CS4450

Computer Networks:
Architecture and Protocols

Lecture 14
Border-Gateway Protocol

Rachit Agarwal
Announcements

- Exam 2 on 10/24

- 10/26: Live coding session; please bring your laptops
Goals for Today’s Lecture

- Deep dive into Inter-domain routing (Border-Gateway Protocol (BGP))
  - One of the most non-intuitive protocols
  - Driven by “business goals”, rather than “performance goals”
    - I will try to provide as much intuition as possible
    - But, for the above reasons, BGP is one of the harder protocols

- Understanding BGP
  - Do a lot of small examples
  - We will focus on a synchronous version:
    - One node in the network acts at a time
    - In practice, BGP implementations are asynchronous
Recap from last lecture
Recap: What does a computer network look like?

“Autonomous System (AS)” or “Domain”
Region of a network under a single administrative entity

An “end-to-end” route

“Border Routers”

“Interior Routers”
Recap: IP addressing enables Scalable Routing
Recap: IP addressing enables Scalable Routing

Can add new hosts/networks without updating the routing entries at France Telecom
Recap: Business Relationships Shape Topology and Policy

- Three basic kinds of relationships between ASes
  - AS A can be AS B’s *customer*
  - AS A can be AS B’s *provider*
  - AS A can be AS B’s *peer*

- Business implications
  - Customer *pays* provider
  - Peers *don’t pay* each other
    - Exchange roughly equal traffic
Recap: Why Peer?

Relations between ASes
- provider
- peer
- customer
- peer

Business Implications
- Customers pay provider
- Peers don’t pay each other

E.g., D and E talk a lot
Peering saves B and C money
ASes provide “transit” between their customers

Peers do not provide transit between other peers
Border Gateway Protocol
Administrative Structure Shapes Inter-domain Routing

- ASes want freedom to pick routes based on policy
  - "My traffic can’t be carried over my competitor’s network!"
  - "I don’t want to carry A’s traffic through my network!"
- Cannot be expressed as Internet-wide “least cost”

- ASes want autonomy
  - Want to choose their own internal routing protocol
  - Want to choose their own policy

- ASes want privacy
  - Choice of network topology, routing policies, etc.
Inter-domain Routing: Setup

- Destinations are IP prefixes (12.0.0.0/8)
- Nodes are Autonomous Systems (ASes)
  - Internals of each AS are hidden
- Links represent both physical links and business relationships
- BGP (Border Gateway Protocol) is the Interdomain routing protocol
  - Implemented by AS border routers
BGP

An AS advertises its best routes to one or more IP prefixes.

Each AS selects the “best” route it hears advertised for a prefix.

Sound familiar?
BGP Inspired by Distance Vector

- Per-destination route advertisements
- No global sharing of network topology
- Iterative and distributed convergence on paths
- But, four key differences
BGP vs. DV

(1) BGP does not pick the shortest path routes!

- BGP selects route based on policy, not shortest distance/least cost

Node 2 may prefer 2, 3, 1 over 2, 1

- How do we avoid loops?
BGP vs. DV

(2) Path-vector Routing

- Idea: advertise the entire path
  - Distance vector: send *distance metric* per dest. d
  - Path vector: send the *entire path* for each dest. d
Loop Detection with Path-Vector

- Node can easily detect a loop
  - Look for its **own node identifier** in the path
- Node can simply **discard** paths with loops
  - e.g. node 1 sees itself in the path 3, 2, 1
BGP vs. DV

(2) Path-vector Routing

- Idea: advertise the entire path
  - Distance vector: send *distance metric* per dest. d
  - Path vector: send the *entire path* for each dest. d

- Benefits
  - Loop avoidance is easy
  - Flexible policies based on entire path
(3) Selective Route Advertisement

- For policy reasons, an AS may choose not to advertise a route to a destination.

- As a result, reachability is not guaranteed even if the graph is connected.

Example: AS#2 does not want to carry traffic between AS#1 and AS#3.
BGP vs. DV

(4) BGP may aggregate routes

- For scalability, BGP may aggregate routes for different prefixes

![Diagram of BGP routing]
BGP Outline

- BGP Policy
  - Typical policies and implementation

- BGP protocol details

- Issues with BGP
Policy:

Imposed in how routes are selected and exported

- **Selection**: Which path to use
  - Controls whether / how traffic *leaves* the network
- **Export**: Which path to advertise
  - Controls whether / how traffic *enters* the network
Typical Selection Policy

- In decreasing order of priority:
  1. Make or save **money** (send to customer > peer > provider)
  2. Maximize **performance** (smallest AS path length)
  3. Minimize use of my **network bandwidth** (“hot potato”)
  4. ...

## Typical Export Policy

<table>
<thead>
<tr>
<th>Destination prefix advertised by...</th>
<th>Export route to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>Everyone (providers, peers, other customers)</td>
</tr>
<tr>
<td>Peer</td>
<td>Customers</td>
</tr>
<tr>
<td>Provider</td>
<td>Customers</td>
</tr>
</tbody>
</table>

Known as the “Gao-Rexford” rules
Capture common *(but not required!)* practice
BGP is Inspired by Distance Vector

- Per-destination route advertisements
- No global sharing of network topology
- Iterative and distributed convergence on paths
- But, **four key differences**
  - BGP does not pick shortest paths
  - Each node announces one or multiple PATHs per destination
  - Selective Route advertisement: not all paths are announced
  - BGP may aggregate paths
    - may announce one path for multiple destinations
BGP Outline

- BGP Policy
  - Typical policies and implementation

- BGP protocol details

- Issues with BGP
Policy:

Imposed in how routes are selected and exported

- **Selection**: Which path to use
  - Controls whether / how traffic leaves the network
- **Export**: Which path to advertise
  - Controls whether / how traffic enters the network
Typical Selection Policy

- In decreasing order of priority:
  1. Make or save **money** (send to customer > peer > provider)
  2. Maximize **performance** (smallest AS path length)
  3. Minimize use of my **network bandwidth** (“hot potato”)
  4. ...
### Typical Export Policy

<table>
<thead>
<tr>
<th>Destination prefix advertised by...</th>
<th>Export route to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>Everyone (providers, peers, other customers)</td>
</tr>
<tr>
<td>Peer</td>
<td>Customers</td>
</tr>
<tr>
<td>Provider</td>
<td>Customers</td>
</tr>
</tbody>
</table>

Known as the “Gao-Rexford” rules
Capture common *(but not required!)* practice
With Gao-Rexford, the AS policy graph is a DAG (directed acyclic graph) and routes are “valley free”
BGP Outline

- BGP Policy
  - Typical policies and implementation

- BGP protocol details

- Issues with BGP
Who speaks BGP?

Border routers at an Autonomous System

Border routers

Internal router
What Does “speak BGP” Mean?

- Implement the **BGP Protocol Standard**
  - Internet Engineering Task Force (IETF) RFC 4271

- Specifies what messages to exchange with other BGP “speakers”
  - Message **types** (e.g. route advertisements, updates)
  - Message **syntax**

- Specifies how to process these messages
  - When you receive a BGP update, do x
  - Follows BGP state machine in the protocol spec and policy decisions, etc.
A border router speaks BGP with border routers in other ASes
A border router speaks BGP with other (interior and border) routers in its own AS
**eBGP, iBGP, IGP**

- **eBGP**: BGP sessions between border routers in **different** ASes
  - Learn routes to external destinations

- **iBGP**: BGP sessions between border routers and other routers within the **same** AS
  - Distribute externally learned routes internally

- **IGP**: Interior Gateway Protocol = **Intradomain** routing protocol
  - Provides internal reachability
  - e.g. OSPF, RIP
1. Provide internal reachability (IGP)  
2. Learn routes to external destinations (eBGP)  
3. Distribute externally learned routes internally (iBGP)  
4. Travel shortest path to egress (IGP)
Basic Messages in BGP

- **Open**
  - Establishes BGP session

- **Update**
  - Inform neighbor of *new routes*
  - Inform neighbor of *old routes* that become inactive

- **Keepalive**
  - Inform neighbor that connection is still viable
Route Updates

- Format: \(<IP \text{ prefix}: \text{route attributes}>\)
- Two kinds of updates:
  - **Announcements**: new routes or changes to existing routes
  - **Withdrawals**: remove routes that no longer exist
- Route Attributes
  - Describe routes, used in selection/export decisions
  - Some attributes are **local**
    - i.e. private within an AS, not included in announcements
  - Some attributes are **propagated** with eBGP route announcements
  - Many standardized attributes in BGP
Route Attributes (1): ASPATH

- Carried in route announcements
- Vector that lists all the ASes a route advertisement has traversed (in reverse order)
Route Attributes (2): LOCAL PREF

- “Local Preference”
- Used to choose between different AS paths
- The higher the value, the more preferred
- Local to an AS; carried only in iBGP messages

BGP table at AS4:

<table>
<thead>
<tr>
<th>Destination</th>
<th>AS Path</th>
<th>Local Pref</th>
</tr>
</thead>
<tbody>
<tr>
<td>140.20.1.0/24</td>
<td>AS3 AS1</td>
<td>300</td>
</tr>
<tr>
<td>140.20.1.0/24</td>
<td>AS2 AS1</td>
<td>100</td>
</tr>
</tbody>
</table>
**Route Attributes (3) : MED**

- “Multi-Exit Discriminator”
- Used when ASes are interconnected via two or more links
- Specifies how close a prefix is to the link it is announced on
- Lower is better
- AS announcing prefix sets MED
- AS receiving prefix (optionally!) uses MED to select link
Route Attributes (4): IGP Cost

- Used for hot-potato routing
- Each router selects the closest egress point based on the path cost in intra-domain protocol

![Diagram showing network with nodes A, B, C, D, E, F, G and edges with costs 3, 5, 8, 4, 9, 8, 10. Arrows indicate direction of traffic towards dst.]
Using Attributes

- Rules for route selection in priority order

1. Make or save **money** (send to customer > peer > provider)
2. Maximize **performance** (smallest AS path length)
3. Minimize use of my **network bandwidth** (“hot potato”)
4. …
Using Attributes

- Rules for route selection in priority order

<table>
<thead>
<tr>
<th>Priority</th>
<th>Rule</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LOCAL PREF</td>
<td>Pick highest LOCAL PREF</td>
</tr>
<tr>
<td>2</td>
<td>ASPATH</td>
<td>Pick shortest ASPATH length</td>
</tr>
<tr>
<td>3</td>
<td>MED</td>
<td>Lowest MED preferred</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>iBGP path</td>
<td>Lowest IGP cost to next hop (egress router)</td>
</tr>
<tr>
<td>6</td>
<td>Router ID</td>
<td>Smallest next-hop router’s IP address as tie-breaker</td>
</tr>
</tbody>
</table>
Open ended programming.
Constrained only by vendor configuration language
BGP Outline

- BGP Policy
  - Typical policies and implementation

- BGP protocol details

- Issues with BGP
BGP: Issues

- Reachability
- Security
- Convergence
- Performance
- Anomalies
Reachability

- In normal routing, if graph is connected then reachability is assured.
- With policy routing, this doesn’t always hold.
Security

- An AS can claim to serve a prefix that they actually don’t have a route to (blackholing traffic)
  - Problem not specific to policy or path vector
  - Important because of AS autonomy
  - **Fixable**: make ASes prove they have a path

- But…

- AS may forward packets along a route different from what is advertised
  - Tell customers about a fictitious short path…
  - Much harder to fix!
Convergence

- If all AS policies follow Gao-Rexford rules,
  - Then BGP is guaranteed to converge (safety)

- For arbitrary policies, BGP may fail to converge!
BGP Example (All good)

![Diagram of a network with nodes and weighted edges]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>430</td>
</tr>
<tr>
<td>R3</td>
<td>130</td>
<td>20</td>
<td>30</td>
<td>430</td>
</tr>
</tbody>
</table>

GOOD GADGET
Example of Policy Oscillation

“1” prefers “1 3 0” over “1 0” to reach “0”
Step-by-step Policy Oscillation

Initially: nodes 1, 2, 3 know only shortest path to 0
1 advertises its path 1 0 to 2

Step-by-step Policy Oscillation
Step-by-step Policy Oscillation
3 advertises its path 3 0 to 1
Step-by-step Policy Oscillation
Step-by-step Policy Oscillation

1 withdraws its path 1 0 from 2
Step-by-step Policy Oscillation
2 advertises its path 2 0 to 3

Step-by-step Policy Oscillation

advertise: 2 0
Step-by-step Policy Oscillation
Step-by-step Policy Oscillation

3 withdraws its path 3 0 from 1
Step-by-step Policy Oscillation
Step-by-step Policy Oscillation

1 advertises its path 1 0 to 2
Step-by-step Policy Oscillation

2 withdraws its path 2 0 from 3

withdraw: 2 0
Step-by-step Policy Oscillation

We are back to where we started!
BGP Example (Persistent Loops)

![Diagram of a network with nodes and weights](image)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>-</td>
</tr>
<tr>
<td>R2</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>420</td>
</tr>
<tr>
<td>R3</td>
<td>10</td>
<td>20</td>
<td>3420</td>
<td>420</td>
</tr>
<tr>
<td>R4</td>
<td>10</td>
<td>210</td>
<td>3420</td>
<td>420</td>
</tr>
<tr>
<td>R5</td>
<td>10</td>
<td>210</td>
<td>3420</td>
<td>-</td>
</tr>
<tr>
<td>R6</td>
<td>10</td>
<td>210</td>
<td>30</td>
<td>-</td>
</tr>
<tr>
<td>R7</td>
<td>130</td>
<td>210</td>
<td>30</td>
<td>-</td>
</tr>
<tr>
<td>R8</td>
<td>130</td>
<td>20</td>
<td>30</td>
<td>-</td>
</tr>
<tr>
<td>R9</td>
<td>130</td>
<td>20</td>
<td>30</td>
<td>420</td>
</tr>
<tr>
<td>R10</td>
<td>130</td>
<td>20</td>
<td>3420</td>
<td>420</td>
</tr>
<tr>
<td>R11</td>
<td>10</td>
<td>20</td>
<td>3420</td>
<td>420</td>
</tr>
</tbody>
</table>
The stable paths problem is illustrated in Fig. 2(c). The stable path assignment implicitly defines a tree rooted at the origin. Note, however, that this is not always a spanning tree. The stable path assignment also called a unique solution to this problem. The solution for problem called the shortest path trees. Note that the ranking at node 4 breaks ties between prefer shorter paths to longer paths and the solutions are shortest. For example, Fig. 2(a) presents a stable paths problem called BAD GADGET and its two solutions. If no such assignment exists, then solvable. Therefore, any stable path adds one permitted path. So far, our examples each has had at most one solution. This is shown in Fig. 3(c). No other path assignments are stable for this problem. Finally, by reordering the ranking of paths at node 4, we produce a specification called BAD GADGET (3 4 2 0) for node 3, yet it has the same unique solution as BAD GADGET. Fig. 3.

BGP Example (Bad bad bad)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>430</td>
</tr>
<tr>
<td>R3</td>
<td>130</td>
<td>20</td>
<td>30</td>
<td>430</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>-----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>R1</td>
<td>10</td>
<td>20</td>
<td>3420</td>
<td>420</td>
</tr>
<tr>
<td>R2</td>
<td>10</td>
<td>210</td>
<td>3420</td>
<td>420</td>
</tr>
<tr>
<td>R3</td>
<td>10</td>
<td>210</td>
<td>3420</td>
<td>420</td>
</tr>
<tr>
<td>R4</td>
<td>10</td>
<td>210</td>
<td>3420</td>
<td>420</td>
</tr>
<tr>
<td>R5</td>
<td>10</td>
<td>210</td>
<td>3420</td>
<td></td>
</tr>
<tr>
<td>R6</td>
<td>10</td>
<td>210</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>R7</td>
<td>130</td>
<td>210</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>R8</td>
<td>130</td>
<td>20</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>R9</td>
<td>130</td>
<td>20</td>
<td>30</td>
<td>420</td>
</tr>
<tr>
<td>R10</td>
<td>130</td>
<td>20</td>
<td>3420</td>
<td>420</td>
</tr>
<tr>
<td>R11</td>
<td>10</td>
<td>20</td>
<td>3420</td>
<td>420</td>
</tr>
</tbody>
</table>

NAUGHTY GADGET
Convergence

- If all AS policies follow Gao-Rexford rules,
  - Then BGP is guaranteed to converge (safety)

- For arbitrary policies, BGP may fail to converge!

- Why should this trouble us?
Performance Non-Issues

- Internal Routing
  - Domains typically use “hot potato” routing
  - Not always optimal, but economically expedient

- Policy not about performance
  - So policy-chosen paths aren’t shortest

- AS path length can be misleading
  - 20% of paths inflated by at least 5 router hops
- AS path length can be misleading
  - An AS may have many router-level hops

BGP says that path 4 1 is better than path 3 2 1
Performance: Real Issue

Slow Convergence

- BGP outages are biggest source of Internet problems
- Labovitz et al. *SIGCOMM’97*
  - 10% of routes available less than 95% of the time
  - Less than 35% of routes available 99.99% of the time
- Labovitz et al. *SIGCOMM 2000*
  - 40% of path outages take 30+ minutes to repair
- But most popular paths are very stable