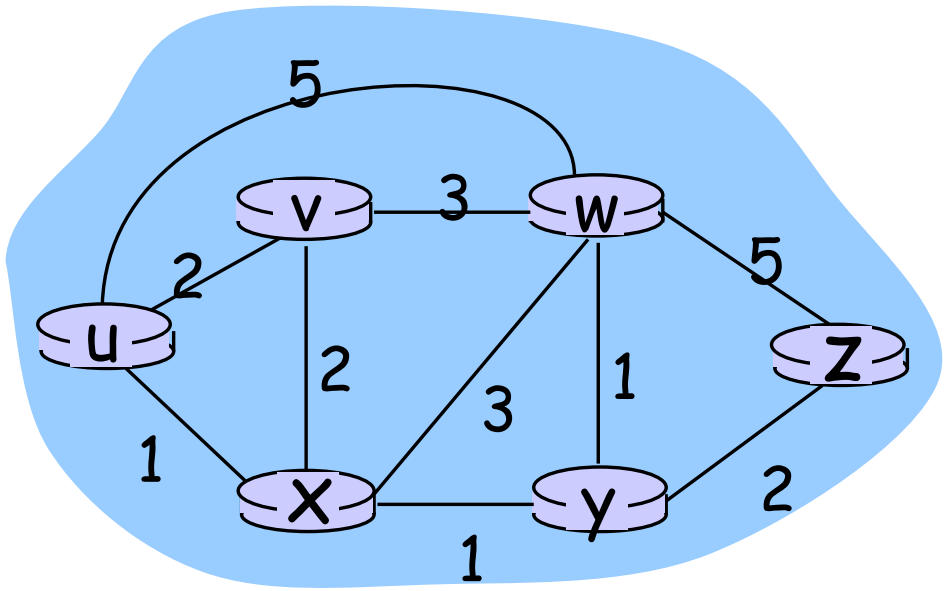
```
N' = {u}
for all nodes v:
    if v is a neighbour of u
       then D(v) = c(u,v)
    else D(v) = ∞
```

Main iteration loop:

```
find w not in N' such that D(w) is a minimum
add w to N'
update D(v) for each neighbour v of w and not in N':
       D(v) = min( D(v), D(w)+c(w,v) )
       /* New cost to v is either old cost to v, or
       known least path cost to w plus cost from w to v.  */
until N' = N
```

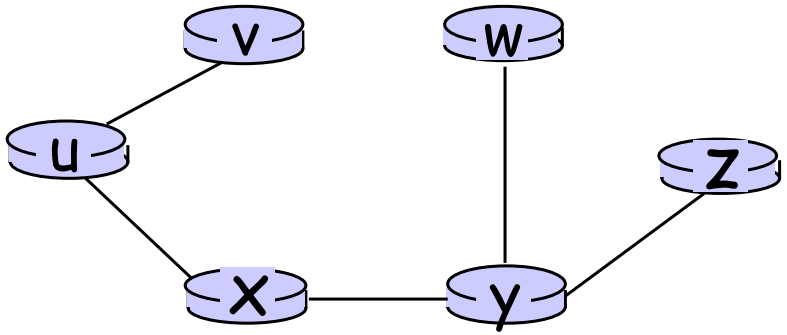
Dijkstra's algorithm: Example

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



Dijkstra's algorithm

Resulting shortest path **tree** from u to all other nodes is:

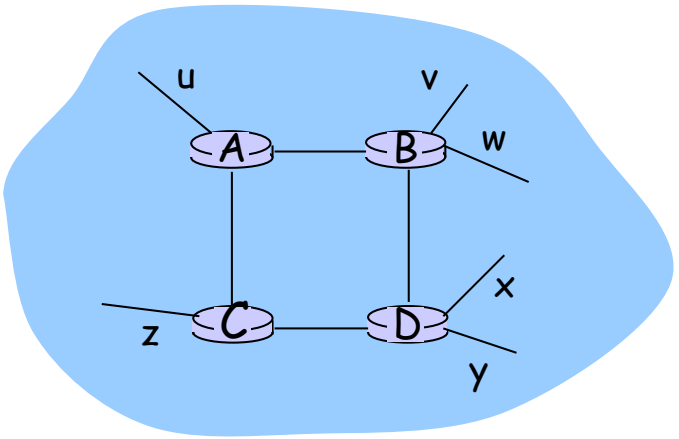


The resulting **forwarding table** for node u:

Destination	Link
v	(u, v)
x	(u, x)
y	(u, x)
w	(u, x)
z	(u, x)

RIP: Routing Information Protocol

- **Distance vector** algorithm, an iterative algorithm that only knows about local links and connected nodes.
- Included in BSD-UNIX[1], the precursor to FreeBSD, OpenBSD *etc.*, in 1982.
- Distance metric is the **number of hops** (maximum of 15).



From router A to subsets:

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

[1] BSD = Berkeley Software Distribution.

RIP Advertisements

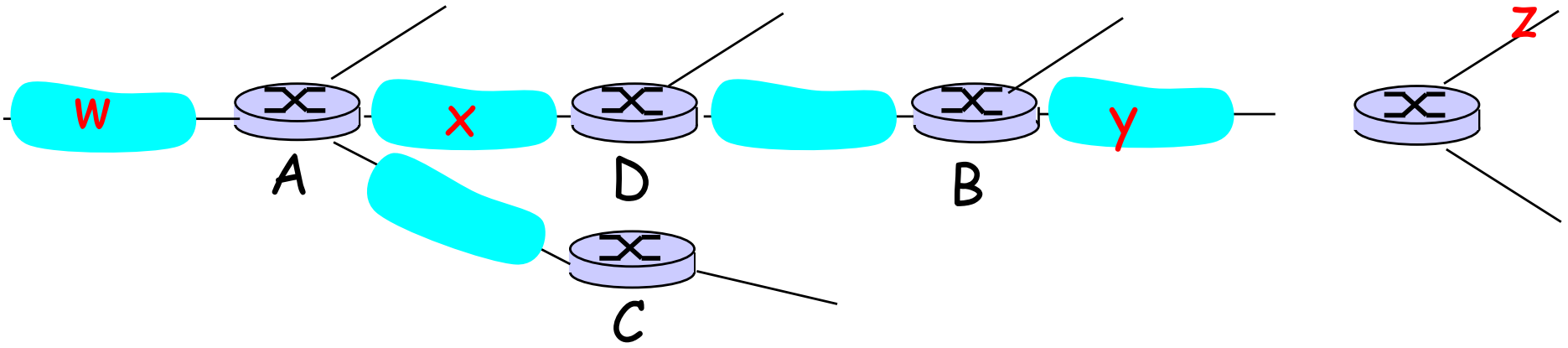
Distributed algorithm that communicates **asynchronously** with directly connected nodes.

Distance vectors are exchanged among neighbours every 30 seconds *via* a **response message**, also known as an **advertisement**.

Each **advertisement** is list of up to 25 destination networks with the subnetwork.

- The **costs**, defined as the number of **hops**.
- Smaller messages that Dijkstra's algorithm.
- Tends to have slower convergence, with no guarantees.

RIP: Example from router D



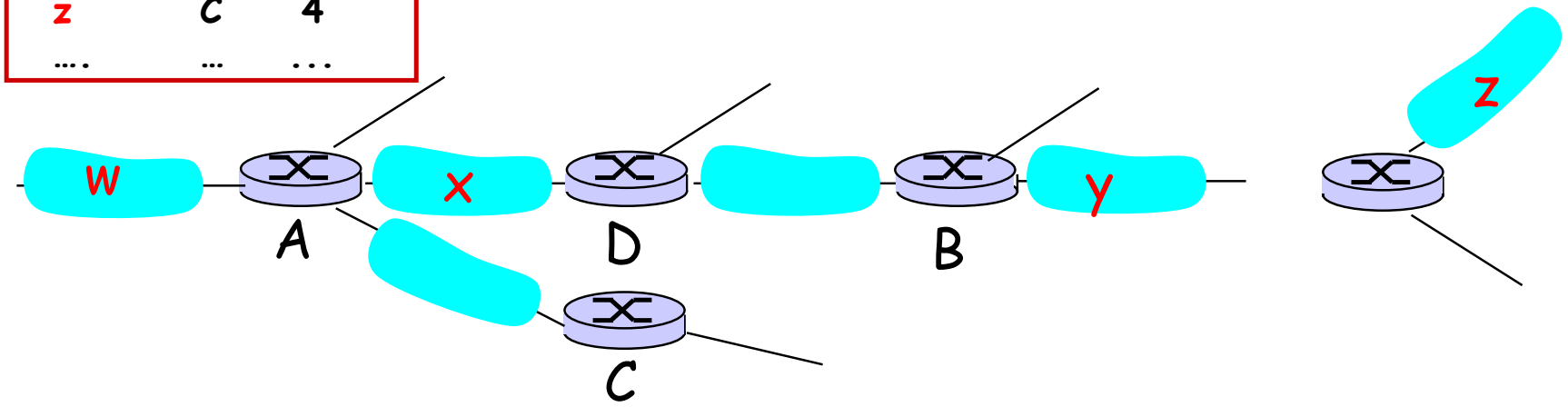
Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	B	7
x	--	1
....	....	....

Routing table in D

RIP: Example

Dest	Next	hops
w	-	1
x	-	1
z	C	4
….	…	…

Advertisement from A to D



Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	~~B~~ A	~~7~~ 5
x	--	1
….	….	….

Routing table in D

RIP: Link failure and recovery

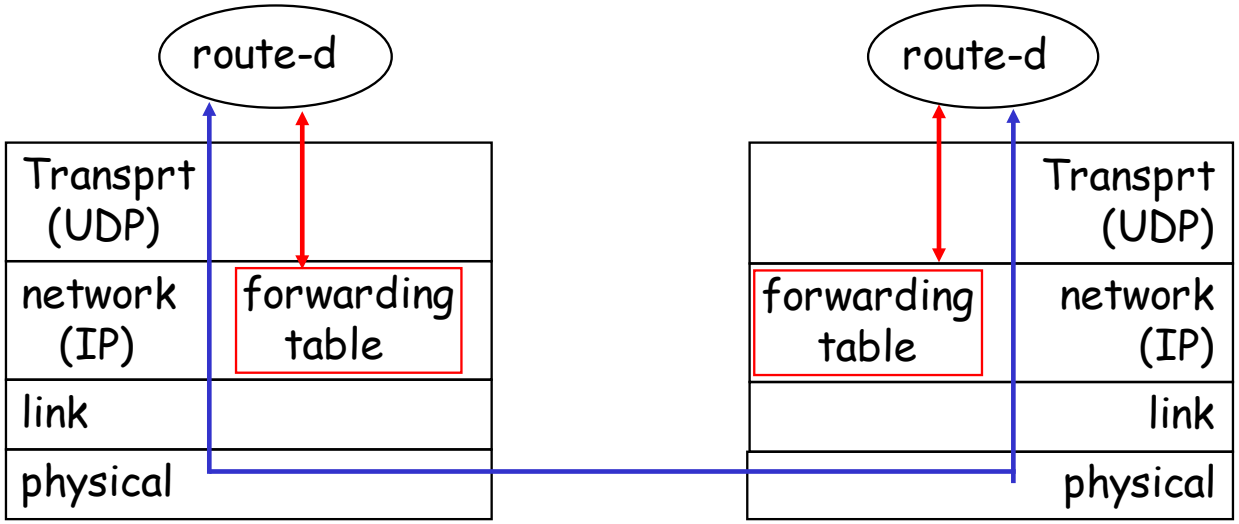
If no advertisement is received after 180 seconds, that neighbour or link is declared **dead**.

- Routes *via* the dead neighbour are **invalidated**.
- New advertisements send to neighbours.
- Neighbours in turn send out new advertisements, if their tables changed.
- Link failure information quickly propagates into the network.

RIP Table Processing

- RIP routing tables are managed by Application layer processes called **route-d** ('daemon').
- Advertisements are sent in UDP packets, periodically repeated.

Note this technically **breaks** the layered network architecture.



OSPF: Open Shortest Path First

OSPF = Open Shortest Path First.

- 'Open' because it is publicly available.

Uses a **link state** algorithm but **only on a subnetwork**.

- Topology map known at each node.
- Route computation using Dijkstra's algorithm.

Only intended to act within an **AS = Autonomous System**:

- Each ISP will have one or more AS's.
- Gives subnetwork administrator greater control.
- OSPF advertisements carry one entry per neighbour.
- Carried directly over IP; does not use TCP or UDP.

Hierarchical OSPF

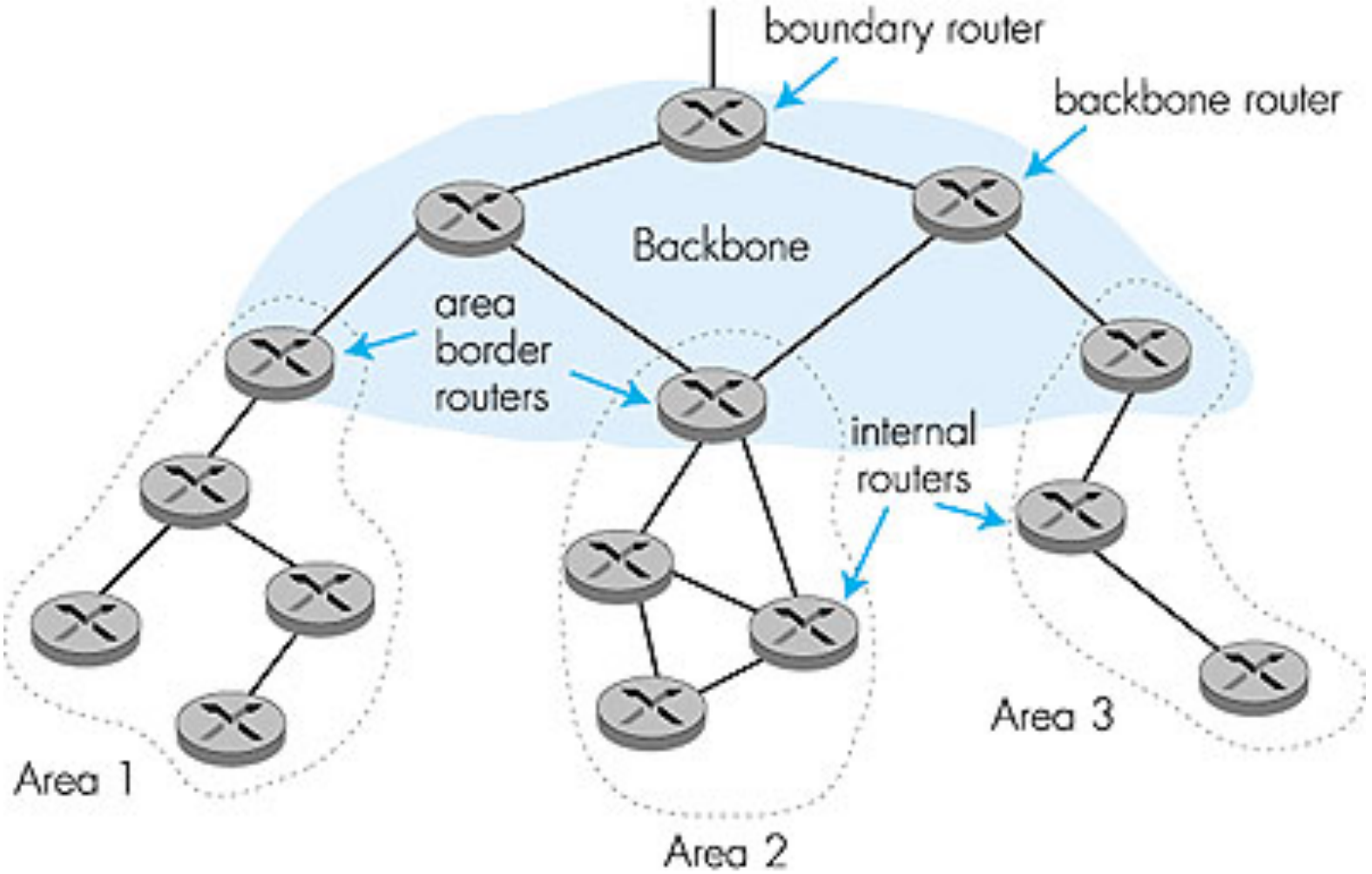
Even a single ISP is normally too large for the whole network to be considered, so a **two-level hierarchy** is typically used with **local areas** and a **backbone**:

- Link-state advertisements only in the local area.
- Each node has detailed area topology, but only knows the shortest path to networks in other areas.

On the diagram on the next slide:

- **Area border routers** summarise distances to networks in own area, advertise to other area border routers.
- **Backbone routers** run OSPF routing limited to the backbone.
- **Boundary routers** connect to other AS's.

Hierachical OSPF



BGP: Border Gateway Protocol

The *de facto* standard for inter-AS routing on the internet is the **BGP** = Border Gateway Protocol.

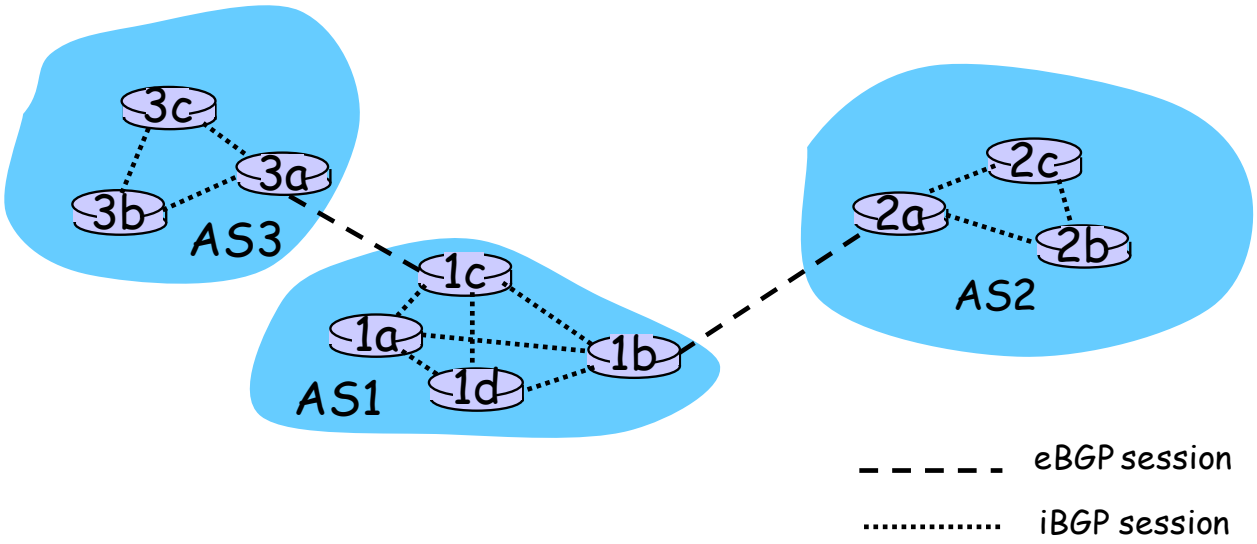
BGP provides each AS a means to:

1. Obtain the reachability information of a subnetwork from neighbouring AS's.
2. Propagate reachability information to all routers internal to an AS.
3. Determine efficient routes to subnetworks based on reachability information **and policy**.
4. Allows subnetwork to advertise its existence to the rest of the Internet: 'I am here.'

BGP Basics

Pairs of routers (BGP peers) exchange routing information over semi-permanent TCP connections: **BGP sessions**.

- Need not correspond to physical links.
- **External session:** Spans two AS's.
- **Internal session:** Both BGP routers in same AS.



BGP Example

BGP sessions advertise CIDR **prefixes**, that is, the 'destinations' are of the form *e.g.* `129.11.128.0/20`.

In the example on the previous slide:

- When AS2 advertises a prefix to AS1, AS2 is **promising** that it will forward any packets destined for that prefix towards the prefix.
- AS2 can also **aggregate** prefixes within its own message.

BGP Route Selection

A BGP router may learn about more than one route to a prefix, in which case it needs to select one.

To help in this, there are some **elimination rules**:

1. Each potential route is assigned a **local preference** by the system administrator as per any policies.
2. For all routes with the highest local preference, the ones with the shortest AS-PATH (*i.e.* the number of AS's the packet would pass through).
3. If still tied, the shortest cost (*i.e.* number of physical links) to the first router in the next AS is used.
4. If still tied, other criteria could be considered.

Overview and next lecture

Today we have looked at a range of **routing algorithms** that are often used (in some form) in the Internet.

- **OSPF**, which employs **Dijkstra's algorithm**.
- **BGP**, used to route between subnetworks.

This ends our discussion of the Network layer.

Next time we will look at the final two layers, the Link layer and (very briefly) the Physical layer.