CS4457 Notes

Jonathan Kerr (jdk2fy), Hua Uehara (hlu8jm)

February 11, 2020

1 Banyan Switch

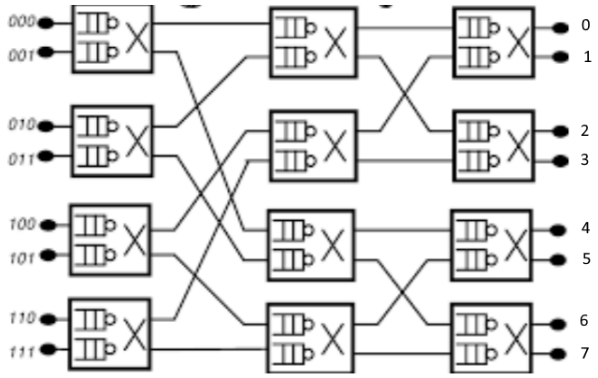

Figure 1: Banyan Switch

A banyan switch overcomes memory bus contention by using a matrix of switching elements (Figure 1). Each entering packet contains a series of bits in its routing header indicating the packet's destination. For example, the second input from the top in Figure 1 has header $001_2 = 6$, and so the packet is directed to the 6th output on the right. (Note Figure 1 uses little endian).

Each individual switching element inspects one bit in the packet header. The packet is sent to the higher output if the bit is 1 and to the lower output if the bit is 0. For example, the top left element in Figure 1 reads the most significant destination bit from each packet it receives. The first packet's most significant bit is 0, so the packet is sent to the upper output, while the second packet's bit is 1, so it is sent to the lower output (Figure 2). The second switching element reached by a packet considers the second most significant destination bit, the third element considers the third most significant bit, and so on.

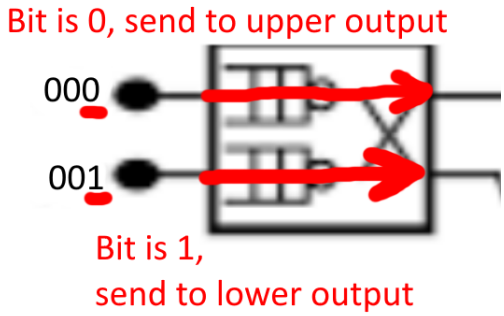

Figure 2: Banyan switching element

2 Distance Vector Algorithm

Each node x maintains a distance vector $D_x = [D_x(y) \; \forall \; y \; \epsilon N]$, where $D_x(y)$ is the minimum distance from x to y, and N is the set of all nodes in the network. Every neighbor v of x informs x of its distance vector D_v. Then x can calculate its distance vector using the Bellman-Ford equation:

$$D_x(y) = \min_{v \varepsilon N(x)} \{c(x,v) + D_v(y)\},$$

where $N(x)$ is the set of neighbors of x and $c(x,v)$ is the cost of the edge between x and v.

Whenever x updates its distance vector, x broadcasts its new distance vector to its neighbors. In turn, every neighbor v will recalculate its distance vector and, if D_v is updated, broadcast D_v. This chain of updates and broadcasts continues until every node's own distance vector contains the true least cost to every other node.

If x is routing a packet destined for y, the least cost path is attained by forwarding the packet to x's neighbor v that satisfies $D_x(y) = c(x,v) + D_v(y)$.

2.1 Link cost changes

Suppose node y detects a change in the link cost to a neighbor, as depicted in Figure 3. y must then update its distance vector and, if the vector changes, broadcast its updated distance vector to its neighbors.

Good news travels fast: If the link cost decreases, as in Figure 3, then the distance vectors of the nodes are quickly updated to the new, correct values. In the Figure 3 example, distance vectors are stabilized after y and z each update their distance vector once.

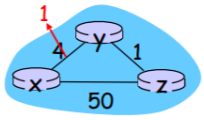

Figure 3: Link cost decrease

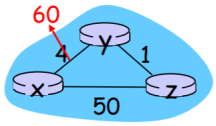

Figure 4: Link cost increase

Bad news travels slow: Now suppose y detects that its link cost to x has increased from 4 to 60, as in Figure 4. Before the change occurred, $D_y(x) = 4$ and $D_z(x) = 5$.

1. After the link cost change, y will update $D_y(x)$ to $c(y,z) + D_z(x) = 1 + 5 = 6$. Note that this update uses the outdated value $D_z(x) = 5$, which we (with a global perspective) can see is invalid. Since y cannot see the whole network, y does not know $D_z(x) = 5$ is invalid.
2. Next, y broadcasts the updated value $D_y(x) = 6$ to z. Then z updates $D_z(x) = c(z,y) + D_y(x) = 1 + 6 = 7$.
3. Next, z broadcasts $D_z(x) = 7$ to y. Then y updates $D_y(x) = c(y,z) + D_z(x) = 1 + 7 = 8$.

This process continues for a total of 44 iterations until y's and z's distance vectors contain the true values $D_y(x) = 51$ and $D_z(x) = 50$. This gradual rise of $D_y(x)$ and $D_z(x)$ to their true values is called the **count-to-infinity problem**.

Poisoned Reverse

The count-to-infinity example in Figure 4 may be solved using poisoned reverse. Under poisoned reverse, a node z tells neighbor y that $d_z(x) = \infty$ if z's next hop towards x is y. For example, in the network pictured in Figure 4 prior to the link cost change, z tells y that $d_z(x) = \infty$. Thus, after the link cost change y will not rely on an outdated value of $d_z(x)$ and the count-to-infinity problem does not occur.

However, the poisoned reverse technique does not always solve the count-to-infinity problem. Poisoned reverse does not prevent the count-to-infinity problem in loops with more than two nodes.

2.2 Distance Vector Algorithm vs Link-State Algorithm (Dijkstra's)

Let N be the number of nodes in the network and let E be the number of edges.

	Distance Vector Algorithm	Link-State Algorithm
Message Complexity	Messages exchanged between neighbors at each iteration. Number of iterations varies.	O(N E)
Speed of Convergence	DV algorithm can converge slowly due to count-to-infinity problem.	$O(N^2)$
Robustness	Node can broadcast incorrect least-cost path, which error may be propagated through the network.	Node can broadcast incorrect link cost for one of its own edges. Each node computes its forwarding table separately, thus providing some robustness.

3 Autonomous Systems

The distance vector algorithm cannot scale to a network with a large number of routers, because routing tables would become too large and the exchange of distance vectors would flood the network. Instead, **the internet is a network of networks**. That is, routers are aggregated into autonomous systems (AS), as in Figure 5.

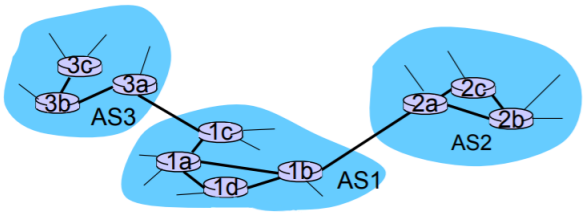

Figure 5: Three autonomous systems

3.1 Intra-AS routing: OSPF

Within an AS, all routers use one intra-domain routing protocol, such as RIP, OSPF, or IS-IS. RIP is a distance vector protocol, similar to the distance vector algorithm described above. Open Shortest Path First (OSPF) and the nearly identical IS-IS protocol use a link-state algorithm. Each router in the AS broadcasts its link-state information. Each router thus receives the costs of all the links in the AS, and each router can construct a graph of the entire AS. Each router then uses Dijkstra's algorithm to calculate shortest routes.

OSPF Features:

- **Security:** OSPF messages may be authenticated by signing messages with a secret key. This authentication distinguishes legitimate messages from fraudulent messages sent by malicious routers which are not privy to the key.
- **Multiple same-cost paths:** Unlike RIP, OSPF allows routing through multiple paths to a destination if the paths have the same cost.
- **Support for uni- and multicast routing:** Multicast OSPF (MOSPF) provides for multicast routing by using the existing OSPF graph.
- **Support for hierarchy:** An OSPF AS can be divided into areas. For example, the autonomous system in Figure 6 has been divided into four areas. Note that the fourth area is labeled as the backbone. The OSPF protocol is used for intra-area routing, and nodes only maintain a topology of their own area. Packets are routed out of an area through a border router. Border routers are located in both the backbone and another area. All inter-area routing requires routing through the backbone.

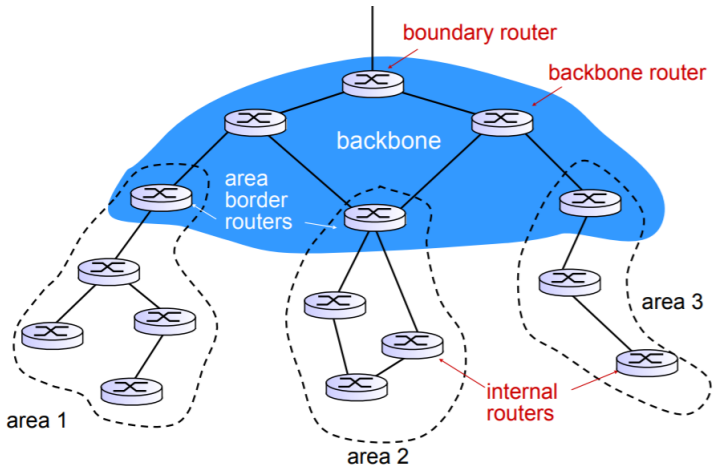

Figure 6: Hierarchical OSPF

3.2 Inter-AS routing: BGP

While intra-domain routing protocols may vary between ASes, the Border Gateway Protocol (BGP) is the de facto inter-domain routing protocol. BGP allows ASes to:

- Obtain subnet reachability information from neighboring ASes
- Propagate reachability information to AS-internal routers.
- Choose routes to other networks based on reachability information and policy

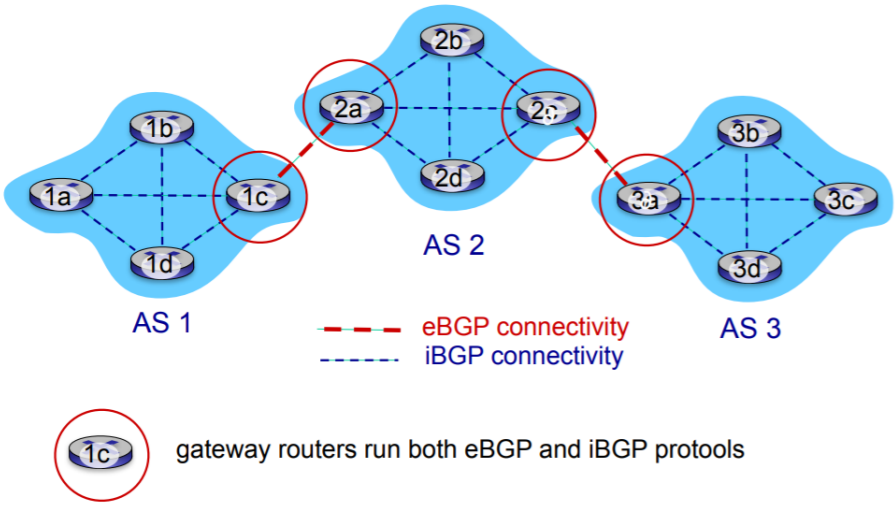

Figure 7: BGP sessions

Consider Figure 7. Gateway routers in separate ASes are connected by external BGP (eBGP) sessions. Router 1c, for example, may use its eBGP session with router 2a to advertise which subnets (represented by CIDR prefixes) are reachable through AS 1. When a gateway receives a route advertisement, the gateway decides whether to accept or decline the path based on its **import policy.** For example, in Figure 8 the gateway 1c may choose to decline the advertised path to X, because AS1 has a policy to never route through AS3. After a gateway receives (and accepts) a prefix advertisement through an eBGP connection, the gateway forwards the advertisement to all nodes in its AS via iBGP sessions. If a gateway learns about multiple paths to a destination, the gateway will prefer a path based on the AS's policy and advertise this preferred path via iBGP.
A prefix advertisement also includes some BGP attributes, most importantly:

1. **AS-PATH:** List of ASes through which the advertisement has passed. In Figure 8, 2a advertises to 1c a route to X. Included in this advertisement is the AS-PATH, "AS2,AS3."
2. **NEXT-HOP:** This attribute indicates the next-hop router in the path to the advertised subnet. In Figure 8, when 2a advertised the path to X to 1c, the next hop is 2a.

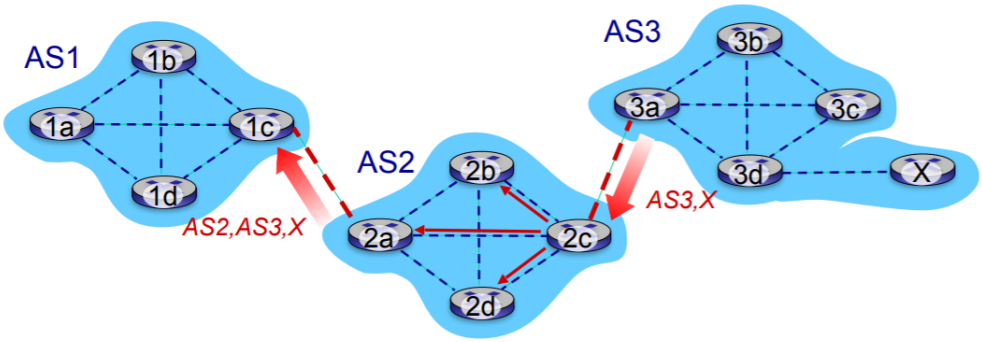

Figure 8: BGP advertisements for prefix X

BGP route selection
In Figure 9, router 2d receives iBGP messages from gateways 2a and 2c advertising two different routes to X. If a router learns of multiple routes to one prefix, one route is selected using the following process:

1. AS policy is used to assign a local preference attribute to each BGP route. The route with the highest local preference is selected.
2. If two or more routes have the same local preference attribute, the route with the shortest intra-AS route (as defined by the intra-AS routing algorithm) is selected.
3. If there is still a tie, the route with the least cost next hop is selected. For example, in Figure 9, 2d's next hop to X is either 2a or 2c. Since the path to 2a has a lesser cost than the path to 2c (201 compared to 263), 2a is the least cost next hop. Routing to the closest next hop router is called **hot potato routing**.

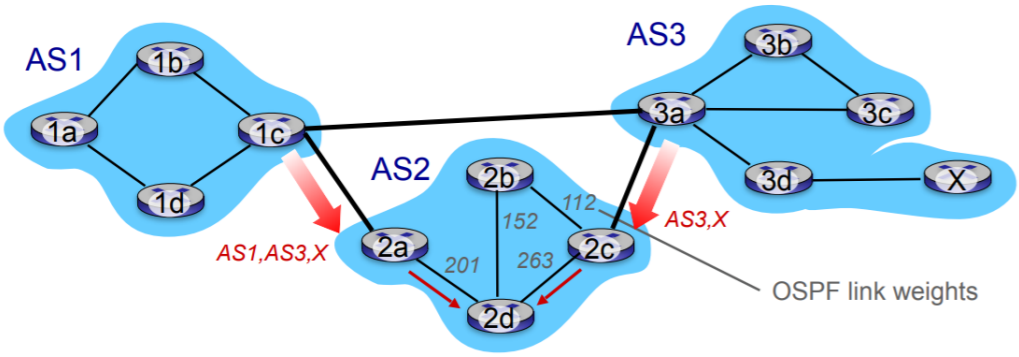

Figure 9: BGP route selection

Achieving policy via advertisements
Suppose the provider networks in Figure 10 wish only to route traffic that has a source and/or destination in one of its customer networks. For example, B may have a policy to only carry traffic to or from its customer X. A will advertise the path WA to B. B should choose *not* to advertise the path WAB to C. If C learns of the path WABC, then B may become responsible for routing traffic from C to W. Since none of W, A, and C are B's customers, routing their traffic violates B's policy.

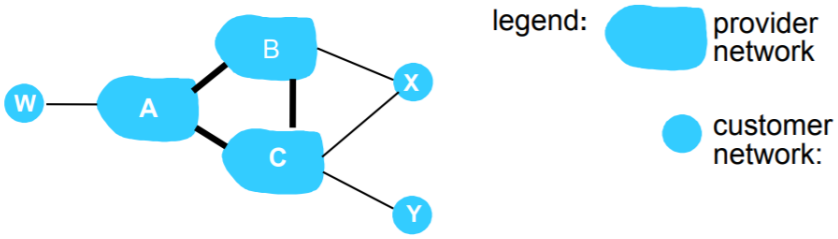

Figure 10: BGP advertisements may be used to enforce policy

BGP Hijacking

Fraudulent BGP advertisements may be created to maliciously reroute traffic. In April 2018, BGP hijacking was used to steal cryptocurrency. Figure 11 provides a rough sketch of the attack. Suppose there is a route through AS1 to the subnet 205.251.194.0/23, which includes the Amazon DNS server located at 205.251.195.239. During the attack, AS2 advertised a false route to 205.251.195.0/24. Since the prefix 205.251.195.0/24 is more specific than 205.251.194.0/23, ASes which receive and accept AS2's advertisement will route traffic to 205.251.195.239 through AS2.

After AS0 accepts AS2's false advertisement, user M1's DNS requests are routed to the fake DNS server. During the attack, the fake DNS server responded to DNS requests for "myetherwallet.com" with IPs of a fake version of the website hosted on the attacker's servers. Thus, the attacker could use login credential sent to the fake website to login to the victims' accounts on the legitimate website.

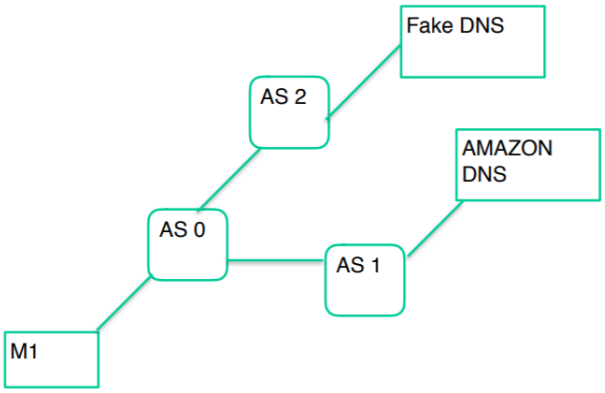

Figure 11: BGP hijacking