Tutorial 11 - BGP

Gidon Rosalki

2026-01-15

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

1 Introduction

So far, the routing protocols that we have discussed generates forwarding tables, either using Distance Vector, or Link State. This is however, only aimed at small networks, and not appropriate for the wider internet. We need to extend this to the wider internet, which allows for different interactions between different networks, and operators, that may be unwilling to work with each other, or perhaps have different requirements over how the packets should be sent. Effectively, we have added *politics* into the picture, which complicates everything.

2 Autonomous Systems

The internet is comprised of multiple networks, and each network has its own routing policy, financial goals, and administrative unit (Google, AT&T, HUJI, Bezeq, etc,). We will call each of these networks an *Autonomous System*, or AS for short. Each AS is identified by a unique number (ASN).

ASes do not necessarily differentiate things geographically, just topologically. Packets will be sent between them according to their preferences. Google may want to send packets in the cheapest manner possible, another AS may be looking to provide its customers with the best service, and so look for the shortest route, and two providers may decide that they do not care how it affects the service / price, as long as they do not talk to each other.

Note that these different goals mean that routing will **not** necessarily be the shortest path. We do not even know the routes that packets will take *within* a different AS. As a result, a different form of routing, that incorporates these types of demands, is needed. Since these entities often want to make money, and thus *enhance shareholder value*, we need to take the financial relationship between them into account.
ASes sign bilateral, long term contracts between them, that concern themselves with how much traffic they carry which destinations can be reached from them, and how much money changes hands in order to achieve this. Neighbouring pairs of ASes typically either have customer provider relationships, or peering relationships:

- Customer/Provider: One AS pays another for reachability to some set of destinations
- Peering: Two ASes exchange traffic for free, typically only for one or two of their customers

The relationship may either be "full", where the customer receives routes for all Internet destinations from its traffic provider, or "partial", where the customer receives routes for some subset of all Internet endpoints.

Peering is motivated by increased redundancy, routing control, traffic management, and predictability of traffic. It also reduces latency, congestion, costs, and more.

There are 3 types of ASes:

- Tier 1 (There are roughly 10): This is the topmost level, with no providers. All ASes are fully connected to each other
- Tier 2 (Roughly 1000): Needs at least one Tier-1 provider, and provides transit service for the customers, generally on either a regional, or national scope
- Stub (Roughly 85% of all ASes, eg HUJI): Connects to one or more providers, and does not provide transit services

3 BGP

Recall that IP addresses are 4 byte numbers (32 bits), presented as ip/k: The first k most significant bits are fixed. For example, on 240.161.192.0/18, the first 18 bits are fixed for this subnet, and the remaining 14 are the available addresses.

BGP (Border Gateway Protocol) is an extension of Distance Vector routing, which supports flexible routing policies. It's key idea is to advertise the *entire* path to an IP prefix. So, when AS 2 builds a link to AS 1, AS 2 will announce the path (2, 1), as opposed to AS 1 which only announced path (1). This was originally intended for loop detection purposes.

As we have already seen extensively in this course, BGP is a distributed protocol, meaning that every node / AS runs its computation independently. Every AS announces the IP prefixes that may be reached from it, to its neighbours, and they utilise these data, along with their existing routes, in order to update the routes that they will use. The appropriate data will then be sent to update their neighbours, and so on.
Every AS chooses routes that are based on its local routing policies. The routes to the destination are built hop by hop, and usually, an AS announces only its **chosen** routes, rather than **all** the routes of which it is aware. Note, different routes for incoming, and outgoing traffic is possible.

The BGP protocol follows the following steps:

1. Receive the BGP routes from neighbours
2. Apply the import policy (remove unwanted routes, maybe too expensive, there is a preference for AS n, and so on)
3. Select the "best" route according to the local policies
4. Apply the export policy (filter out the routes that you do not want to announce to your neighbours)
5. Transmit the chosen BGP route to neighbours

BGP occurs 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.

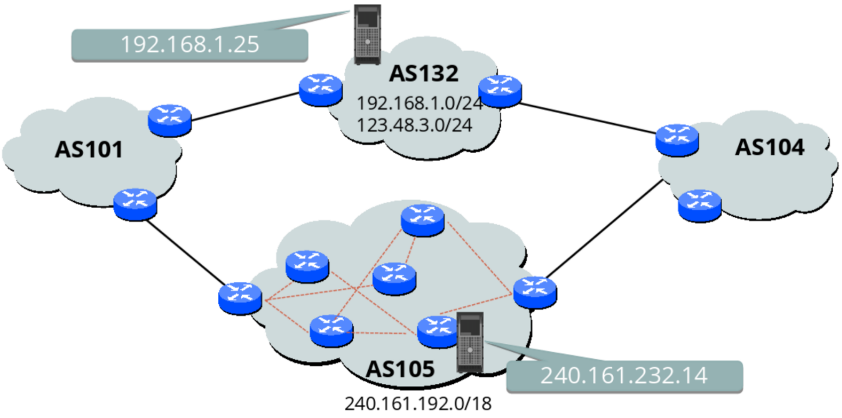


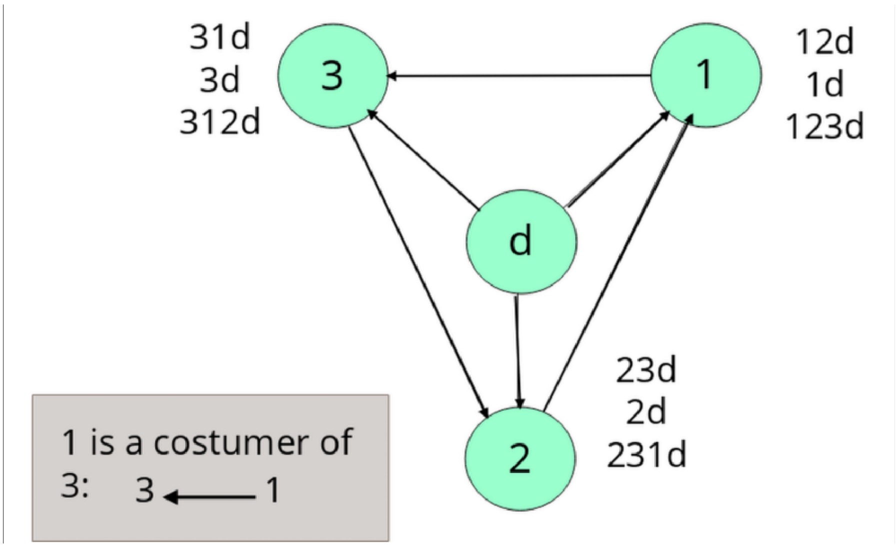
Figure 1: Routing between ASes

How will we route from 240.161.232.14 to 192.168.1.25?
First, we need to reach one of the edge routers. These are the BGP routers which are connected to other ASes' edge routers. This is followed by Internal Border Gateway Protocol (iBGP). This is a routing protocol for routers within the same AS, which announces reachability to external destinations. Finally, Interior Gateway Protocol (IGP) is used to compute paths within the AS (e.g., distance vector or link-state based protocols). In each router (edge/internal), it maps an egress point (edge router) to an outgoing link (next hop towards the egress point).

4 BGP Stability

The customers can have multiple preferences. However, this does not mean that the routes are static, there can be a hierarchy of preferences which change the selected routes, resulting in the paths potentially changing over time. This can happen when preferences change, when previously existing paths are no longer available, when new paths are available, and so on.

A stable BGP state is a state when no AS wants to change its routes. We will note that BGP is **not** always stable. Consider:



Assuming that every AS announces its chosen route to its neighbours, does a stable state exist in this network? Let us assume towards contradiction that there is. AS(1) has 3 possible routes (12d), (1d), (123d). Note that the orders of available routes are in order of preference. If AS(1) chooses (12d), then AS(2) **must** choose (2d). AS(3) cannot choose (31d), based off AS(1)'s choice, so it chooses (3d). However, in this case, AS(2) thus switches to the route (23d), and so AS(1)'s choice is no longer available.
Let us assume instead that AS(1) chooses (1d). So, AS(3) chooses (31d). Since for AS(2), the route (23d) does not exist, it chooses (2d), but then AS(1) will switch to its preferred route (12d).
Finally, should AS(1) choose (123d), then AS(2) has by definition chosen (23d), and AS(3) has chosen (3d), but then once AS(1) hears of the route (1d), it will switch to it, as it is AS(1)'s preferred route.

4.1 BGP Safety

Given a network running BGP, we say that it is **BGP Safe** if it (eventually) converges to a single, stable, BGP state, no matter from which state it started, or the message propagation time. We do not really know (yet) when a BGP network will be safe, but we do have the following theorem:

Theorem 1. *The existence of more than one stable BGP states in a network implies that it is* ***not*** *BGP safe*

In order to find stable states, we *could* try every state, but there are simpler methods.

4.1.1 Gao-Rexford Conditions

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

We will note that in the above example, the topology condition did not hold, AS(1) is a customer of AS(2), who is a customer of AS(3), who is a customer of AS(1). We can change this, by swapping the arrow from AS(1) to AS(3), to be from AS(3) to AS(1). We will note that now the preference condition does not hold, since AS(3) has a preference for sending to its provider AS(1), rather than its customer d. This may be fixed by swapping AS(3)'s preferences to be (3d), (31d), (312d).

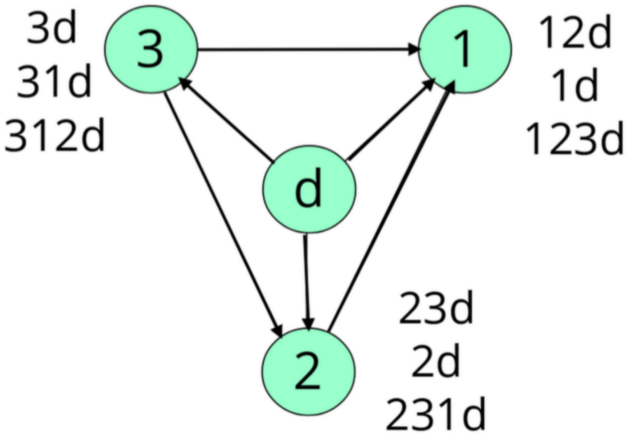


Figure 2: Stabilised version

AS(3) will choose the path (3d), and announce it to AS(1), and AS(2), since it is a customer learned path. AS(2) will choose the path (23d), ad announce it to AS(1), since it is a customer learned path. Finally, AS(1) does not know the path (12d), since it was not announced by AS(2), and so it will choose the path (1d). Everyone has their preference of paths, and so the network has stabilised.

There are not too many questions to be asked on this subject, they generally follow the format of “is this network stable”, and if it is not stable “change this network to be stable”.