Lecture 13

Gidon Rosalki

2026-01-18

Notice: If you find any mistakes, please open an issue at `https://github.com/robomarvin1501/notes_networking`

1 BGP details

Firstly, we will note that BGP takes place at the *application* layer. It takes place over TCP, on port 179. BGP connected routers establish the connection, exchange all active routes, and then exchange incremental updates while the connection is alive.
These incremental updates include **Announcement** updates, where upon selecting a new active route, the AS adds its own ASN to path and (optionally) advertises the path to each neighbour, and **Withdrawal** updates, where if the active route is no longer available, send a withdrawal message to the neighbours.

In order to avoid loops, whenever a node (router) observes a new path, it can search if it's already on the path, and if so, drop the new path. Also, note that BGP is neither a distance vector, or link state vector, since it is a path based protocol in the sense that we send the complete path, and not just the next hops. This is unlike link state, since each edge router **only** reports its routes to neighbour edge routers, and it is unlike distance vector, since the full paths are sent.

Let us consider these route update messages. They include the destination prefix (here denoted d), and the route attributes, including

- AS path (the route through the different ASes that it took to reach the recipient)
- The next hop IP address (this is the router from which the message left the current AS, such that the next AS may send messages back along the route)

2 BGP (In)Stability

There are some structures of ASes which can cause infinite oscillation, resulting in an unstable state of the network. This makes the network unpredictable, and hard to debug. It might lead to the network being constantly flooded with BGP updates, using up all the available bandwidth. This instability can also cause deteriorated instability.

We want to guarantee that BGP will always reach a stable state, and ideally this should happen quickly. This is called BGP **safety**. There are two practical mechanisms for enabling this:

1. Penalising "flaps" (route fluctuations)
2. Enforcing a minimum time interval between BGP updates (to the same neighbour).

These are only occasionally used, since they are only occasionally effective.

We will note that the internet is generally fairly stable. BGP routes to popular destinations tend to be the same for around 10 days, and most traffic in the internet travels along stable routes. Part of the reasoning here is that there are no customer, provider cycles. ASes prefer to use their customers for routing, than their providers, since that way they make money. Additionally, ASes do not announce their provider learned routes to other providers, if only from a money making standpoint.

2.1 Gao-Rexford conditions

If the following 3 conditions hold, then the BGP network is stable. We can guarantee BGP safety in a network if **all** the following conditions hold:

1. **Topology condition**: No customer-provider cycles in the AS graph
2. **Preference condition**: Prefer customer learned routes over provider, and peer learned routes (go by preference through paying customers, than through peers)
3. **Export condition**: Prover and peer learned routes are exported *only* to customers

So, to summarise BGP, it enables network operators great expressiveness at the potential cost of persistent route oscillations. Protocol divergence is bad for networks, since the network becomes unpredictable, and there is performance degradation. Gao-Rexford conditions imply BGP safety, which is a partial explanation for the Internet's relative stability.

3 Inter vs Intra

An AS is **not** a single node, but rather is comprised of multiple routers. Thus, we need to distribute BGP information within the AS. In order to achieve this, Internal BGP (iBGP) sessions are held between the routers of a single AS.

To make this happen, we need to join together information from iBGP, and IGP (Interior Gateway Protocol). iBGP announces reachability to *external* destinations, and maps a destination prefix to an egress point (we can reach 128.112.0.0/16 from 192.0.2.1), and IGP is used to compute paths *within* an AS, and masp an egress point to an outgoing link (we may reach the external IP 192.0.2.1 via the AS internal IP 10.1.1.1).
IGP is shortest path routing with static link weights. It computes the shortest paths to other routers based on the link weights, in order to determine the next hop to every other router. The link weights are configured by the network administrator.

We will note that since an AS has many edge routers, it may have numerous connections to a neighbouring AS that are equivalent. This means that multiple border routers may learn good routes with the same LocalPref and AS-path length. We are thus left with the question of how the internal routers know which egress point to use. To resolve this we use *hot potato routing.* Each router selects the *closest* egress point based on IGP link weights. So, to put it all together, a route from a router within the AS, to another AS is chosen by:

1. Highest **local preference**
2. **Shortest** AS path
3. **Closest** egress point
4. **Arbitrary** tie break

3.1 Summary

ASes are networks of routers, which combine together iBGP, and IGP routing. iBGP announces reachability to *external* destinations, and IGP computes routes *within* the AS.

4 BGP security

BGP is built on trust, and is thus exceedingly vulnerable to bad actors. Every few years, there is a serious BGP based attack that makes the news.

4.1 Session security

BGP sessions run over TCP. There is a TCP connection between neighbouring BGP routers, and they send their BGP messages over this TCP connection. This makes BGP vulnerable to attacks on TCP. One could eavesdrop by tapping the link to infer routing policies, tamper with the packets that are traversing the link, by dropping, modifying, or adding packets, thus changing, filtering, or replaying BGP routes. One may also reset the session, and congest the link.

Fortunately, one may quite easily defend a BGP session. Two end points can use known IP addresses, and ports to communicate, and agree to authenticate and encrypt their messages. Additionally, there is often very limited physical access to a link, since for most of these, one needs direct physical access, and the link is often located in a (somewhat) secured building.

4.2 Manipulating BGP

BGP is eminently vulnerable to manipulation. In 2008, Pakistan ordered their ISPs to block YouTube. To do this, their ASes started publishing that YouTube is at a different location, which led nowhere. This leaked (accidentally) to the wider internet, and brought YouTube down for a few hours. This happens easily since there is no proof that it is actually true that the person claiming to own an IP block *actually* owns that IP block, so BGP simply believes that whoever claims to own it, actually does own it.
This is called Prefix Hijacking, where two ASes are competing as to who owns a prefix. The hijacking AS needs a router with BGP sessions that is configured to originate the prefix. This could happen because a network operator makes a mistake, a network operator launches an attack, or an outsider breaks into the router, and reconfigures it.

The targeted AS may not even notice that prefix hijacking is going on. If targeted AS tries to snoop, and check, they may only experience degradation, and not the full hijack. The best way to diagnose it is to analyse updates from *many* different vantage points, by launching traceroute from many different vantage points.

4.3 Origin Authentication

To defend against prefix hijacks, there are many best common practices, such as filtering routes based off their prefix (so for example, stubs should only announce their own IP prefixes), and packet filters to avoid unwanted traffic. This is not good enough, it depends on vigilant application of BCPs, and does not handle misconfiguration (malicious, or otherwise). It also does not address the fundamental question of who own what IP block.

The proposed solution to this is called **Origin Authentication**. Here, we use a secure database that maps IP prefixes to owner ASes. This is imperfect, since a bad actor can convince others that they have a nonexistant link, which is shorter than the standard link from it to some IP prefix. This attracts both the sources that want a shortest link, and the sources that are trying to avoid the ASes that the bad actor has cut out of its route.