01 - Internet Routing  
  
The next few lessons will cover internet routing. Contrary to what you might think, the internet is not a single network, but rather a collection of tens of thousands of independently operated networks, or autonomous systems. Sometimes simply called AS's. Networks such as Comcast, Georgia Tech, and Google, are different types of autononomous systems. An autononous system might be internet service provider, a content provider, a campus network or any other independently operated network. Now when you're sitting at home on Comcast and trying to reach content in Google or Georgia Tech, your traffic actually traverses multiple autonomous systems. This process of internet routing actually involves two distinct types of routing. One is intradomain routing, which is the process by which traffic is routed inside any single autonomous system. The other is interdomain routing, which is the process of routing traffic between autonomous systems. So computing a path between a node in an ISP like Comcast and another node in a network like Georgia Tech's involves computation of both intradomain paths and interdomain paths. In this part of the lesson we'll look at intradomain routing. Then we'll study interdomain routing, as well as the business relationships that make interdomain routing so complicated. So let's jump into our study of intradomain routing and topology.

02 - AS Quiz  
  
As a quick quiz, which of the following types of routing protocols are responsible for routing within an autonomous system?

03 - AS Solution  
  
Intradomain routing protocols are responsible for routing within an autonomous system. Interdomain routing protocols, on the other hand, are responsible for routing traffic between autonomous systems.

04 - Intra-AS Topology  
  
Before we jump into intra-domain routing, let's take a look at what a topology might look like inside a single autonomous system. A topology inside an AS consists of nodes and edges that connect them. The nodes are sometimes called points of presence, or PoPs. A PoP is typically located in a dense population center, so that it can be close to the PoPs of other providers for easier interconnection. And, also close to other customers for cheaper backhaul to customers that may be pruchasing connectivity from this particular AS. The edges between pops are typically constrained by the location of fiber paths. Which for the sake of convenience. Typically parallel major transportation routes such as railroads and highways. Here's an example of a single AS topology which is the Abilene Network, which is a research network in the United States. Each of these locations would be considered a Pop, and each of these PoPs may have one or more edges between them. Georgia Tech is an autonomous system that connects at the Atlanta PoP of the Abilene Network. Here's a close up of the Abilene Network in the south eastern U.S.. The Abilene network connects to other universities in the southeast near Atalanta and an internet exchange point called SOX or southern crossroads. Now thus far we've just talked about the topology of an autonomous system which essentially defines the graph. The next step is to compute paths over that topology. A process called routing. Routing is the process by which nodes discover where to forward traffic so that it reaches a certain node. There are two types of intra-domain routing. One is called distance vector, and the other is called link state. In the rest of this lesson we'll explore the two different types of intra-domain routing and the advantages and disadvantages of each of them. Let's first take a look at distance vector routing.

05 - Distance-Vector Routing  
  
In distance vector routing, each node sends multiple distance vectors to each of its neighbors, essentially amounting to copies of its own routing table. Routers then compute costs to each destination in the topology, based on shortest available path. Distance vector routing protocols are based on the Bellman-Ford algorithm. A node x's forwarding table is based on the solution to the following equation. Suppose that node x is trying to find a shortest cost route to node y. In this case note X is trying to find a path through some intermediate node, V, that minimizes the cost between X and V, and the already known shortest cost path between V and Y. Again the solution to this equation for all destinations, Y, in the topology is X's forwarding table. Let's now take a look at distance vector routing by way of example.

06 - Example of Distance Vector Routing 1  
  
Let's suppose that we have a three node network with the costs on the edges as shown. Initially, each node has a single distance vector representing the shortest path cost to each other incident node in the graph. For example, the distance between x and x is obviously zero. And the shortest known distance between x and y, from x's perspective is one, the direct path. Similarly, the shortest known distance between x and z to x at the outset is five because all it knows is the direct path. Note that a shorter path between x and z exists via y, but x simply doesn't know about it yet. Now in distance vector routing, every node send its vectors to every other adjacent node. And each node then updates its routing table according to the Bellman-Ford equation. Let's look at what happens when node x learns of y's distance vectors. Well in this case, the distance from x to z will be computed as the minimum of the sums of all distances to z through any intermediate node. So the cost between x and y is one, and the distance between y and z as discovered by y's distance vector is two. Therefore, x can update its shortest cost distance to z as three. Similarly, x will receive a distance vector from z, five two zero, but of course, when it uses the Bellman-Ford equation to update its distances, again the distance between z and x will be updated from five to three. We can repeat this exercise at other nodes, as they receive distance vectors from other nodes in the topology. And quickly, every node in the network has a complete routing table. Now when costs decrease, the network converges quickly but one problem is that when failures occurs, bad news can actually travel slowly.

07 - Example of distance Vector Routing 2  
  
Let's look at a different example. So for the sake of illustration, I've increased the cost between x and z to 50, and now everyone starts with a different set of initial distance vectors. Now eventually, after running the distance vector protocol, we would see the tables converge as such. Let's suppose that the cost of the link between x and y suddenly increased from 1 to 60. Well now in this case, y would need to update Its view of the shortest path between Y and X. Now its no longer one, but its not 60 either. To see why let's go back to our Bellman-Ford equation. We can see that Y thinks it can get to Z with a cost of two, and that Z can get to X with a cost of three. So in fact it's going to update this entry from one to five. Then it will tell it's neighbor z it's new distance vector. In other words, that now its distance to x is no longer one but five. At this point, z needs to re-compute it's shortest path to x. Now, it knows that it can get to y with a cost of two but it thinks still that y can get to x with a cost of five. Therefore, this entry is no longer three but seven. And now z sends it's new distance vector back to y. Y then updates it's distance vector for z and this process continues. So then, y thinks it is now nine units away from x. So z has to do this all over again and now z thinks that its shortest path is two plus nine or 11. And this process repeats, of course, until z finally realizes that it has a shorter path of 50 directly through x. After this counting up process exceeds the value of 50. This problem is called the count to infinity problem, and the solution is called poison reverse. The idea here is that if y must route through z to get to x in its table, as it did here, then y advertizes an infinite cost for the destination x to z. So instead of sending five, zero, two, y would send infinity, zero, two. This would thus prevent z from routing back through y and Immediately, it would choose, the shortest path to x, of path cost 50.

08 - Routing Information Protocol  
  
An example of a distance vector routing protocol is the routing information protocol or RIP. The first version of RIP was defined in 1982 where edges had unit cost and infinity. Where the count to infinity problem was 16. Table refreshes occur every 30 seconds and when an entry changes, it sends a copy of that update to all of it's neighbors except for the one that induced the update. This rule is sometimes called the split horizon rule. The small value for infinity ensures that the count to infinity doesn't take very long and every round has a time out limit of 180 seconds. Which is basically reached, when a router hasn't received an update from a next hop for six 30 second periods. In practice, when a router or link fails in RIP, things can often take minutes to stabilize. So because of problems such as slow convergence and count to infinity, protocol designers, look to, other alternatives.

09 - Link State Routing  
  
The prevailing alternative and the one that is used in most operational networks today is link state routing. In link state routing, each node distributes a network map to every other node in the network. And then, each node performs a shortest path computation between itself and all other nodes in the network. So initially, each node adds the cost of its immediate neighbors, D(v), and every other distance to a node that is infinite. Then each node floods the cost between nodes u and v to all of its neighbors. And the distance to any node v becomes the minimum of the cost between u and w plus the cost to w. Or the current shortest path to v. The shortest path computation is often called the dijkstra, shortest path ROUTING ALGORITHM. Two common link state routing protocol are open shortest paths first or OSPF and intermediate system - intermediate system or IS-IS. In recent years, IS-IS has gained increasing use in large internet service providers and is the more commonly used. Link state routing protocol in large transit networks today. One problem with link state routing is scale. The complexity of a link state routing protocol grows as n cubed where n is the number of nodes in the network.

10 - Coping with Scale Hierarchy  
  
One way of coping with scale, is to introduce hierarchy. OSPF has a notion of areas, and ISIS has an analogous notion of levels. In a backbone network, the network's routers may be divided into levels, or areas. And the backbone, itself, may have its own area. In OSPF, the backbone area is called area zero, and each area in the backbone that's not in area zero has an area zero router. The area zero routers perform shortest path computation. And the routers in each of the other areas independently perform shortest path computation. Now paths are computed by computing the shortest path within an area. Or, if the path must leave an area, it's computed by stitching together the shortest path to the area zero backbone router, and then the shortest path across area zero, followed by another intra-area shortest path.

11 - Interdomain Routing  
  
We're now moving on to cover interdomain routing or routing between ASes. Recall that internet routing consists of routing between tens of thousands of independently operated networks, or autonomous systems. Each of these networks operate in their own self-interest, and have independent economic and performance objectives, and yet they must cooperate to provide global connectivity so that when you're sitting at home, you can retrieve content that might be hosted at the Georgia Tech network. Now, each independently operated network is called an autonomous system, or AS. And each AS advertises reachability to some destination by sending what are called route advertisements or announcements. The protocol that ASes use to exchange these route advertisements is called the Border Gateway Protocol, or simply, BGP. A route advertisement has many important attributes, but for now, let's just talk about three. Now a router here, let's say on the Comcast network, might receive a route advertisement, typically from its neighboring AS. That route advertisement might contain a destination prefix, such as the IP prefix for Georgia Tech. Then it might contain what's called a next hop IP address, which is the IP address of the router that Comcast router must send traffic to, to send traffic along that route. Typically that next hop IP address is the IP address for the first router in the neighboring network. And Comcast router knows how to reach that next hop IP address because its border router and the border router in the neighboring AS are on the same subnet. Typically this might be a /30 subnet, therefore this IP address is reachable from Comcast's border. A third important attribute is whats called the AS path which is a sequence of what are called AS numbers, that describe the route to the destination. Now strictly speaking, the AS path is nothing more than the sequence of ASes that the route traversed to reach the recipient AS. So for example, Georgia Tech's AS number is 2637 and Abilene's is 10578. So the AS path that Comcast would hear if it received a route advertisement from Abilene for Georgia Tech, would be 10578 followed by 2637. So in the remainder of the lesson we'll look at other BGP route attributes. But these are essentially the three most important because they describe how to stitch together an interdomain path to a global destination. So we have the destination IP prefix for the destination that a router might want to send traffic to. The next hop, which is the IP address for the router for the next hop along the path. And finally, the AS path, which is the sequence of ASes that the route traversed en route to the AS that's hearing the announcement. The last AS number on the AS path is often called the origin AS, because that is the AS that originated the advertisement for this IP prefix. In this case, the origin AS is 2637, or Georgia Tech, because it is the AS that originated the announcement for this prefix.

12 - Interdomain Routing 2  
  
Now thus far, we've talked about interdomain routing BGP, or the border gateway protocol, as consisting of route advertisements solely between border routers of adjacent autonomous systems. In fact, this is a specific type of BGP called external BGP, or EBGP. But in fact, as we know, each one of these autonomous systems has routers of its own, inside. Those routers also need to learn, routes to external destinations. The protocol that is used to transmit routes inside an autonomous system for external destinations, it's called internal BGP or IBGP. Okay, so to review, external BGP is responsible for transmitting routing information between border routers of adjacent ASs about external destinations. And internal BGP is responsible for disseminating BGP route advertisements about external destinations to routers inside any particular AS. Note that the distinction between IGP and an intra-domain routing protocol or an IGP.

13 - IGP vs iBGP  
  
The IGP or the intra-domain routing protocol, disseminates routes inside an AS to internal destinations. Whereas iGBP or internal-gateway border protocol, disseminates routes inside an AS to external destinations. So let's suppose that a router inside AS-A is trying to reach a destination inside AS-B. ASA would learn the route by EBGP. And the next topic of course, at this router, would be the border router at B. And now a router inside autonomous system A would learn the route to B via IBGP. Then the BGP next stop, would be the border router. And so, this router inside ASA, needs to use the IGP, to reach the iGBP, next hop.

14 - Protocol Quiz  
  
So as a quick quiz, which routing protocol is responsible for disseminating routes inside NAS to external destinations? Is it the IGP? Is it iBGP. Or is it eBGP?

15 - Protocol Solution  
  
IBGP is responsible for disseminating routes inside NAS about destination IP prefixes that are located outside that AS. The iBGP next hop is typically a next hop IP address that is reachable via the AS's intradomain routing protocol, or IGP.

16 - BGP Route Selection  
  
Let's now take a quick look at BGP route selection. It is often the case that a router on a particular autonomous system might learn multiple routes to the same destination. In this case, a router on autonomous system one, might learn a route to a destination in AS4 via both AS2 and AS3. In this situation. The router in AS one must select a single best route to the destination among the choices. The selection among multiple alternatives is known as the BGP route selection process. Let's now take a quick look at that process.

17 - BGP Route Selection Process  
  
The first step in the BGP route selection process is to prefer a route with the higher local preference value. The local preference value is simply a numerical value that a network operator in the local AS can assign to a particular route. This attribute is purely local, it does not get transmitted between autonomous systems, so it is dropped in EBGP route advertisements. But it allows a local network operator the ability to explicitly state that one route should be preferred over the other. Among routes with equally high local preference values, BGP prefers routes with shorter AS path length. The idea is that a path might be better if it traverses a fewer number of autonomous systems. The third step involves comparison of multiple routes advertised from the same autonomous system. The multi-exit discriminator value allows one AS to specify that one exit point in the network is more preferred than another. So lower MED values are preferred, but this step only applies to compare routes that are advertised from the same autonomous system. Because the neighboring AS sets the MED value on routes that it advertises to a neighbor, MED values are not inherently comparable across routes advertised from different ASs. Therefore this step only applies to routes advertised from the same AS. Fourth, BGP speaking routers inside an autonomous system will prefer a BGP route with a shorter IGP path cost to the IGP next up. The idea here is that if a router inside an autonomous system learns two routes via IBGP then it wants to prefer the one that results in the shortest path the to the exit of the network. This behavior results in what is called hot potato routing, where an autonomous system sends traffic to the neigjboring autonomous system via a path that traverses as little of its own network as possible. Finally, if there are multiple routes with the highest possible local preference, the shortest AS path and the shortest IGP path, the router uses a tiebreak to pick a single breaking route. This tiebreaking step is arbitrary. It might be the most stable, or the route that's been advertised the longest. But often, to induce determinism, operators typically prefer that this tie breaking step is performed based on the route advertisement from the router with the lowest router ID, which is typically the neighboring router's IP address. Let's now take a closer look into local preference, AS path length, muli-exit discriminator and hot potato routing. Now as I mentioned the first step in the router selection process is for routers to prefer routes with higher local preference values. Now an operator can actually set the local preference value on incoming BGP route advertisements to affect which route a router ultimately selects. Let's see how this works.

18 - Local Preference  
  
Now, a router in AS1 might learn two routes to a destination, one via the AS path 24 and the other via the AS path 34. Local preference or simply local pref allows an operator to configure the router to assign different preference values to each of the routes that it learns. The default local preference value Is 100. If the operator prefers that this router select the path through AS two, it can configure the router to set a higher local preference for that route. Such as 110. This results in this router selecting the route through AS two and sending traffic to the destination in AS four via AS two. In this way a operator can adjust local preference value of incoming route to control put bound traffic or to control how traffic leaves it tominac system in route to a destination this is extremely useful in configuring primary and back up routes for example here the route though A.S two might be the primary route,and the route through AS three, is the backup route. Now typically as I mentioned, local preference is used to control outbound traffic. But sometimes autonomous systems can attach what's called a BGP community to a route to effect how a neighboring autonmous system sets local preference. A community is nothing more, but a fancy jargon word for a tag on a route. So let's suppose that AS four wanted to control inbound traffic by affecting how AS two or AS three set local preference. In this case, let's suppose, that AS two wanted traffic to arrive via AS three, its primary, rather than by AS two, its backup. In this case, AS two might advertise its BGP routes with primary and backup communities. The backup community value might cause a router in AS two to adjust its local preference value, thus affecting how AS two's outbound traffic choices are made. So, again local preference is used to control outbound traffic, in this case AS two's outbound traffic decision, but the use of a BGP community on the route advertisement can sometimes be used to cause a neighboring AS to make different choices regarding it's outbound traffic. Thereby, allowing an AS to specify a primary or back up path for incoming traffic. This type of arrangement requires prior agreement.

19 - Multiple Exit Discriminator  
  
Let's suppose that two autonomous systems connect in two different cities, San Francisco and New York. Let's further suppose that AS1 wants traffic to destination d to enter via New York City, rather than via the peering link in San Francisco. Well remember that all things being equal, routers inside AS2 will select the BGP route with the shortest IGP path cost to the next hop, resulting in hot potato routing. So some routers will select the San Francisco egress, and other routers might select the New York egress. To override this default hot potato routing behavior, AS1 might advertise its BGP routes to AS2 with MED values. For example, if the MED value on the route learned, at the border router in New York, was 10, and the MED value from the route learned from the router in San Francisco was 20, than instead of performing hot potato routing, all of these routers that would ordinarily be closer to the San Francisco egress, would instead pick the route learned via the New York egress. Because the preference for a lower MED value comes before the preference for a next hop with the lower IGP path process. So all of these routes would instead be carried over AS2's backbone network and exit via New York. Thus MED overrides hot potato routing behavior allowing NAS to explicitly specify that it wants another neighboring AS to carry the traffic on its own backbone network, rather than dumping the traffic at the closest egress and forcing traffic across the neighbor's backbone. MEDs are typically not used in conventional business relationships, but they're sometimes used, for example, if AS1 does not want AS2 free riding on AS1's backbone network. So effectively MED allows AS1 to say, yes, I will connect or peer with you, but it is your job to carry the traffic long distances across the country. This mechanism is sometimes used when a transit provider peers with a content provider, and the transit provider doesn't want the content provider essentially getting free transit through the neighboring AS. In the absence of MED overriding any behavior, typically what will happen is a router inside the AS2 would learn multiple routes via internal BGP to different egress points for the same destination d, and it would simply pick the next hop or, the egress router, with the lowest IGP path cost. In this case, 5. It's very common practice to set these IGP costs in accordance with distance, or propagation delay, thus resulting in routers inside the AS picking shorter paths. Now one problem with this notion of hot potato routing is that a very small change in IGP path cost can result in a lot of BGP routing changes. Remember that it's probably not just one destination that's being routed through the San Francisco egress, but maybe tens of thousands of routes. So a single IGP path cost change can result in rerouting of tens of thousands of IP prefixes in BGP. People have looked at various ways to improve the stability of BGP routing, by decoupling the IGP and the BGP, in this part of the route selection process.

20 - Interdomain Routing Business Models  
  
So now we're going to look at Interdomain Routing Business Models. So the one thing to remember about interdomain routing is that it's really all about routing money. Let's consider this AS that wants to send traffic to a particular destination. Well, in the internet there are two different types of business relationships: a customer-provider business relationship where money flows from customer to provider, regardless of the direction that traffic flows, the other type of business relationship is a peering relationship where an AS can exchange traffic with another AS free of charge. This is sometimes also called settlement-free peering. So already you can see given three possible ways to reach the destination. This AS is first going to prefer a route through its customer, because regardless of the direction of traffic on this link, money is always flowing from the customer. The peering link is the second most preferable because it's free. And the least preferable route is through the provider, because the AS has to pay money every time it sends traffic on this link. This leads to the basic rules of preference in interdomain routing, where customer routes are preferred over peer routes, which are in turn preferred over provider routes. The other consideration that an AS has to make is filtering, or export decisions. In other words, given that an AS learns a route from its neighbor, to whom should it re-advertise that route? To understand filtering and export decisions, let's add a couple more AS's to the graph. Let's add another peer, and let's add another provider. Let's call this AS in the middle of the picture Cox Communications. This ISP might have smaller regional customers and it might also buy transit connectivity from other providers. Now let's suppose that this AS learns routes to a destination via its customer, its peer and its provider. Now we already have established that it would prefer the customer route, so that it can make money by sending traffic to that destination. But what about filtering decisions? Well, routes that are learned from a customer, Cox of course would want to re-advertise to everyone else, because the more people use that route, the more money Cox makes. Therefore a route that's learned from a customer, gets advertised to everybody else. On the other hand, a route that's learned from a provider, if it were actually selected, would of course, only be advertised to customers. It wouldn't make any sense to take a route like this and advertise it to another provider. The reason, of course, is that money is flowing in the direction of the providers. So any route that's learned from a provider would never be advertised to another provider, because it would result in Cox essentially becoming a transit provider between two of its own providers and paying them both for the privilege of carrying that traffic. So routes learned from a provider would only ever be advertised to other customers. And similarly, routes from peers would only be advertised to other customers, not to other peers or other providers. So to summarize, interdomain routing has both ranking rules, where, given multiple choices, an AS might prefer a customer route over a peer route over a provider route. And then, given that it selected a particular route from either a customer, a provider, or a peer, Ii makes different decisions about where to re-advertise that route to other neighboring ASs. Now as it turns out if every AS in the internet followed these rules exactly, then routing stability is guaranteed. Now you might wonder, isn't routing stability guaranteed already? And it turns out that it isn't.

21 - Interdomain Routing Can Oscillate  
  
In fact, interdomain routing can oscillate indefinitely. To see why, consider the following 4 AS topology, where each AS specifies preferred paths, presumably via local preference. So each AS prefers the AS in the clockwise direction, rather than the shorter, direct path. Now it's pretty easy to see that there's no stable solution. Let's suppose that we started off with everybody selecting the direct path. Well, in this case, any one of these ASs would notice that it has a more preferred path. So for example, AS1 would see that because AS3 has picked the direct path, then, in fact, it could prefer 1 3 0. Now we enter into a situation where oscillations can occur indefinitely. Similarly, here now AS3 sees that it has a more preferred path, 3 2 0, so it might switch to that. In doing so, it breaks AS1's path. 1 3 0 no longer works. So AS1 has to switch back to its less preferred direct path, but now we're in the same situation all over again. Because now AS2's preferred path becomes available via 1, so AS2 now reroutes, and AS3's most preferred path, 3 2 0, no longer works. So it must switch to the direct path. Now, it's very easy to see that this oscillation continues ad infinitum. This particular pathology was first discovered by Varadhan, Govindan, and Estrin, in a paper called persistent route oscillation in interdomain routing, in 1996. Later, Tim Griffin formalized this pathology and derived conditions for stability. Those stability conditions came to be known as a BGP correctness property called safety. It turns out, that if ASs follow the ranking and export rules that we discussed, that safety is guaranteed. But, there are various times when those rules are violated. Business relationships, such as regional peering, and paid peering can occasionally cause those conditions to be violated. So as it turns out, to this day, BGP is not guaranteed to be stable in practice, and many common practices result in the potential for this type of oscillation to occur.