Routing: Overview and Key Protocols

Shivkumar Kalyanaraman
Rensselaer Polytechnic Institute
shivkuma@ecse.rpi.edu

Based in part upon slides of Prof. Raj Jain (OSU), S. Keshav (Cornell), J. Kurose (U Mass), Noel Chiappa (MIT), Tim Griffin (AT&T), Ion Stoica (UCB),

Shivkumar Kalyanaraman
Overview

- Routing vs Forwarding vs Bridging
- Distance vector vs Link state routing
- Addressing and Routing: Scalability
- OSPF, RIP protocols
- Inter-domain Routing Issues
- BGP protocol
Routing vs. Forwarding

- **Forwarding**: select an output port based on destination address and routing table
  - Data-plane function
  - Often implemented in hardware

- **Routing**: process by which routing table is *built*..
  - ... so that the series of local forwarding decisions takes the packet to the destination with high probability, and ...*(reachability condition)*
  - ... the path chosen/resources consumed by the packet is *efficient* in some sense... *(optimality and filtering condition)*

- Control-plane function
- Implemented in software
### Forwarding Table

- Can display forwarding table using **"netstat -rn"**
- Sometimes called **"routing table"**

<table>
<thead>
<tr>
<th>Destination</th>
<th>Gateway</th>
<th>Flags</th>
<th>Ref</th>
<th>Use</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>UH</td>
<td>0</td>
<td>26492</td>
<td>lo0</td>
</tr>
<tr>
<td>192.168.2.5</td>
<td>192.168.2.5</td>
<td>U</td>
<td>2</td>
<td>13</td>
<td>fa0</td>
</tr>
<tr>
<td>193.55.114.6</td>
<td>193.55.114.6</td>
<td>U</td>
<td>3</td>
<td>58503</td>
<td>le0</td>
</tr>
<tr>
<td>192.168.3.5</td>
<td>192.168.3.5</td>
<td>U</td>
<td>2</td>
<td>25</td>
<td>qaa0</td>
</tr>
<tr>
<td>224.0.0.0</td>
<td>193.55.114.6</td>
<td>U</td>
<td>3</td>
<td>0</td>
<td>le0</td>
</tr>
<tr>
<td>default</td>
<td>193.55.114.129</td>
<td>UG</td>
<td>0</td>
<td>143454</td>
<td></td>
</tr>
</tbody>
</table>
Interconnection Devices

LAN = Collision Domain

Extended LAN = Broadcast domain

Router

Application
Transport
Network
Datalink
Physical

Gateway
Router
Bridge/Switch
Repeater/Hub

Shivkumar Kalyanaraman
Routing problem

- Collect, process, and condense global state into local forwarding information
- Global state
  - inherently large
  - dynamic
  - hard to collect
- **Hard issues:**
  - consistency, completeness, scalability
  - Impact of resource needs of sessions
Consistency

- Defn: A series of independent local forwarding decisions must lead to connectivity between any desired (source, destination) pair in the network.

- If the states are inconsistent, the network is said not to have "converged" to steady state (i.e. is in a transient state)
  - Inconsistency leads to loops, wandering packets etc
  - In general a part of the routing information may be consistent while the rest may be inconsistent.
  - Large networks => inconsistency is a scalability issue.

- Consistency can be achieved in two ways:
  - Fully distributed approach: a consistency criterion or invariant across the states of adjacent nodes
  - Signaled approach: the signaling protocol sets up local forwarding information along the path
Completeness

- **Defn:** The network as a whole and every node has sufficient information to be able to compute *all* paths.
  - In general, with more information available locally, routing algorithms tend to converge faster, because the chances of inconsistency reduce.
  - But this means that more distributed state must be collected at each node and processed.
  - The demand for completeness also limits the scalability of the algorithm.

- Since both consistency and completeness pose scalability problems, large networks have to be structured hierarchically and abstract entire networks as a single node.
Internet Routing Model

- 2 key features:
  - Dynamic routing
  - Intra- and Inter-AS routing, AS = locus of admin control

- Internet organized as "autonomous systems" (AS).
  - AS is internally connected

- Interior Gateway Protocols (IGPs) within AS.
  - Eg: RIP, OSPF, HELLO

- Exterior Gateway Protocols (EGPs) for AS to AS routing.
  - Eg: EGP, BGP-4
Dynamic Routing Model

transport (UDP)

network (IP)

routing tables

data link

physical

transport (UDP)

network (IP)

routing tables

data link

physical
Intra-AS and Inter-AS routing

Gateways:
• perform **inter-AS** routing amongst themselves
• perform **intra-AS** routers with other routers in their AS

**Gateways:**
- A.a
- A.c
- B.a
- C.b

**Network layer**
- **Inter-AS routing algorithm**
- **Intra-AS routing algorithm**

**Link layer**
- **Routing Table**

**Physical layer**
- **DL**
- **PHY**

**Gateways:**
- A.a
- A.c
- B.a
- C.b

**Intra-AS, intra-AS routing in gateway A.c**

Shivkumar Kalyanaraman
Intra-AS and Inter-AS routing: Example

Intra-AS routing within AS A

Inter-AS routing between A and B

Intra-AS routing within AS B

Host h1

Host h2
Basic Dynamic Routing Methods

- **Source-based:** source gets a map of the network, source finds route, and either
  - signals the route-setup (e.g., ATM approach)
  - encodes the route into packets (inefficient)

- **Link state** routing: *per-link* information
  - Get map of network (in terms of link states) at all nodes and find next-hops locally.
  - Maps consistent => next-hops consistent

- **Distance vector:** *per-node* information
  - At every node, set up distance signposts to destination nodes (a vector)
  - Setup this by peeking at neighbors’ signposts.
The subset of a shortest path is also the shortest path between the two intermediate nodes.

Corollary:
If the shortest path from node i to node j, with distance $D(i,j)$ passes through neighbor k, with link cost $c(i,k)$, then:

$$D(i,j) = c(i,k) + D(k,j)$$
Distance Vector

$DV = \text{Set (vector) of Signposts, one for each destination}$
Distance Vector (DV) Approach

Consistency Condition: \( D(i,j) = c(i,k) + D(k,j) \)

- The **DV (Bellman-Ford) algorithm** evaluates this recursion iteratively.
- In the \( m^{th} \) iteration, the consistency criterion holds, assuming that each node sees all nodes and links \( m \)-hops (or smaller) away from it (i.e. an \( m \)-hop view)

Example network

A’s 1-hop view (After 1\(^{st} \) iteration)

A’s 2-hop view (After 2\(^{nd} \) Iteration)
Distance Vector (DV) Example

- A’s distance vector $D(A, *)$:
  - After Iteration 1 is: $[0, 7, \text{INFINITY}, \text{INFINITY}, 1]$
  - After Iteration 2 is: $[0, 7, 8, 3, 1]$
  - After Iteration 3 is: $[0, 7, 5, 3, 1]$
  - After Iteration 4 is: $[0, 6, 5, 3, 1]$

Example network

A’s 1-hop view (After 1st iteration)

A’s 2-hop view (After 2nd Iteration)

Shivkumar Kalyanaraman
**Link State (LS) Approach**

- The *link state (Dijkstra) approach is iterative, but it pivots around destinations j, and their predecessors k = p(j)*.
- Observe that an alternative version of the consistency condition holds for this case: \( D(i,j) = D(i,k) + c(k,j) \)
- Each node i collects all link states \( c(*,*) \) first and runs the complete Dijkstra algorithm locally.

Shivkumar Kalyanaraman
Dijkstra’s algorithm: example

<table>
<thead>
<tr>
<th>Step</th>
<th>set N</th>
<th>D(B),p(B)</th>
<th>D(C),p(C)</th>
<th>D(D),p(D)</th>
<th>D(E),p(E)</th>
<th>D(F),p(F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A</td>
<td>2,A</td>
<td>5,A</td>
<td>1,A</td>
<td>infinity</td>
<td>infinity</td>
</tr>
<tr>
<td>1</td>
<td>AD</td>
<td>2,A</td>
<td>4,D</td>
<td>2,D</td>
<td>infinity</td>
<td>infinity</td>
</tr>
<tr>
<td>2</td>
<td>ADE</td>
<td>2,A</td>
<td>3,E</td>
<td></td>
<td>4,E</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ADEB</td>
<td></td>
<td>3,E</td>
<td></td>
<td>4,E</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ADEBC</td>
<td></td>
<td></td>
<td></td>
<td>4,E</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ADEBCF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The shortest-paths spanning tree rooted at A is called an SPF-tree

Shivkumar Kalyanaraman
Summary: Distributed Routing Techniques

**Link State**
- Topology information is **flooded** within the routing domain.
- Best end-to-end paths are computed locally at each router.
- Best end-to-end paths determine next-hops.
- Based on minimizing some notion of distance
- Works only if policy is **shared** and **uniform**
- Examples: OSPF, IS-IS

**Vectoring**
- Each router knows little about network topology
- Only best next-hops are chosen by each router for each destination network.
- Best end-to-end paths result from composition of all next-hop choices
- Does not require any notion of distance
- Does not require uniform policies at all routers
- Examples: RIP, BGP
RIP: Routing Information Protocol

- Uses hop count as metric (max: 16 is infinity)
- Tables (vectors) "advertised" to neighbors every 30 s.
  - Each advertisement: up to 25 entries
- No advertisement for 180 sec: neighbor/link declared dead
  - Routes via neighbor invalidated
- New advertisements sent to neighbors (Triggered updates)
  - Neighbors in turn send out new advertisements (if tables changed)
- Link failure info quickly propagates to entire net
- poison reverse used to prevent ping-pong loops (infinite distance = 16 hops)
RIPv1 Problems (Continued)

- Split horizon/poison reverse does not guarantee to solve count-to-infinity problem
  - $16 = \text{infinity} \Rightarrow \text{RIP for small networks only!}$
- Slow convergence
- Broadcasts consume non-router resources
- RIPv1 does not support subnet masks (VLSMs)
- No authentication
RIPv2

- Why? Installed base of RIP routers
- Provides:
  - VLSM support
  - Authentication
  - Multicasting
  - “Wire-sharing” by multiple routing domains,
  - Tags to support EGP/BGP routes.
- Uses reserved fields in RIPv1 header.
- First route entry replaced by authentication info.
**Link State Protocols**

- Key: Create a network “map” at each node.

- 1. Node collects the state of its connected links and forms a “Link State Packet” (LSP)

- 2. Flood LSP => reaches every other node in the network and everyone now has a network map.

- 3. Given map, run Dijkstra’s shortest path algorithm (SPF) => get paths to all destinations

- 4. Routing table = next-hops of these paths.

- 5. Hierarchical routing: organization of areas, and filtered control plane information flooded.
Hello: Packet Format

![Diagram of network topology with interface indexes and costs labeled](image)

**Figure 4.4** Point-to-point network topology, with interface indexes and costs labeled.

![Table of OSPF Hello packet details](image)

**Figure 4.6** An OSPF Hello packet.
Topology Dissemination

- A.k.a LSP distribution
- 1. *Flood* LSPs on links except incoming link
  - Require at most $2E$ transfers for n/w with $E$ edges
- 2. *Sequence numbers* to detect duplicates
  - Why? Routers/links may go down/up
  - Issue: wrap-around, larger sequence number is not the most recent!
OSPF Router-LSA: Scenario

Figure 4.4 Point-to-point network topology, with interface IFIndexes and costs labeled.
Router-LSA:

Figure 4.4 Point-to-point network topology, with interface IF indexes and costs labeled.

Figure 4.5 Router 10.1.1.1’s router-LSA.
Topography Dissemination (Continued)

- **Checksum field:**
  - Drop packet if in error, get retransmission from neighbor

- **Age field** (similar to TTL)
  - Number of seconds since LSA originated
  - Periodically incremented after acceptance
  - Originating router refreshes LSA after 30 min
  - Delete if Age = MaxAge
  - Low age field + large seq # => that LSA is flapping or frequently changing …
Recovering from a partition

- On partition, LSP databases can get out of synch

- Databases described by database descriptor records
- Routers on each side of a newly restored link talk to each other to update databases (determine missing and out-of-date LSPs) => **selective synchronization**

Shivkumar Kalyanaraman
Inter-Domain Routing: Big Picture

Large number of diverse networks

Shivkumar Kalyanaraman
Requirements for Inter-AS Routing

- Should scale for the size of the global Internet.
  - Focus on reachability, not optimality
  - Use address aggregation techniques to minimize core routing table sizes and associated control traffic
  - At the same time, it should allow flexibility in topological structure (eg: don’t restrict to trees etc)

- Allow policy-based routing between autonomous systems
  - Policy refers to arbitrary preference among a menu of available routes (based upon routes’ attributes)
  - Fully distributed routing (as opposed to a signaled approach) is the only possibility.
  - Extensible to meet the demands for newer policies.
Who speaks Inter-AS routing?

- Two types of routers
  - Border router (Edge), Internal router (Core)
  - Two border routers of different ASes will have a BGP session

Shivkumar Kalyanaraman
Customers and Providers

Provider

Customer

IP traffic

Customer pays provider for access to the Internet
Nontransit vs. Transit ASes

Internet Service providers (ISPs) have transit networks.

Nontransit AS might be a corporate or campus network. Could be a “content provider”

Traffic NEVER flows from ISP 1 through NET A to ISP 2.
The Peering Relationship

Peers provide transit between their respective customers

Peers do not provide transit between peers

Peers (often) do not exchange $$$

Shivkumar Kalyanaraman
BGP-4

- **BGP = Border Gateway Protocol**
- Is a **Policy-Based** routing protocol
- Is the **de facto EGP** of today’s global Internet
- Relatively simple protocol, but configuration is complex and the entire world can see, and be impacted by, your mistakes.
  - **1989 : BGP-1 [RFC 1105]**
    - Replacement for EGP (1984, RFC 904)
  - **1990 : BGP-2 [RFC 1163]**
  - **1991 : BGP-3 [RFC 1267]**
  - **1995 : BGP-4 [RFC 1771]**
    - Support for Classless Interdomain Routing (CIDR)
BGP Operations (Simplified)

Establish session on TCP port 179

Exchange all active routes

Exchange incremental updates

While connection is ALIVE exchange route UPDATE messages
Four Types of BGP Messages

- **Open**: Establish a peering session.
- **Keep Alive**: Handshake at regular intervals.
- **Notification**: Shuts down a peering session.
- **Update**: Announcing new routes or withdrawing previously announced routes.

**announcement** = **prefix** + **attributes values**
Two Types of BGP Neighbor Relationships

- External Neighbor (eBGP) in a different Autonomous Systems
- Internal Neighbor (iBGP) in the same Autonomous System

iBGP is routed (using IGP!)
I-BGP and E-BGP

IGP: Interior Gateway Protocol. Examples: IS-IS, OSPF

R1
AS1
E-BGP
R2
IGP
R3
AS2
A
E-BGP
R4
AS3
R5

border router
internal router

announce B
IBGP vs EBGP

- I-BGP nodes: typically ABRs, or other nodes where default routes terminate
- I-BGP peering sessions between every pair of routers within an AS: full mesh.

![Diagram of IBGP sessions](image)
Route Reflection

AS1

128.23.0.0/16

RR-C1

RR-C2

RR-C3

RR-C4

RR1

RR2

RR3

ER

AS2

EBGP

IBGP

10.0.0.0/24

10.0.0.0/24

Shivkumar Kalyanaraman
AS Confederations

- Divide and conquer: Divides a large AS into sub-ASs
Address Aggregation: CIDR

Inter-domain Routing Without CIDR

Service Provider

204.71.0.0
204.71.1.0
204.71.2.0
............
204.71.255.0

Global Internet Routing Mesh

Inter-domain Routing With CIDR

Service Provider

204.71.0.0
204.71.1.0
204.71.2.0
............
204.71.255.0

204.71.0.0/16

Global Internet Routing Mesh

Shivkumar Kalyanaraman
RFC 1519: Classless Inter-Domain Routing (CIDR)

Pre-CIDR: Network ID ended on 8-, 16, 24- bit boundary
CIDR: Network ID can end at any bit boundary

IP Address: 12.4.0.0  IP Mask: 255.254.0.0

<table>
<thead>
<tr>
<th>Address</th>
<th>00001100</th>
<th>00000100</th>
<th>00000000</th>
<th>00000000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask</td>
<td>11111111</td>
<td>11111110</td>
<td>00000000</td>
<td>00000000</td>
</tr>
</tbody>
</table>

Network Prefix for hosts

Usually written as 12.4.0.0/15, a.k.a “supernetting”
### Longest Prefix Match (Classless) Forwarding

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next Hop</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0/0</td>
<td>10.14.11.33</td>
<td>ATM 5/0/9</td>
</tr>
<tr>
<td>12.0.0.0/8</td>
<td>10.14.22.19</td>
<td>ATM 5/0/8</td>
</tr>
<tr>
<td>12.4.0.0/15</td>
<td>10.1.3.77</td>
<td>Ethernet 0/1/3</td>
</tr>
<tr>
<td>12.5.8.0/23</td>
<td>attached</td>
<td>Serial 1/0/7</td>
</tr>
</tbody>
</table>

Destination = 12.5.9.16

---

**Payload**

---

**IP Forwarding Table**

- **OK**
- **better**
- **even better**
- **best!**
What is Routing Policy

- Policy refers to *arbitrary preference among a menu of available routes* (based upon routes’ attributes)
  - Public description of the relationship between external BGP peers
  - Can also describe internal BGP peer relationship

- Eg: Who are my BGP peers
- What routes are
  - Originated by a peer
  - Imported from each peer
  - Exported to each peer
  - Preferred when multiple routes exist
- What to do if no route exists?
BGP Route Processing

Receive BGP Updates

Apply Policy = filter routes & tweak attributes

Based on Attribute Values

Best Routes

Apply Policy = filter routes & tweak attributes

Transmit BGP Updates

Apply **Import Policies**

Best Route Selection

Apply **Export Policies**

Install forwarding Entries for best Routes.

**IP Forwarding Table**
Policy Implementation Flow

- **Incoming**: Adj RIB In
- **Main BGP RIB**
- **Outgoing**: Adj RIB Out

- **IGPs**
- **Main RIB/FIB**
- **Static & HW Info**

Shivkumar Kalyanaraman
Import and Export Policies

- For **inbound** traffic
  - Filter outbound routes
  - Tweak attributes on outbound routes in the hope of influencing your neighbor’s best route selection
- For **outbound** traffic
  - Filter **inbound** routes
  - Tweak attributes on inbound routes to influence best route selection

*In general, an AS has more control over outbound traffic*
# BGP Policy Knob: Attributes

<table>
<thead>
<tr>
<th>Value</th>
<th>Code</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ORIGIN</td>
<td>[RFC1771]</td>
</tr>
<tr>
<td>2</td>
<td>AS_PATH</td>
<td>[RFC1771]</td>
</tr>
<tr>
<td>3</td>
<td>NEXT_HOP</td>
<td>[RFC1771]</td>
</tr>
<tr>
<td>4</td>
<td>MULTI_EXIT_DISC</td>
<td>[RFC1771]</td>
</tr>
<tr>
<td>5</td>
<td>LOCAL_PREF</td>
<td>[RFC1771]</td>
</tr>
<tr>
<td>6</td>
<td>ATOMIC_AGGREGATE</td>
<td>[RFC1771]</td>
</tr>
<tr>
<td>7</td>
<td>AGGREGATOR</td>
<td>[RFC1771]</td>
</tr>
<tr>
<td>8</td>
<td>COMMUNITY</td>
<td>[RFC1771]</td>
</tr>
<tr>
<td>9</td>
<td>ORIGINATOR_ID</td>
<td>[RFC2796]</td>
</tr>
<tr>
<td>10</td>
<td>CLUSTER_LIST</td>
<td>[RFC2796]</td>
</tr>
<tr>
<td>11</td>
<td>DPA</td>
<td>[Chen]</td>
</tr>
<tr>
<td>12</td>
<td>ADVERTISER</td>
<td>[RFC1863]</td>
</tr>
<tr>
<td>13</td>
<td>RCID_PATH / CLUSTER_ID</td>
<td>[RFC1863]</td>
</tr>
<tr>
<td>14</td>
<td>MP_REACH_NLRI</td>
<td>[RFC2283]</td>
</tr>
<tr>
<td>15</td>
<td>MP_UNREACH_NLRI</td>
<td>[RFC2283]</td>
</tr>
<tr>
<td>16</td>
<td>EXTENDED_COMMUNITIES</td>
<td>[Rosen]</td>
</tr>
<tr>
<td>...</td>
<td>reserved for development</td>
<td></td>
</tr>
<tr>
<td>255</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We will cover a subset of these attributes.

Not all attributes need to be present in every announcement.

From IANA: [http://www.iana.org/assignments/bgp-parameters](http://www.iana.org/assignments/bgp-parameters)
UPDATE message in BGP

- Primary message between two BGP speakers.
- Used to advertise/withdraw IP prefixes (NLRI)
- **Path attributes** field: unique to BGP
  - Apply to all prefixes specified in NLRI field
  - Optional vs **Well-known**; **Transitive** vs Non-transitive

  **2 octets**

<table>
<thead>
<tr>
<th>Withdrawn Routes Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Withdrawn Routes (variable length)</td>
</tr>
<tr>
<td>Total Path Attributes Length</td>
</tr>
<tr>
<td>Path Attributes (variable length)</td>
</tr>
<tr>
<td>Network Layer Reachability Info. (NLRI: variable length)</td>
</tr>
</tbody>
</table>
Path Attributes: ORIGIN

- **ORIGIN:**
  - Describes how a prefix came to BGP at the origin AS
  - Prefixes are learned from a source and "injected" into BGP:
    - Directly connected interfaces, manually configured static routes, dynamic IGP or EGP
  - Values:
    - IGP (EGP): Prefix learnt from IGP (EGP)
    - INCOMPLETE: Static routes
Path Attributes: AS-PATH

- List of **ASs** thru which the prefix announcement has passed. AS on path adds ASN to AS-PATH
- Eg: 138.39.0.0/16 originates at AS1 and is advertised to AS3 via AS2.
- Eg: AS-SEQUENCE: “100 200”
- Used for loop detection and path selection
Traffic Often Follows ASPATH

AS 1: 135.207.0.0/16

AS 2

AS 3

AS 4

135.207.0.0/16
ASPATH = 3 2 1

IP Packet
Dest = 135.207.44.66
... But It Might Not

AS 2 filters all subnets with masks longer than /24

From AS 4, it may look like this packet will take path 3 2 1, but it actually takes path 3 2 5

Shivkumar Kalyanaraman
Shorter AS-PATH Doesn’t Mean Shorter # Hops

BGP says that path 4 1 is better than path 3 2 1

Duh!
Path Attributes: NEXT-HOP

- Next-hop: node to which packets must be sent for the IP prefixes. May not be same as peer.
- UPDATE for 180.20.0.0, NEXT-HOP = 170.10.20.3
Recursive Lookup

- If routes (prefix) are learnt thru iBGP, NEXT-HOP is the iBGP router which originated the route.
  - Note: iBGP peer might be several IP-level hops away as determined by the IGP
  - Hence BGP NEXT-HOP is not the same as IP next-hop
  - BGP therefore checks if the “NEXT-HOP” is reachable through its IGP.
  - If so, it installs the IGP next-hop for the prefix
  - This process is known as “recursive lookup” – the lookup is done in the control-plane (not data-plane) before populating the forwarding table.
- Example in next slide
Join EGP with IGP For Connectivity

135.207.0.0/16
Next Hop = 192.0.2.1

192.0.2.0/30
Next Hop = 192.0.2.1

192.0.2.0/30
Next Hop = 192.0.2.1

Forwarding Table

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.0.2.0/30</td>
<td>10.10.10.10</td>
</tr>
<tr>
<td>135.207.0.0/16</td>
<td>192.0.2.1</td>
</tr>
</tbody>
</table>

Forwarding Table

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>135.207.0.0/16</td>
<td>10.10.10.10</td>
</tr>
<tr>
<td>192.0.2.0/30</td>
<td>10.10.10.10</td>
</tr>
</tbody>
</table>

Shivkumar Kalyanaraman
Load-Balancing Knobs in BGP

- **LOCAL-PREF**: outbound traffic, local preference (box-level knob)
- **MED**: Inbound-traffic, typically from the same ISP (link-level knob)
Path Attribute: LOCAL-PREF

- Locally configured indication about which path is preferred to exit the AS in order to reach a certain network. Default value = 100. Higher is better.
 Attributes: MULTI-EXIT Discriminator

- Also called METRIC or MED Attribute. Lower is better
- AS1: multihomed customer.
- AS2 (provider) includes MED to AS1
- AS1 chooses which link (NEXTHOP) to use
- Eg: traffic to AS3 can go thru Link1, and AS2 thru Link2

Shivkumar Kalyanaraman
MEDs Can Export Internal Instability

192.44.78.0/24 MED = 15

192.44.78.0/24 MED = 56 OR 10

Heavy Content Web Farm

Shivkumar Kalyanaraman
ASPATH Padding: Shed inbound traffic

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

customer AS 2

provider AS 1

192.0.2.0/24

ASPATH = 2

192.0.2.0/24

ASPATH = 2

ASPATH = 2 2 2
If AS 1 does not announce the more specific prefix, then most traffic to AS 2 will go through AS 3 because it is a longer match.

AS 2 is “punching a hole” in the CIDR block of AS 1 => subverts CIDR
CIDR at Work, No load balancing

Table at ISP3

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next Hop</th>
<th>ORIGIN AS</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.32/11</td>
<td>ISP1</td>
<td>ISP1</td>
</tr>
<tr>
<td>140.64/10</td>
<td>ISP2</td>
<td>ISP2</td>
</tr>
</tbody>
</table>

AS1
128.40/16
140.127/16

ISP1
128.32/11

ISP2
140.64/10

ISP3

Link A
128.40/16

Link B
140.127/16

140.127/16

ORIGIN AS
Next Hop
Prefix
CIDR Subverted for Load Balancing

Table at ISP3

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next Hop</th>
<th>ORIGIN AS</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.32/11</td>
<td>ISP1</td>
<td>ISP1</td>
</tr>
<tr>
<td>140.64/10</td>
<td>ISP2</td>
<td>ISP2</td>
</tr>
<tr>
<td>140.255.20/24</td>
<td>ISP1</td>
<td>AS1</td>
</tr>
<tr>
<td>128.42.10/24</td>
<td>ISP2</td>
<td>AS1</td>
</tr>
</tbody>
</table>

Diagram:
- AS1: 128.40/16, 140.127/16
- ISP1: 128.32/11
- ISP2: 140.64/10
- ISP3: 140.255.20/24, 128.40/16, 128.42.10/24, 140.127/16
How Can Routes be Colored?

BGP Communities

A community value is 32 bits

By convention, first 16 bits is ASN indicating who is giving it an interpretation

community number

- Used within and between ASes
- The set of ASes must agree on how to interpret the community value
- Very powerful BECAUSE it has no (predefined) meaning

Community Attribute = a list of community values. (So one route can belong to multiple communities)

Two reserved communities

do_not_advertise 0xFFFFFF02: don’t pass to BGP neighbors

do_not_export = 0xFFFFFF01: don’t export out of AS

RFC 1997 (August 1996)
Communities Example

- 1:100
  - Customer routes
- 1:200
  - Peer routes
- 1:300
  - Provider Routes

To Customers
- 1:100, 1:200, 1:300
To Peers
- 1:100
To Providers
- 1:100

Import

Export

AS 1
BGP Route Selection Process

*Series of tie-breaker decisions...*

- If NEXTHOP is inaccessible do not consider the route.
- Prefer largest LOCAL-PREF
- If same LOCAL-PREF prefer the shortest AS-PATH.
- If all paths are external prefer the lowest ORIGIN code (IGP<EGP<INCOMPLETE).
- If ORIGIN codes are the same prefer the lowest MED.
- If MED is same, prefer min-cost NEXT-HOP
- If routes learned from EBGP or IBGP, prefer paths learnt from EBGP
- Final tie-break: Prefer the route with I-BGP ID (IP address)
Route Selection Summary

- Highest Local Preference
  - Enforce relationships

- Shortest ASPATH
- Lowest MED
- i-BGP < e-BGP
- Lowest IGP cost to BGP egress

- Lowest router ID
  - Throw up hands and break ties
BGP Table Growth

Large BGP Tables Considered Harmful

- Routing tables must store best routes and alternate routes
- Burden can be large for routers with many alternate routes (route reflectors for example)
- Routers have been known to die
- Increases CPU load, especially during session reset
Routing Concepts
- DV and LS algorithms
- RIP, OSPF, BGP