Computer Networks - Lecture 17 & 18

- Author: Ruben Schenk
- Date: 10.06.2021
- Contact: ruben.schenk@inf.ethz.ch

5.6 Border Gateway Protocol (BGP)

So far, we have looked at `intra-domain routing protocols` like distance vector and link state algorithms. These work fine within, e.g., an autonomous system, but as soon as a network gets too big, they quickly become infeasible:

- Distance vector protocols converge slowly, so for a network as big as the internet, convergence would never happen.
- Link state algorithms need the whole network topology, which is impossible for the internet.

Since the internet is a network of networks, referred to as an `autonomous system (AS)`, we use special `inter-domain routing protocols` like `BGP` to connect the autonomous systems. By using BGP, autonomous systems exchange information about IP prefixes that they can reach. The protocol needs to solve three key challenges:

- *Scalability*: The number of networks and prefixes is huge.
- *Privacy*: Networks don't want to expose internal topologies.
- *Policy Enforcement*: The network needs to control where to send and receive traffic in the absence of an internet-wide link-cost metric.

BGP relies on `path-vector routing`, similar to distance-vector routing, but with the key idea, that we advertise an *entire AS-level path* instead of distances.

5.6.1 BGP Policies

Two ASes connect only if they have a business relationship. We distinguish between two types of relationships:

- Customer-Provider Relationship:
  - In a customer-provider relationship, a customer pays a provider to get internet connectivity globally, e.g. Swisscom is a customer of Deutsche Telekom. The amount the customer pays is based on the peak usage.
- Peer-Peer Relationship
  - In a peer-peer relationship, peers don't pay each other for connectivity bur rather connect out of common interests, mainly since they exchange a large amount if traffic, e.g. Deutsche Telekom and AT&T.

Policy Rules

BGP obeys the following policy rules:

- Providers transit traffic for their customers.
- Peers to not transit traffic between each other.
- Customers do not transit traffic between their providers.

Selection and Export

In `selection`, we must decide which path to use for *outbound* traffic. The general rule is to prefer routes coming from customers over peers over providers.

In `export` we must decide which path to use for *inbound* traffic. Routes coming from customers are propagated to everyone else, that is, to peers and providers, where as routes coming from peers and providers are only propagated to customers.
Note that this requires Tier-1's to be connected through a *full-mesh of peer links*, otherwise the Internet would be partitioned.

5.6.2 BGP Protocol

There are two different types of BGP sessions:

- `external BGP (eBGP)`: Those sessions connect border routers in different ASes and are used to learn routes to external destinations.
- `internal BGP (iBGP)`: Those sessions connect the routers in the same AS and are used to disseminate externally-learned routes internally.

BGP as a protocol is rather simple and consists of four basic types of messages:

- `OPEN`: Establish TCP-based BGP session.
- `NOTIFICATION`: Report unusual conditions.
- `UPDATE`: Inform neighbor of $a)$ a new best route, $b)$ a change in the best route, or $c)$ the removal of the best route.
- `KEEPALIVE`: Inform a neighbor that the connection is alive.

BGP Updates

A `BGP update` message carries an IP prefix together with a set of attributes that describe route properties that can be used in route selection/export decisions. Such attributes can either be local (only seen in iBGP sessions) or global (seen on both iBGP and eBGP sessions). Possible attributes are:

- `NEXT-HOP`: A global attribute which indicates where to send traffic next. It identifies the egress point and is set when a route enters an AS and does *not* change within the AS.
- `AS-PATH`: A global attribute that lists all ASes a route has traversed (in reverse order).
- `LOCAL-PREF`: A local attribute set at the border, it represent how "preferred" a route is. A higher value results in all routers using this route to reach any external prefixes, even if they are closer to another egress point.
- `MED (Multi-Exit Discriminator)`: A global attribute which encodes the relative "proximity" of a prefix with respect to the announcer. In contrast to `LOCAL-PREF`, a lower `MED` indicates closeness and is preferred over a higher value.

BGP Decisions

Given the set of all acceptable routes for each prefix, the `BGP decision process` elects a single route. BGP thus is a single path protocol.

> `Route picking` in BGP works as follows: Out of all possible routes, BGP decision processing picks exactly one with the following precedence:

1. Highest *LOCAL-PREF*.
2. Shortest *AS-PATH* length.
3. Lower *MED*.
4. On a tie of the first three attributes, we pick the routes that were learned via eBGP over routes that were learned internally via iBGP:
    - A lower IGP metric to the next hop
    - A smaller egress IP address (tie-breaker)

5.6.3 Problems with BGP

There are several problems with BGP:

- Reachability: BGP does not guarantee reachability even if a graph is connected.
- Security: Simply absent. AS can advertise any prefix, AS can arbitrarily modify content og an AS-PATH, and can forward traffic along different paths than the advertised one.
- Convergence: With arbitrary policies, the protocol might fail to converge tue to policy oscillations.
- Performance: Path selection happens for economic reasons, not based on performance.
- Anomalies: BGP is bloated and underspecified at the same time such that there are conflicting interpretations.
- Relevance: BGP policies are rapidly changing.

If all ASes policies follow the customer/peer/provider rules, also called Gao-Rexford rules, then BGP is guaranteed to `converge`.

6. Link Layer

The `link layer` is concerned with transferring messages over one or more connected links, thus providing a service to the network layer by building on top of the physical layer.
We'll refer to any device that runs the link layer as a `node` and to the communication channels that connect adjacent nodes as `links`. Over a given link, a transmitting node encapsulates a *datagram* in a `link-layer frame`.

6.1 Framing

The physical layer gives us a stream of bits. But how do we interprete it as a sequence of frames?

The job of the link layer is to interprete that bitstream as a sequence of frames.

6.1.1 Byte Count

The first idea is to use a `byte count`. In this approach we start each frame with a length field that denotes the size of the frame in bytes.

The problem with this approach is that once a framing error is made, there is no way to recover from it, and all the subsequent frames are decoded incorrectly.

6.1.2 Byte Stuffing

A better idea is `byte stuffing`. The approach is to have a special `FLAG` byte that denotes the start and end of a frame and a special `ESC` byte.

- If `FLAG` appears in the data, replace it with `ESC FLAG`.
- If `ESC` appears in the data, replace it with `ESC ESC`.

This way, any unescaped (ESC) `FLAG` is a start/end of a frame.

6.1.3 Bit Stuffing

We can also stuff at the bit level:

- We call a FLAG six consecutive 1s
- On transmit, after five 1s in the data, insert a 0
- On receive, a α after five 1s is deleted