Inter-domain Routing
Setting

- Start with simpler goal of **intra-domain routing**

- Routing is the process that all routers go through to calculate the routing tables
  - each router knows how to get to every destination in the network
Network as a Graph

- Routing is essentially a problem in graph theory.
Two Approaches

- **Link state routing:**
  - every node collects a representation of the entire graph and computes shortest paths
  - forwards packets along shortest paths to destination

- **Distance vector routing:**
  - each node knows only about its next hop links
  - each node maintains a vector of costs to all dests
  - periodically exchange with neighbors its routing table
• What are the issues that we have to take into account as we generalize this to a routing protocol for the Internet?

• What should be the goals of an ideal routing protocol for the Internet?
Setting

- BGP is the “inter-domain” routing protocol
  - Each “domain” is a separately administered entity
  - Also referred to as “autonomous systems” (ASes)
  - Each AS might have multiple prefixes (a contiguous set of addresses, e.g., MIT has 18.*.*.* and UW is 128.208.*.*

- Routers route packets based on “longest prefix match”
  - Routing table contains the next hop based on a prefix basis
  - Find the best prefix match and route using it
  - What are the implications of using “longest prefix matching”? 
Business Relationships

- Neighboring ASes have business contracts
  - How much traffic to carry
  - Which destinations to reach
  - How much money to pay

- Common business relationships
  - Customer-provider
  - Peer-peer
  - Sibling
Customer-Provider Relationship

- Customer needs to be reachable from everyone
  - Provider ensures all neighbors can reach the customer
- Customer needs to reach everyone
- Payments in both directions
- Typically “95th percentile billing”
Peer-Peer relationship

- Peers exchange traffic between customers
  - AS lets its peer reach (only) its customers
  - Often the relationship is settlement-free (i.e., no $$$)
AS Structure

• Top of the Internet hierarchy
  • Has no upstream provider of its own
  • Typically has a large (inter)national backbone
  • Around 10-12 ASes: AT&T, Sprint, Level 3, ...

• Lower-layer providers (tier-2, ...)
  • Provide transit service to downstream customers
  • But need at least one provider of their own

• Stub ASes
  • Do not provide transit service
  • Connect to upstream provider(s)
What is BGP?

- Policy-based path vector routing
  - Path vector: a path of ASes
  - Respect the policies (customer-provider, peer-to-peer, etc.)
    - mechanism for filtering and selecting paths
  - at import and export time
Path-Vector Routing

- Extension of distance-vector routing
- Support flexible routing policies
- Faster convergence (avoid count-to-infinity)
- Key idea: advertise the entire path
- Distance vector: send distance metric per dest \( d \)
- Path vector: send the entire path for each dest \( d \)
• How powerful is the framework? How would you use import/export policies to influence routing?
• how to implement provider-customer relationships?
• how to implement peering relationships?

• What are the implications of BGP policy-based routing?
## BGP Route Preferences

<table>
<thead>
<tr>
<th>Priority</th>
<th>Rule</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LOCAL_PREF</td>
<td>Highest LOCAL_PREF (§4.2.3). E.g., Prefer transit customer routes over peer and provider routes.</td>
</tr>
<tr>
<td>2</td>
<td>ASPATH</td>
<td>Shortest ASPATH length (§4.3.5) Not shortest number of Internet hops or delay.</td>
</tr>
<tr>
<td>3</td>
<td>MED</td>
<td>Lowest MED preferred (§4.3.5). May be ignored, esp. if no financial incentive involved.</td>
</tr>
<tr>
<td>4</td>
<td>eBGP &gt; iBGP</td>
<td>Did AS learn route via eBGP (preferred) or iBGP?</td>
</tr>
<tr>
<td>5</td>
<td>IGP path</td>
<td>Lowest IGP path cost to next hop (egress router). If all else equal so far, pick shortest internal path.</td>
</tr>
<tr>
<td>6</td>
<td>Router ID</td>
<td>Smallest router ID (IP address). A random (but unchanging) choice; some implementations use a different tie-break such as the oldest route.</td>
</tr>
</tbody>
</table>
Hot-Potato (Early-Exit) Routing

- Hot-potato routing
  - Each router selects the closest egress point
  - ... based on the path cost in intradomain protocol

- BGP decision process
  - Highest local preference
  - Shortest AS path
  - Closest egress point
  - Arbitrary tie break
Export Policy

- Modify attributes of the active route
  - To influence the way other ASes behave
  - Example: AS prepending
- Artificially inflate AS path length seen by others
  - Convince some ASes to send traffic another way

![Diagram](image-url)
Establish session on TCP port 179

Exchange all active routes

Exchange incremental updates

While connection is ALIVE exchange route UPDATE messages
Incremental Protocol

- A node learns multiple paths to destination
  - Applies policy to select a single active route
  - ... and may advertise the route to its neighbors

- Incremental updates
  - Announcement
    - Upon selecting a new active route, add node id to path
    - ... and (optionally) advertise to each neighbor
  - Withdrawal
    - If the active route is no longer available
    - ... send a withdrawal message to the neighbors
• BGP inside an AS
  • Need to propagate BGP paths through the AS
  • Need to interface with intra-domain protocol
AS is not a single node

- Multiple routers in an AS
- Need to distribute BGP information within the AS
- Internal BGP (iBGP) sessions between routers
BGP Route

- Destination prefix (e.g., 128.112.0.0/16)
- Route attributes, including
  - AS path (e.g., “7018 88”)
  - Next-hop IP address (e.g., 12.127.0.121)
Joining BGP and IGP

- **Border Gateway Protocol (BGP)**
  - Maps a destination prefix to an egress point
  - 128.112.0.0/16 reached via 192.0.2.1

- **Interior Gateway Protocol (IGP)**
  - Used to compute paths within the AS
  - Maps an egress point to an outgoing link
An AS may learn many routes

- Multiple connections to neighboring ASes
- Multiple border routers may learn good routes
- ... with the same local-pref and AS path length
Primary-Backup Paths

Prepending will (usually) force inbound traffic from AS 1 to take primary link.

AS 1

provider

192.0.2.0/24 ASPATH = 2

primary

customer

192.0.2.0/24

backup

192.0.2.0/24 ASPATH = 2 2 2

AS 2
Prepending Doesn’t Always Work

AS 1
provider

192.0.2.0/24
ASPATH = 2

customer

AS 2

AS 3
provider

192.0.2.0/24
ASPATH = 2 2 2 2 2 2 2 2 2 2 2

backup

192.0.2.0/24
BGP Communities

AS 1
provider

AS 2

192.0.2.0/24
ASPATH = 2

primary

customer

AS 3
provider

192.0.2.0/24
ASPATH = 2
COMMUNITY = 3:70

backup

Customer import policy at AS 3:
If 3:90 in COMMUNITY then set local preference to 90
If 3:80 in COMMUNITY then set local preference to 80
If 3:70 in COMMUNITY then set local preference to 70

AS 3: normal customer local pref is 100, peer local pref is 90
Customer Installs Backup

Provider A (Tier 1) -> Provider B (Tier 1)
Provider C (Tier 2) -> Provider A (Tier 1)
Customer sends community that lowers local preference below a provider’s primary backup.
Failure Happens!

Provider A (Tier 1)

Provider B (Tier 1)

Provider C (Tier 2)

customer

customer is happy that backup was installed ...
Primary is Repaired...

Provider A (Tier 1) --- Provider B (Tier 1)

Provider C (Tier 2) --- customer

This is a stable BGP routing!

One “solution” --- reset BGP session on backup link!
• Is BGP secure?
• Is BGP high performant?
Observations

- There is no guarantee that a BGP configuration has a unique routing solution.
  - When multiple solutions exist, the (unpredictable) order of updates will determine which one wins.
- There is no guarantee that a BGP configuration has any solution
- Complex policies (weights, communities setting preferences, and so on) increase chances of routing anomalies.