# 49202 Communication Protocols

## The Network Layer - Part 3

Daniel R. Franklin

Faculty of Engineering & IT

April 30, 2024

# Network layer - continued

- Today we will finish our discussion of the network layer:
  - Interior and exterior routing protocols

# Routing protocols

- So far, we have discussed IP routing in general terms:
    - An IP datagram arrives at a router
    - The router inspects the destination IP address
    - It iterates through each entries in its routing table, bitwise-ANDing the datagram's destination address with the netmask in the entry and comparing the result with the destination prefix in the entry
    - The longest match, if any, is used to decide how to deliver the datagram, if possible - either locally or forwarding via the specified interface to the specified next-hop router
- So how do we construct the routing table?

# Static Routing

- The simplest approach: each host and router has a static, manually-configured routing table, with entries for each destination prefix that we know about
- Routes are *fixed* - they do not adapt to changing network conditions
  - Advantage: no protocol is needed - no communications overhead
  - Disadvantages: no automatic optimisation of routes, no ability to recovery from failure or adapt to availability of new links, easy to make mistakes
- There is one place where static routing is both simple and useful: *stub networks*
  - A stub network is a network with only a *single* router connecting it to the outside world
  - Hosts in this network can have use a simple routing table with only two entries - local delivery (the host's own network) and non-local delivery (everything else)
  - The everything-else route prefix is 0.0.0.0/0 (all IP addresses will match)

# Dynamic routing

- More complex networks require a *routing protocol* to *automatically*
  - Discover the topology of the network;
  - Optimise the routes to all destinations; and
  - Re-optimise their routing table based on new information (e.g. addition or removal of a link)
- Routing decisions are based on minimising an arbitrary **cost measure** or **metric**
- More hops (links) in a path = more cost (adding more hops only increase total cost of a path)
- Some links may be considered more 'costly' than others:
  - Low-bandwidth or high-latency links
  - Literally expensive-to-operate ($) links - e.g. satellite backup links
- Therefore, the least-cost path may not necessarily be the one with the fewest hops

# Routing stability

- When the topology of the network changes (addition or removal of a router or link), there is a period of *inconsistency* which exists while the updates are propagating through the network
- During this time, **routing loops** may occur
- Packets from a single source-destination stream may also take different routes, arriving out-of-order
- The period of transition between the time at which a topology change occurs and all routing tables across the network being fully updated is called the **convergence period**

# Interior and exterior routing protocols

- **Autonomous systems** (ASs) refer to large networks operated by a single administrative authority - for example, the UTS network or the Telstra network.
- The Internet Assigned Numbers Authority (IANA) assigns each a number: `https://www.iana.org/assignments/as-numbers/`
- Based on their scope, routing protocols are classified as **interior** or **exterior**
  - **Interior routing protocols** (also known as interior gateway protocols) manage the creation and maintenance of routing tables inside an autonomous system. Here, changes happen *frequently* and routing may be quite complex
  - **Exterior routing protocols** (also known as exterior gateway protocols) manage the creation and maintenance of routing tables across autonomous systems. Between these systems, changes happen *rarely*. Due to the high traffic volumes between ASs, the tables should be kept as small and simple as possible
- The dominant interior routing protocol today is Open Shortest Path First (OSPFv2 for IPv4, OSPFv3 for IPv6); the dominant exterior routing protocol is Border Gateway Protocol (BGP).

# Least cost routing algorithms

- Both interior and exterior routing protocols aim to minimise the *cost* of routes to each known destination network
- Problem: Given a cost assigned to each link between two nodes $X$ and $Y$, find the least-cost path
- There are two basic strategies which may be used to solve this problem:
  - Distance vector routing (based on the Bellman-Ford algorithm) - sometimes termed *routing by (indirect) rumour*
  - Link state routing (based on Dijkstra's algorithm) - *routing by (direct) knowledge*
- These algorithms achieve the same aim but differ in terms of performance (convergence time) complexity (processing and communications overhead)

# Distance vector routing

- Distance vector routing is based on the Bellman-Ford algorithm. Generally, the cost metric adopted is a simple hop count - this is the *distance* that we try to minimise.
- Each node maintains a set of minimum distance (in terms of hops) between it and all other nodes in the network (initially distances are set to infinity, except for links to immediate neighbours)
- Nodes exchange this information with immediate neighbours *only*, based on a **periodic timer mechanism** (e.g. sending an update every 30 seconds)
- When a node $x$ receives an update from node $y$, it compares its current minimum-distance vectors with the sum of the distance to node $y$ and the distances listed in $y$'s distance vector

# Distance vector routing

- If a shorter path exists, $x$ replaces the old entry in its distance vector
- If any elements have changed, the new distance vector is distributed to all one-hop neighbours (no update if no change)
- Distance vector routing works well, but suffers from a **large delay** when routing updates need to propagate over many hops:
  - The **upper bound** (worst-case) convergence time is proportional to the **maximum** of {the **minimum** number of hops between any pair of nodes}.
- Interior routing protocol **RIP** uses distance vector routing, while exterior routing protocol **BGP** employs some DV concepts and are used to manage routes between large organisations on the Internet (actually the approach used is called **path vector routing**).
- Notes: The general computational complexity of the Bellman-Ford algorithm, in a network with $V$ nodes and $E$ edges, is between $O(E)$ and $O(VE)$.

# Link state routing

- Link state routing is also a distributed routing protocol in which each node $n$ employs *flooding* to distribute information about the cost of all links to which it is connected
- The algorithm allows all nodes in the network to quickly build up a map of the network, with the associated costs on each link.
- Each node then performs a complete computation of the best routes from itself to all other nodes, using the “shortest path first” **Dijkstra algorithm**
- The result of computation will be the next hop for each entry in the routing table
- The general computational complexity of Dijkstra’s algorithm, in a network with $V$ nodes and $E$ edges, is $O(V^2)$; if $E << V^2$, this may be reduced to $O(V + E)\log V)$.

# Open shortest path first (OSPF)

- The OSPF protocol implements link-state routing and runs directly on top of the Internet's network layer (like with ICMP, no transport layer is used)
- On router startup, a "hello protocol" is used to discover neighbouring routers which speak OSPF; **hello packets** are sent to special **multicast all-routers IP address** `224.0.0.5` every "hello-interval" seconds (typically 10).
- A router seeing an initial **hello** will send a unicast response in reply; now the routers are aware of their **adjacency** on this particular link
- If the routers are in a single *broadcast domain* (e.g. connected to a switch), the go through a process for election of a **designated router** and **backup designated router**. This is not needed if the link is strictly point-to-point (e.g. a simple cable).
- Routers wishing to share updated link state information inside a broadcast domain will send this to both the DR and the BDR only

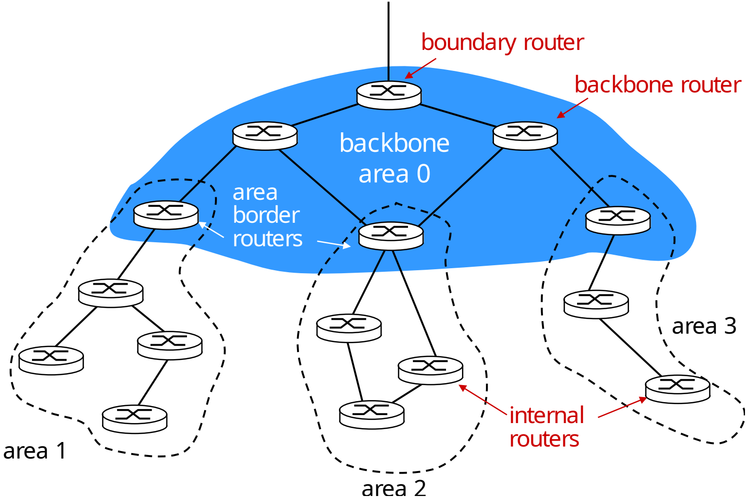
# Open shortest path first (OSPF)

- Initial synchronisation of router databases is performed through the "exchange" protocol - routers exchange their current databases via one or more **database description** packet exchanges.
- A change of link state is distributed *immediately* to all nodes via a **flooding protocol**
- The node detecting the change will send a **link state advertisement** to its immediate neighbours who in turn relay the update to others
- The flooding process traverses the entire network within a few seconds at the most. Duplicate LSAs are discarded.
- The router updates its **link state database** (all knowledge of all link states), and then uses **Dijkstra's Algorithm** to calculate an optimised routing table.

# OSPF areas

- OSPF is a hierarchical routing protocol
- A typical OSPF network consists of a **backbone** core network (Area 0) and other numbered areas
- Routers may have different interfaces in different areas - therefore, routers constitute the borders of OSPF areas. These are called Area Border Routers (ABRs).
- Routers with external connectivity are called Autonomous System Border Routers (ASBRs).
- Different types of link state advertisements can be created in response to network events; these can be local to a link / layer 2 network segment, OSPF area or the entire autonomous system

# OSPF area hierarchy



# OSPF LSA types

- There are currently 11 types of OSPF link state advertisements - some examples include:
    - Type 1: OSPF Router LSA. The most commonly seen LSA type, generated by routers; describes themselves, their own interfaces, costs and known neighbouring routers
    - Type 2: OSPF Network LSA. Sent by the DR to other routers; identifies DR & BDR
    - Type 3: OSPF Summary LSA. Generated by ABRs; list all prefixes available in an Area to routers in other Areas
    - Type 4: OSPF ASBR Summary LSA
    - Type 5: OSPF ASBR External LSA; tells routers inside an Area how to get out via the ASBR.
    - ...plus types 6-11 (9-11 are ‘opaque’ and can carry auxilliary information with link, area and AS scope)
- For example, if a link goes down or comes up, a Type 1 area-local LSA will be generated (you will see these in the lab, along with Type 5).

# Dijkstra's algorithm

- Stated as follows:
  - Find the shortest paths from a given source node to all other nodes by developing paths in rder of increasing path length
  - The algorithm proceeds in stages...
  - By the $k$th stage, the shortest paths to the $k$ nodes closest to the source node have been determined

# Definitions

- $s$ = source node
- $w(i, j)$ = link cost from node $i$ to node $j$; $w(i, j) = \infty$ if there is no link from node $i$ to node $j$
  - **Important**: $w(i, j)$ is not necessarily equal to $w(j, i)$
  - Linux uses a default link cost of 100 units (note: this is in the **egress** direction on a router)
  - Some other systems (e.g. Alcatel, Cisco) use a link cost equal to a reference bandwidth (e.g. 10 Gb/s) divided by the link bandwidth
- $T$ = set of nodes processed up to this point - we then add destination nodes one at a time
- $L(n)$ = total least-cost path from node $s$ to node $n$ known at this point in time

# Dijkstra's algorithm

- Initialize:
  - $T = \{s\}$ (i.e. only the source node at the start)
  - $L(s) = 0$ (cost to myself is 0)
  - $L(n) = w(s, n)$ for $n \neq s$, i.e., for nodes other than myself:
    - If we have a direct connection to that node, the cost is the link cost to that node;
    - Otherwise, the cost is infinite (at the start).
- Find a node $v$ that is not yet in set $T$ such that

$$L(v) = \min_{z \notin T} L(z)$$

- That is, find $v$ amongst nodes not yet part of set $T$ such that the total cost from node $s$ to node $v$ is the lowest amongst all such nodes
- Incorporate $v$ into $T$.

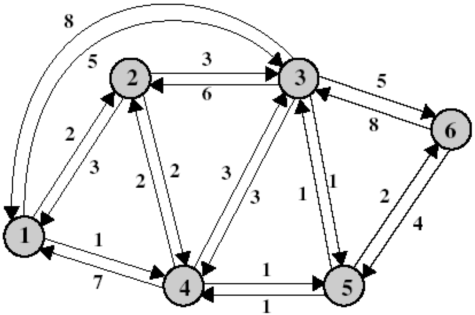
# Dijkstra's algorithm

- Update all the currently known least cost path list $L(n)$, including paths to destinations which can be reached via node $v$:

$$\forall n : L(n) = \lfloor \{ L(n),\ L(v) + w(v, n) \} \rfloor$$

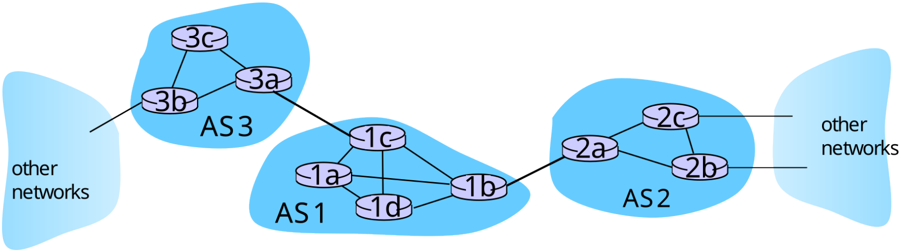
  (yes, set theory notation is beautiful... or not)
- This means for all nodes in our list, **if** we can find a **lower cost path** to node $n$ via node $v$, we replace the old $L(n)$ with $L(v)$ plus the **additional** cost of getting from node $v$ to node $n$
- We are finished when all nodes have been added to set $T$
- At the end, the set $L(n)$ lists the minimum costs from node $s$ to $n$

# Dijkstra's algorithm example (calculated on Node 1)




| Iteration | T | L(2) | Path | L(3) | Path | L(4) | Path | L(5) | Path | L(6) | Path |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 1-2 | 5 | 1-3 | 1 | 1-4 | $\infty$ | - | $\infty$ | - |
| 2 | 1,4 | 2 | 1-2 | 4 | 1-4-3 | 1 | 1-4 | 2 | 1-4-5 | $\infty$ | - |
| 3 | 1,2,4 | 2 | 1-2 | 4 | 1-4-3 | 1 | 1-4 | 2 | 1-4-5 | $\infty$ | - |
| 4 | 1,2,4,5 | 2 | 1-2 | 3 | 1-4-5-3 | 1 | 1-4 | 2 | 1-4-5 | 4 | 1-4-5-6 |
| 5 | 1,2,3,4,5 | 2 | 1-2 | 3 | 1-4-5-3 | 1 | 1-4 | 2 | 1-4-5 | 4 | 1-4-5-6 |
| 6 | 1,2,3,4,5,6 | 2 | 1-2 | 3 | 1-4-5-3 | 1 | 1-4 | 2 | 1-4-5 | 4 | 1-4-5-6 |

# Exterior routing protocols - inter-AS routing

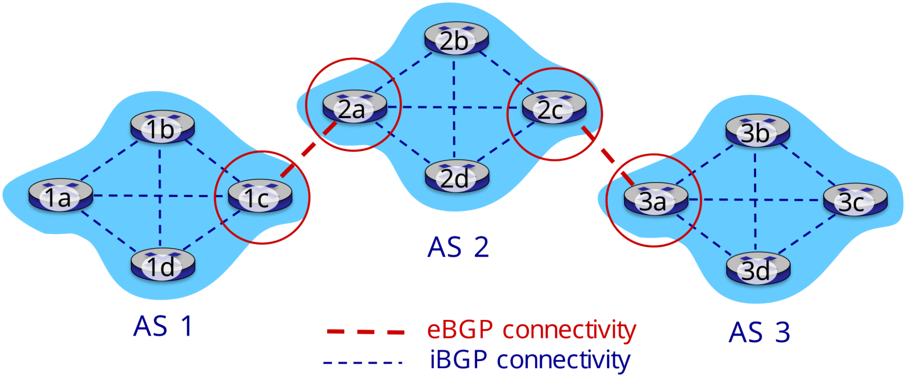



- Suppose that a router in AS1 receives a datagram destined outside of AS1:
  - The router should forward packet to a gateway router, but which one?
- AS1 must:
  1. Learn which destinations are reachable via AS2, and which via AS3
  2. Propagate this reachability information to all routers in AS1
- This is the job of inter-AS routing!

# Exterior routing protocols - border gateway protocol

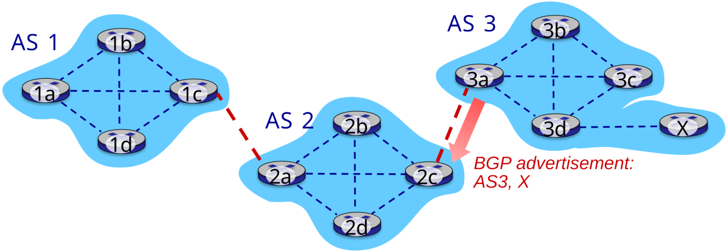
- BGP (Border Gateway Protocol): the de facto inter-domain routing protocol
- Based on distance vector routing
- BGP provides each AS a means to:
  - Obtain subnet reachability information from neighboring ASes: eBGP
  - Propagate reachability information to all AS-internal routers: iBGP
  - Determine “good” routes to other networks based on reachability information and policy allows subnet to advertise its existence to rest of Internet: “I am here”

# eBGP-iBGP connections




- Gateway routers (red circles) run both eBGP and iBGP protocols

# BGP basics

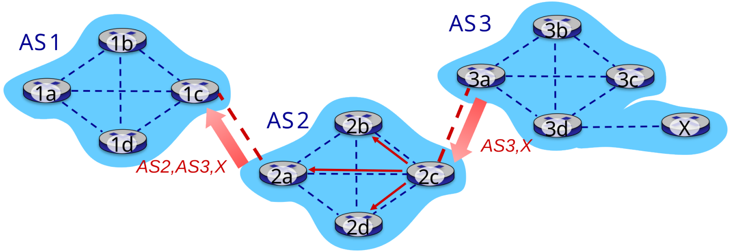



- BGP session: two BGP routers ("peers") exchange BGP messages over semi-permanent TCP connections:
  - Advertising paths to different destination network prefixes (BGP is a "path vector" protocol)
- When AS3 gateway router 3a advertises path AS3,X to AS2 gateway router 2c, **AS3 is promising to AS2 that it will forward datagrams towards X**

# Path attributes and BGP routes

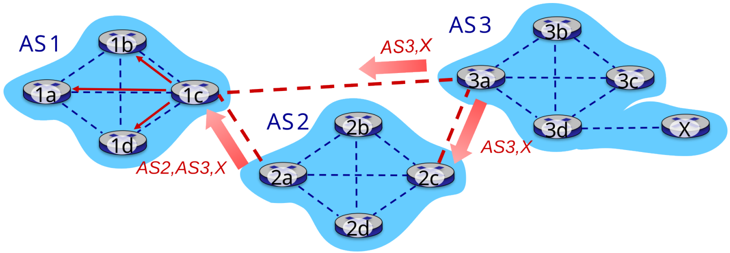
- Advertised prefix includes BGP attributes
  - Prefix + attributes = “route”
- Two important attributes:
  - AS-PATH: list of ASes through which prefix advertisement has passed
  - NEXT-HOP: indicates specific internal-AS router to next-hop AS
- Policy-based routing:
  - Gateway receiving route advertisement uses import policy to accept/decline path (e.g., never route through AS Y).
  - AS policy also determines whether to advertise path to other other neighboring ASes

# BGP path advertisement




- AS2 router 2c receives path advertisement **AS3,X** (via eBGP) from AS3 router 3a
- Based on AS2 policy, AS2 router 2c accepts path **AS3,X**, and propagates it (via iBGP) to all AS2 routers
- Based on AS2 policy, AS2 router 2a advertises (via eBGP) path **AS2,AS3,X** to AS1 router 1c

# BGP path advertisement

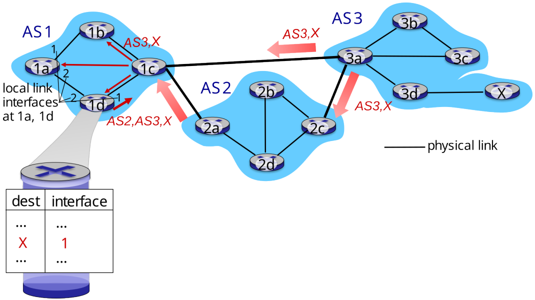



- Gateway router may learn about multiple paths to destination:
  - AS1 gateway router 1c learns path **AS2,AS3,X** from 2a
  - AS1 gateway router 1c learns path **AS3,X** from 3a
  - Based on **policy**, AS1 gateway router 1c chooses path **AS3,X**, and advertises this path within AS1 via iBGP (get to AS3 via router 1c)

# BGP messages

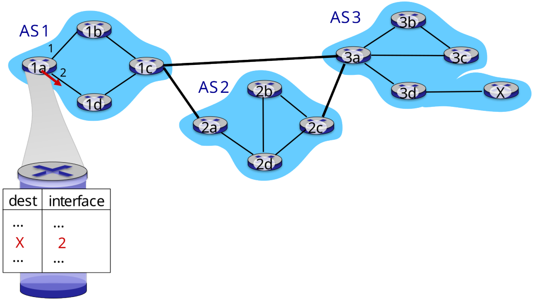
- BGP messages exchanged between peers over a TCP connection
- Four message types:
  - OPEN: opens TCP connection to remote BGP peer and authenticates sending BGP peer (very important!)
  - UPDATE: advertises new path (or withdraws old)
  - KEEPALIVE: keeps connection alive in absence of UPDATES; also ACKs OPEN request
  - NOTIFICATION: reports errors in previous message; also used to close connection

# BGP, OSPF, forwarding table entries




- Q: how does router set forwarding table entry to distant prefix?
- Recall: 1a, 1b, 1d learn about dest X via iBGP from 1c: “path to X goes through 1c”
- 1d: OSPF intra-domain routing: to get to 1c, forward over outgoing local interface 1

# BGP, OSPF, forwarding table entries

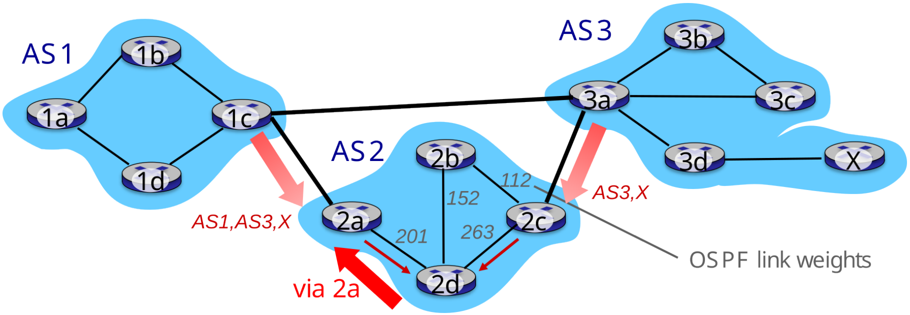



- Q: how does router set forwarding table entry to distant prefix?
- recall: 1a, 1b, 1c learn about dest X via iBGP from 1c: "path to X goes through 1c"
- 1d: OSPF intra-domain routing: to get to 1c, forward over outgoing local interface 1
- 1a: OSPF intra-domain routing: to get to 1c, forward over outgoing local interface 2

# BGP route selection

- A router may learn about more than one route to destination AS, selects route based on:
  1. Local preference value attribute: policy decision
  2. Shortest AS-PATH
  3. Closest NEXT-HOP router: hot potato routing
  4. Additional criteria

# Hot Potato Routing




- 2d learns (via iBGP) it can route to X via 2a or 2c
- *Hot potato routing*: choose local gateway that has least intra-domain cost (e.g., 2d chooses 2a, even though more AS hops to X): **don't worry about inter-domain cost!**

# Why different Intra-, Inter-AS routing?

- Policy:
  - Intra-AS: single admin, so no policy decisions needed
  - Inter-AS: admin wants control (using policy) over how its traffic routed, who routes through its network.
- Performance:
  - Intra-AS: can focus on performance
  - Inter-AS: policy may dominate over performance scale:
- Hierarchical routing saves table size, reduced update traffic

# The scale and growth of BGP

- As of April 2024, there are 970710 active entries in the global BGP route table out of a total of 2874697 known routes
- There are 75986 unique ASs in the Internet, of which 64870 are origin-only (i.e. only an endpoint, not transit). The rest are mixed or transit-only.