0. Introduction
- Today: routing, some addressing
- Enormous growth of Internet => routing protocols redesigned to scale, and also to enforce policy.

1. Routing
- Goal: allow each switch to know, for every node in the network, a min-cost route to that node (if one exists).
- Remember: networks aren't static. Link costs change, machines/links go down, etc.
- Our approach: distributed routing. Each node learns about the network and determines its own routes. In general:
  - Nodes use a HELLO protocol to discover neighbors
  - Nodes receive advertisements to learn about the network
  - Nodes integrate those advertisements to figure out routes
- Protocol 1: Distance-vector. Nodes advertise to neighbors with their cost to all known nodes, and update routes when an advertisement indicates that there is a better route.
- Protocol 2: Link-state: Nodes advertise to everyone (via flooding) their costs to their neighbors, and integrate using Dijkstra's.
  - You do NOT need to know Dijkstra's Algorithm for 6.033.
- Problem: DV and LS don't scale to the Internet
  - DV = low overhead, but convergence time is proportional to longest path. Good for small networks.
  - LS = fast convergence, but high overhead because of flooding. Good for MIT-sized networks, but not the Internet.

2. (Three ways we deal with) scale
- Path-vector routing
  - Like DV, but include the full path in the routing advertisements. Overhead increases (advs are larger), but convergence time decreases (avoid counting to infinity).
  - Overhead is still lower than LS's
- Routing Hierarchy
  - Internet is divided into Autonomous Systems (ASes). ASes are universities, ISPs, government branches, etc. Each AS has a unique ID (its AS number). There are tens of thousands of them (but not billions)
  - Use one routing protocol to route across ASes, and a different protocol to route within ASes.
    - Implies that there are devices on the edge of each AS that can "translate" between or "speak" both protocols.
  - BGP is the path-vector protocol used across ASes.
- Topological addressing
  - Despite being between ASes, BGP still routes to IP addresses
(e.g., to 18.0.0.1, not to AS3)
- Addresses are given to ASes in contiguous blocks, so that they can be specified succinctly via a particular notation ("CIDR" notation).
- Keeps advertisements small(er than they would be otherwise)

3. Policy Routing
- ASes also want to implement policy; they want "policy routing"
- Policy routing: switches make routing decisions based on some set of policies set by a human. Routing protocol must disseminate enough information to enable those policies
- What policies are typical in BGP? ASes don't want to send traffic on a path unless they have financial incentive to do so.
- Mechanism of enforcement: selective advertisements. AS1 won't tell AS2 about a path unless it will make money by letting AS2 use the path.
  => each AS will have a different view of the network, and that view will (almost certainly) *not* contain every physical link

4. Typical BGP Relationships (which will eventually lead us to typical BGP policies)
- Customer/provider
  - Customers pay for access (transit), which the provider provides
- Peers
  - Peers provide mutual access to a subset of each other's routing tables, namely, the subset that contains their transit customers
  - Why peer? Can save money and improve performance. Sometimes, it may be the only way to connect your customers to some part of the Internet.
  - Why *not* peer? You'd rather have customers.

5. BGP Relationships => BGP Export Policies
- First decision: which routes do I advertise to which neighbors? These are an AS's "export policies"
  - High-level: "Tell everyone about yourself (your internal IPs) and your customers; tell your customer about everyone."
  - More specifically:
    - Providers export customer's routes to everyone
    - A customer exports its provider's routes to *its* customers
      - These two should make sense: since the customer is paying for Internet, the provider should give them a route to as many destinations as possible. Similarly, the provider should allow *other* parts of the network to reach its customers.
      - AS exports *only* customer routes to peers.
        - Why not full table? AS doesn't want to provide transit for its peers; they're not paying it for transit.

Example: (Very similar to the example I gave in lecture)
T          <-- X buys transit from T
\|
Z === [ X ] === Y   <-- X peers with Z and Y
\|      |  |  |     |
C  C1  C2  C3      D

- Z will tell X about C; C is a customer of Z, and X and Z are peers
- X will tell Z, Y, and T about C1, C2, and C3.
- Y will tell X about D.
- X will *not* tell Y about C; it makes no money to provide transit from Y to C
- X doesn't tell Y about T; it would lose money to provide transit from Y to T.

- In example, Y appears disconnected from part of the network. BGP doesn't prevent this. In practice, it never happens.
- Almost every AS is a customer of someone else (i.e., Y would buy transit from someone)
  - Typically: small ASes buy Internet from Tier-3 ISPs, which buy Internet from Tier-2 ISPs, which buy Internet from Tier-1 ISPs. Tier-1's are huge; there are only a handful (10-15)
  - Additionally, all Tier-1 ISPs peer with one another. So each Tier-1 ISP can provide global connectivity.
  - This is an example where we need peering in order to reach part of the Internet.

6. BGP Relationships => BGP Import Policies
- If an AS hears about a route to X from multiple neighbors, how does it decide? These are its "import policies".
  - First: make money. Prefer routes via customers -- which you make money on -- to routes via peers -- which you don't make, but don't lose money on -- to routes via providers -- which you lose money on.
  - In the case of a tie (which happens often): there are a whole host of other attributes that BGP provides. A common one is AS-hop-count.
  - Each AS sets its own policies

7. BGP in light of distributed routing
- HELLO protocol: BGP sends KEEPALIVE messages to neighbors.
- Advertisements: sent to neighbors. Look different depending on which neighbor.
  - BGP runs on top of TCP, a reliable-transport protocol. Doesn't have to do periodic advertisements to handle failure. Instead, push advs when routes change.
- Integration: via policies described above
- Failures: routes can be explicitly withdrawn in BGP when they fail. Routing loops avoided because BGP is path-vector.

8. Problems with BGP
Does it scale? Well, it works on the Internet. But...
- BGP routing tables are getting big (exceeding the amount of
memory dedicated to the table in some switches).
- We see route instability due to misconfigurations or conflicting
AS policies. "Route-flap damping" (ignore advs about
frequently-changing routes) helps with this, but increases
convergence time.
- ASes can multi-home: buy Internet from more than one ISP,
usually for back-up or load-balancing. More multi-homed
networks => bigger routing tables. The load-balancing itself is
also tricky.
- iBGP. An AS actually has multiple BGP routers on its edge, and
a protocol called iBGP keeps them all in sync. iBGP requires an
AS's BGP routers to be connected in a complete graph, and so it
doesn't scale particularly well.
- Basically: Internet has grown enough that scalability of BGP is
becoming a concern.

Is it secure?
- Goodness no. ASes can advertise about a prefix that they don't
actually own.
- Similar problem (and solution) as in DNS. We'll talk more about
it after spring break.

Is it simple?
- The protocol itself: yes.
- BGP in practice: no. Again, mo' money, mo' problems. Also,
human operator error due to the complexity of setting the
policies.