COSC 417: Topics in Networking

TOPIC 5: IPG ROUTING II – LINK-STATE ROUTING

SCHEDULE

1. *Recap*
2. *Link-State Protocols*
   1. *How L-S protocols work*
   2. *The Dijkstra SPF*
3. *OSPF*

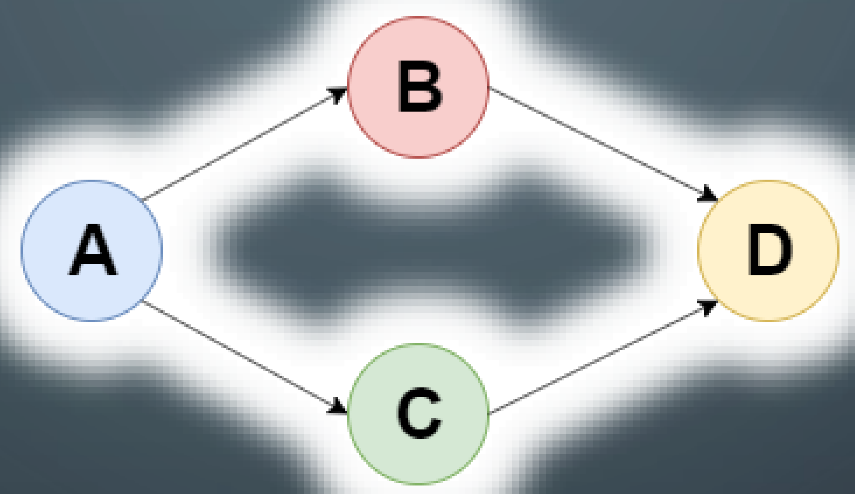
RECAP – IGP ROUTING

- Last lecture we began discussing *Interior Gateway Protocols*, which are used to exchange routing information within the autonomous system
- There are many IGP protocols, broadly classed into *distance-vector* and *link-state* protocols

RECAP – DISTANCE-VECTOR PROTOCOLS

- We took a look at *distance-vector* routing protocols last class
- Based on the Bellman-Ford algorithm
- Nodes don't need to store full maps of network topology, instead relying on neighbours sharing information
- Suffer from count-to-infinity problems, slow convergence

RECAP – BELLMAN-FORD



- $dist_A(D) = minimum\{\ edge_A(B) + dist_B(D),$

  $edge_A(C) + dist_C(D)\ \}$

RECAP – THE RIP PROTOCOLS

- The *Routing Information Protocol* (RIP) was the first major IGP protocol to come into use (1980s)
- Only metric used to gauge which route is 'best' is the number of 'hops' – the less hops in the route, the better
- Max hop count of 15 to prevent loops
- Later variants with more features/standards

ALRIGHT, WHAT'S NEXT?

- As I mentioned in the last lecture, RIP and other distance-vector protocols have mostly fallen into obsolescence
- Nowadays, most IGP routing is done using what are known as *link-state* routing protocols, predominantly the *OSPF* and *IS-IS* protocols

LINK-STATE ROUTING

WHAT ARE LINK-STATE PROTOCOLS?

- *Link-state* routing protocols differ from distance-vector protocols in how they share their routing data, and how they build their routing tables
- These protocols avoid some of the problems that appear in distance-vector protocols, namely the count-to-infinity problem and convergence speed

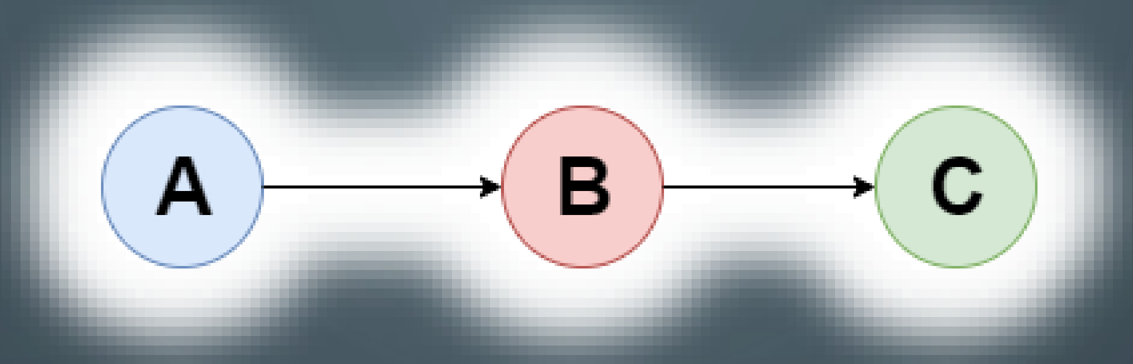
THE TWO MAJOR STEPS OF LINK-STATE PROTOCOL

- Link-state protocols can generally be broken down into two discrete steps that are performed:
  - Step 1: share neighbour information with network and build a network map
  - Step 2: use the complete network map to generate a routing table

BROADCAST SHARING

- The first major difference is that while distance-vector algorithms rely on neighbours eventually learning and sharing routes across the network, link-state algorithms broadcast that information to the entire network
- This procedure is known as *flooding* – packets are passed along on all interfaces (except the one they arrived on)

A COMPARISON



- In distance-vector, A would rely on B eventually sharing the route to C
- In link-state, A, B, and C would be broadcasting their routes and route costs to each other – A could learn of the B-C route from C, for example

MEETING YOUR NEIGHBOURS

- The first step in a link-state protocol is actually identifying your neighbouring routers
- Each router is going to identify which ports/interfaces it has open, and regularly check for directly connected neighbours on those interfaces

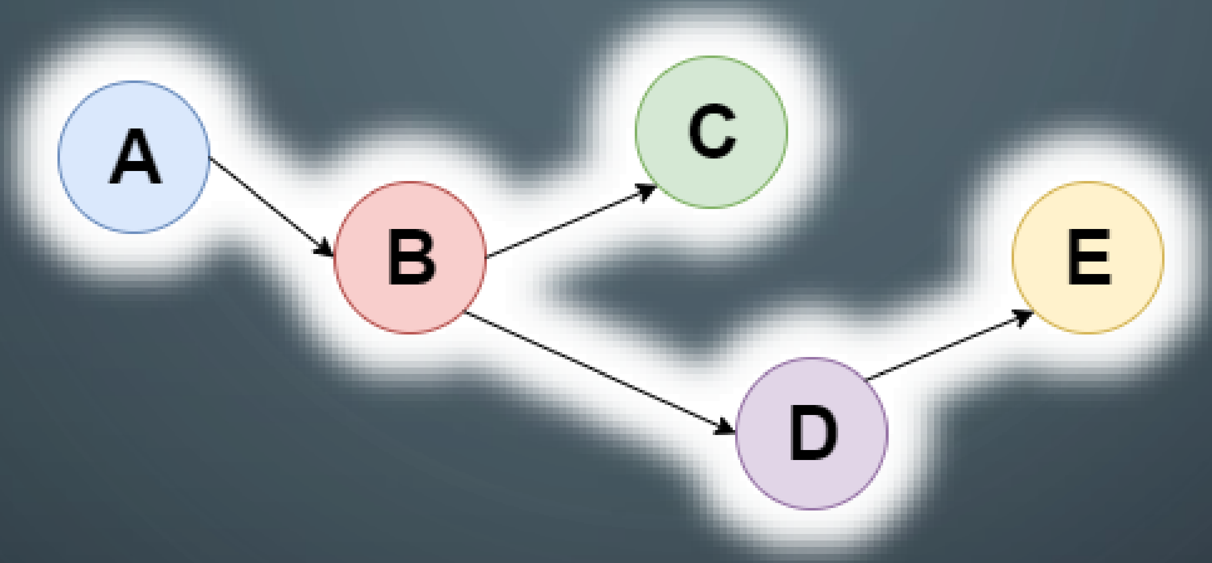
THE LINK STATE PACKET

- The router is then going to send out a *link state packet*, containing the addresses of each of it's neighbours, and the edge cost for each neighbour (!)
- This is an important distinction from Bellman-Ford, where routes sent the cost to get to X node, but not the edge cost of all neighbours

TELLING THE WORLD ABOUT YOUR NEIGHBOUR

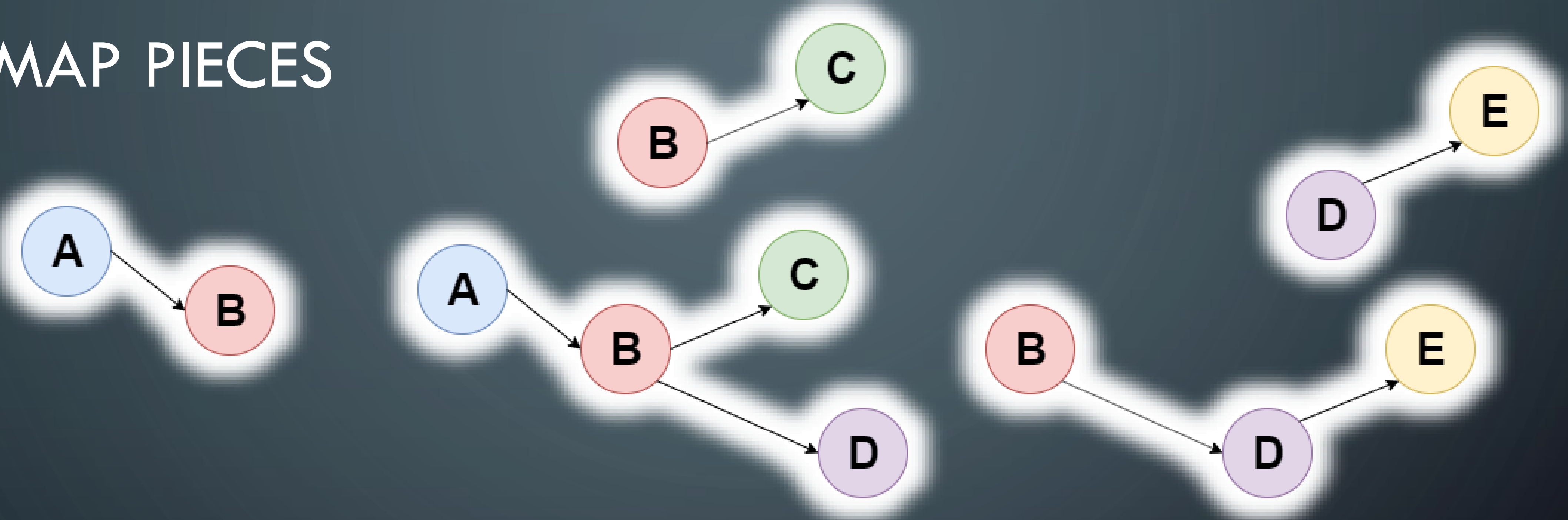
- The *link state packet* is sent via flooding to every other node in the network
- This can be done without routing → flooded packets are simply auto-forwarded through connected interfaces
- Every other router in the autonomous system receives this information, and proceeds to update their own information

VISUALIZING THIS



- Consider this network – if every single node broadcast it's neighbours, how many LSPs will be sent? What will the LSPs look like?

MAP PIECES



- The five potential LSPs that would be flooded on this network
- Each of these smaller ‘fragments’ would be received by all routers in the network

MAKING THE WHOLE MAP

- Once LSPs have been received from all nodes in the network, the node can take all these fragments and use them to create a complete map of the entire network, including edge costs
- Hence, why we say in link-state protocols, each node maintains a *complete map of the topology*

ANOTHER IMPORTANT CONSIDERATION

- Link-state protocols often perform far fewer updates than distance-vector protocols (remember, RIP sends packets every 30 seconds)
- LSPs are generated primarily when changes occur (new neighbours, link failures, node failures)
- Why do you think this would be?

THE DOWNSIDE OF FLOODING

- Flooding on the network potentially produces an exponential number of packets
- Doing this frequently can cause undue load and bandwidth usage, particularly if many routers are flooding at once
- Some means of avoiding this – using caching to prevent over-distribution of LSPs to neighbours

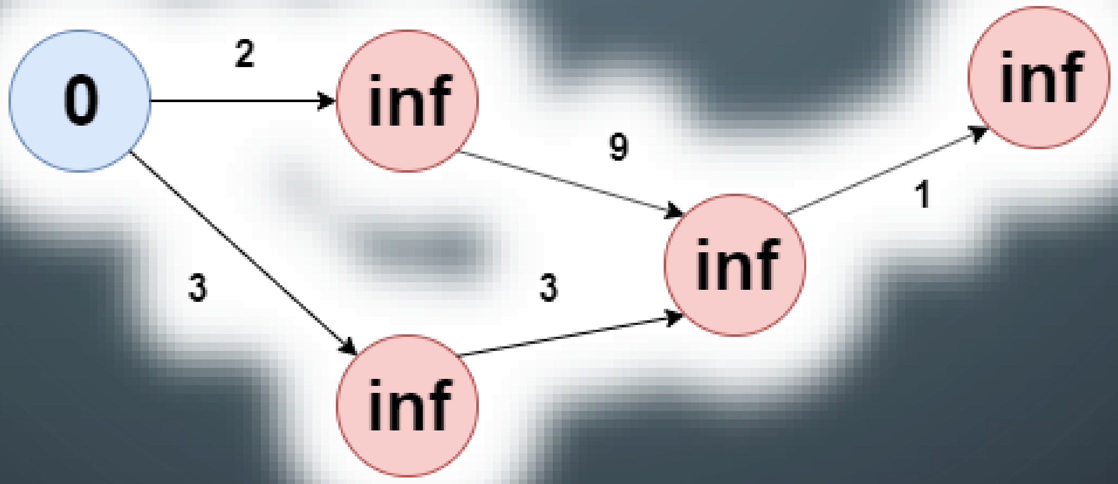
GENERATING A ROUTING TABLE

- Once nodes have assembled a map of the network topography from the LSPs, they can generate their actual routing table
- Since we have a complete network topology, we can use Dijkstra’s shortest path algorithm, originally developed in 1956

DIJKSTRA'S SHORTEST PATH FIRST (SPF)

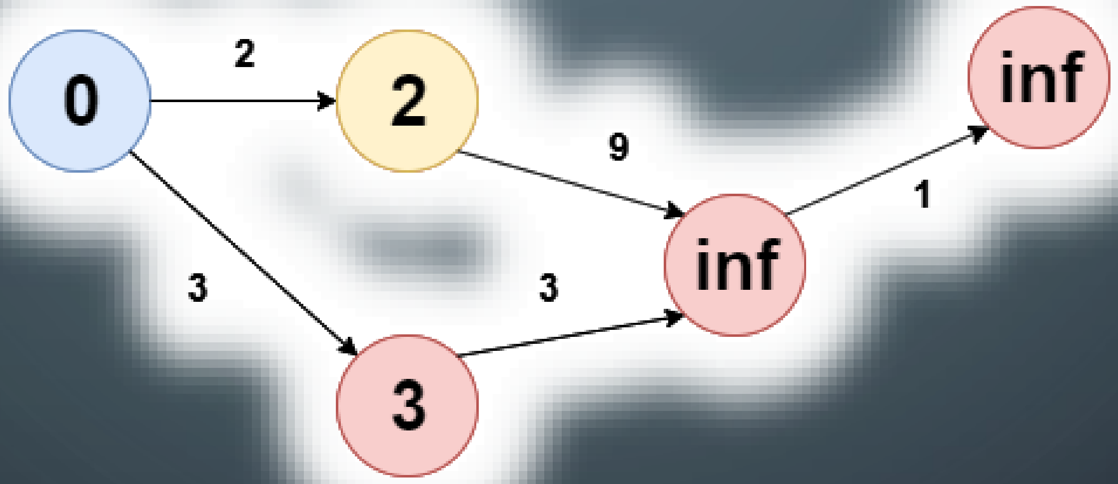
- Begin by labelling every node as infinite distance away
- From your starting point, pick an unvisited node with the closest distance (lowest cost)
- Proceed to check if there are shorter routes to it's neighbours and update the costs if necessary
- Flag this node as visited and move on to the next unvisited node with the lowest cost

DIJKSTRA'S WALKTHROUGH



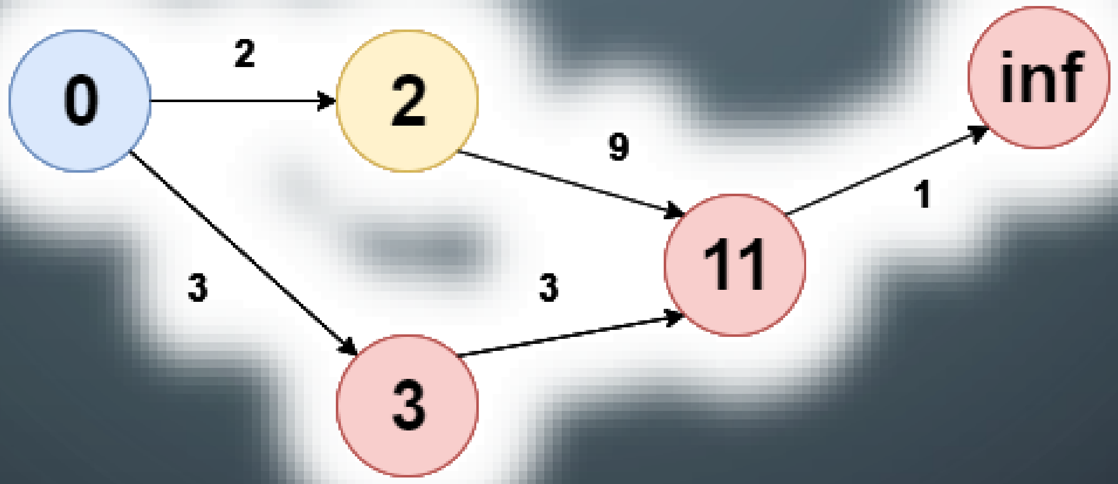
- First, we start off at our source (0) and label everything else as infinite

DIJKSTRA'S WALKTHROUGH



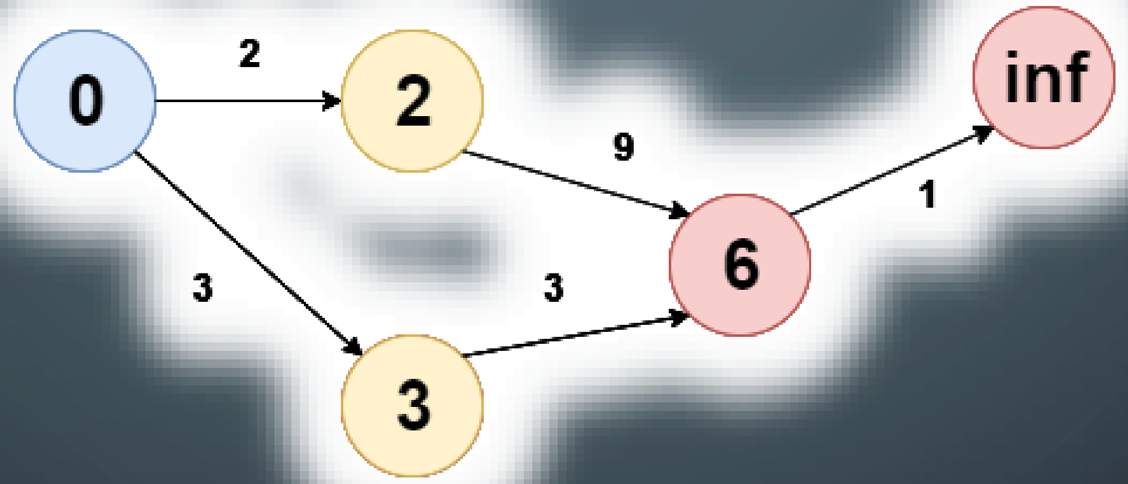
- 2 is lower cost than three, so we visit that node first

DIJKSTRA'S WALKTHROUGH



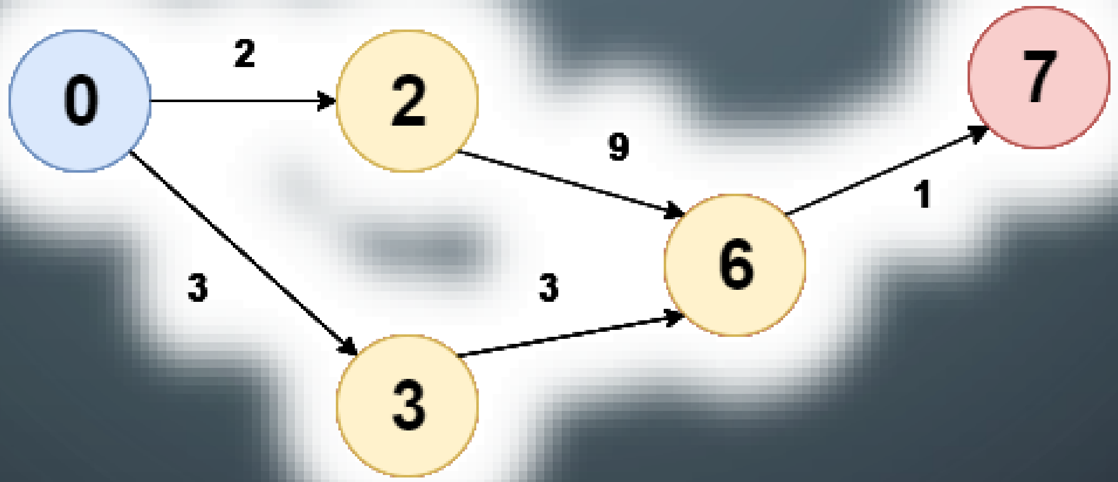
- We can now visit the neighbours of this node, and update their costs
- 2 + 9 = 11, which is less than infinity, so the new cost for that node becomes 11

DIJKSTRA'S WALKTHROUGH



- Now we go back and visit the lowest cost unvisited node, in this case with a cost of three
- We can update it's neighbour again, 3 + 3 = 6, which is less than 11 → new shortest path found!

DIJKSTRA'S WALKTHROUGH



- Then we would move onto the next node (now cost 6), whose neighbour we would visit (cost 7), and the shortest paths are found

HOW IT RELATES TO LINK-STATE PROTOCOL

- Each node performs Dijkstra's SPF algorithm (or a variant) to determine the shortest path between it and all other (known) nodes in the network
- The first step in these shortest-paths is the next-hop for traffic from the node to a destination

OPEN SHORTEST PATH FIRST (OSPF)

WHAT IS OSPF?

- *Open Shortest Path First* is a popular IGP routing protocol that uses link-state routing
- Standardized initially in 1989, but further revisions for things like IPv6 support
- "Open" refers to the free and open nature of the specification (no licensing cost to implement)

MESSAGES IN OSPF

- In addition to the LSP (advertisement), OSPF uses several other types of message to pass information between nodes:
    - Hello messages – used to discover neighbouring routers/nodes
    - Link status requests, updates, and acknowledgements – used to query and respond to other nodes regarding link status

GREETING YOUR NEIGHBOURS

- In OSPF, the Hello message is part of a state machine that is used to help establish connections between neighbours
- When a Hello message is received, a two-way communication channel is opened directly using IP datagrams (no TCP or UDP carrier protocol)
- Via this communication channel, topology information is shared via LSPs until both nodes have synchronous (identical) maps

INTERROGATING YOUR NEIGHBOURS

- Routers may also send Link State Requests (LSRs), which are a request from one router to another for updated information
- In response, the queried router will send back a Link State Update (LSU) containing the updated topology
- The original sender of the LSR message then sends an acknowledgement back that it received the LSU

AN IMPORTANT NOTE ON LINK COST

- In the OSPF specifications, the metric for link (edge) cost is defined, but not specified as any particular value
- This was done purposefully, so that network administrators can choose whichever metric(s) they consider most important for their particular network
- Metrics that are based on link bandwidth are popular (prioritize links with higher bandwidth/lower % cost)

AREAS IN OSPF

- An important aspect of the OSPF system is the idea of *areas*
- Areas are effectively sub-regions of the autonomous system that are used to help isolate the network into smaller groupings of nodes
- This is in part driven by the flooding nature of LSP advertisements → exponential packets, so reduction of node group sizes is very important

SPECIAL AREA DEFINITIONS

- OSPF defines some special areas that have certain characteristics, as well as router types dependent on their positioning within the areas
  - Backbone Areas
    - Interfaced with backbone routers
  - Stub Areas
  - Transit Areas
    - Area border routers – carry traffic between areas

BACKBONE AREAS

- Backbone areas in an OSPF autonomous system are connected to every other area in the system
- Traffic can transit between areas via the backbone area
- Backbone routers are routers that have an interface into the backbone area

STUB AREAS

- Stub areas are effectively similar to the idea of stub autonomous systems that we discussed previously
- Stub areas don't receive advertisements from external AS's, and routing takes place along a predefined route (often into the router backbone)
- Depending on configuration, they may not even receive advertisements from other areas within the AS

TRANSIT AREAS

- A transit area, like a transit autonomous system, allows traffic to flow through
- Traffic is usually not generated within the transit area, but is instead generated outside in a different area, and then travels through the transit area on it’s way to another area or destination

SOME NOTES ON AREAS

- An area doesn’t necessarily have to be defined as backbone, stub, or transit
  - These are special designations for areas that do specific things
- Areas are defined by an internal, 32-bit address. This is *not* an IPv4 address, but entirely internal to the autonomous system

A FINAL POINT ON OSPF ROUTING

- It is common in OSPF (and other link-state routing protocols) for routers to perform *load balancing*
- Since there will often be more than one path between two locations (even if those paths have slightly different costs), routers have some choice in how they send traffic
- They can use this choice to help even out load across the network by distributing traffic across multiple links

NEXT WEEK

- Quiz on Friday
- Wrapping up IGP (IS-IS protocol), moving into looking glass servers and routing tables

SCHEDULE

1. *Recap*
2. *Link-State Protocols*
   1. *How L-S protocols work*
   2. *The Dijkstra SPF*
3. *OSPF*

SO LONG, FOLKS!