

# BGP Update Packets



- BGP peers initially exchange their full BGP routing tables using Update messages.
- Incremental updates are sent only if there is a topology change, if policy towards a peer changes, or if a router requests an update from a peer (eg Route Refresh).
- Multiple paths require multiple Update messages. All the attributes in an Update message refer to that path and the networks that can be reached through it.
- Updates propagate Network Layer Reachability Information (NLRI) which includes the network prefix, prefix length, and Path Attributes
- Withdrawn routes are also advertised through Updates

# BGP Path Attributes



- Path Attributes contain information used to calculate best paths to the networks
- There are Well-Known and Optional Attributes

# Well-Known Attributes



- Well-Known Attributes are part of the BGP standards and required to be recognized by all BGP routers
- There are two categories of Well-Known Attributes:
  - **Well-Known Mandatory:** Must be present for every route. Propagated to other BGP routers.
  - **Well-Known Discretionary:** Optionally present. Used when configured by an administrator. Propagated to other BGP routers.

# Optional Attributes



- Optional attributes are defined in private vendor extensions or later standard extensions which were not included in the original BGP standards.
- They are not required to be recognized by a vendor's BGP implementation
- There are two categories of Optional Attributes:
  - **Optional Transitive:** If the attribute is not recognised:
    - Router will include it in the update when passing the route to neighbours.
    - The 'Partial' bit is set to indicate the router did not recognise the attribute.
  - If the attribute is recognised, the router will take the appropriate action based on its understanding of the attribute.
  - **Optional Non-Transitive:** If the attribute is not recognised, the router will not include it in the update packet when passing the route on to neighbours.

# Well-Known Mandatory Examples



- **AS-path:** The sequence of AS numbers through which the network is reached.
- **Next-hop:** The IP address of the next-hop router (where the receiving router should forward packets to). BGP routers update the next-hop address as necessary when propagating the route.

# Well-Known Mandatory Example: Origin

- The Origin attribute describes how the route was learned by BGP
  - i (IGP): via the 'network' command or route aggregation (summarization) within BGP
  - ? (Unknown or Incomplete): redistributed into BGP
  - e (External): from the legacy EGP routing protocol (obsolete)

# Well-Known Discretionary Examples

- **Local Preference:** influences the outbound path for a route from an AS
- **Atomic Aggregate:** indicates that the route is an aggregate (summary) route and more granular information may have been lost. Summary routes do not by default include the AS\_Path information from before the aggregation, MED, or BGP communities.

# AS\_SET



- The 'as-set' keyword can be included in an aggregate-address command.
- It retains the original AS path information in an AS\_SET portion of the AS\_Path.
- The AS\_SET only counts as one hop, even if it includes multiple ASs.

```
router bgp 65001  
aggregate-address 203.0.113.0 255.255.255.0 as-set summary-only
```

# Optional Transitive Examples



- **Aggregator:** identifies the AS and the router within that AS that performed summarization.
- **Community:** the administrator can assign a community value to particular routes as a method of 'tagging' them. Routing policy can be configured to take an action on those routes at a particular point in the network based on their community, for example filtering out or preferring those routes.

# Optional Non-Transitive Examples



- **MED Multi-Exit Discriminator:** influences the inbound path from neighboring AS for a route

# Weight



- **Weight** is Cisco proprietary, locally significant and never exported in BGP updates. It is not classified into a category because it is never included in BGP Update packets.