The Global Internet
Big Picture of the Internet
The Global Internet and Inter-domain Routing

- Why does Border Gateway Protocol (BGP) exist?
  - What is interdomain routing and why do we need it?
  - Why does BGP look the way it does?

- How does BGP work?
  - Path vector algorithm
  - Various boring details

- pay more attention to the “why” than the “how”
Routing

- Provides paths between networks
- We know several designs already
  - link-state
  - distance vector
- But previous lectures assumed single “domain”
  - all routers have same routing metric (shortest path)
  - no privacy issues, no policy issues
Internet is more complicated.....

- Internet not just unstructured collection of networks
- Internet is comprised of a set of “autonomous systems” (ASes)
  - Independently run networks, some are commercial ISPs
  - Autonomy of control: ex: company, university, etc
  - Currently around 35,000 Ases
- Enables hierarchical aggregation of routing information
- ASes are sometimes called “domains”
  - hence “interdomain routing”
Autonomous Systems

- **Intradomain Routing (within an AS)**
  - Performed using domain-specific algorithm
  - Selected by domain administrators
  - Allows heterogeneous interior gateway protocols (IGP)

- **Interdomain Routing (between AS’s)**
  - Performed using standard global algorithm
  - Homogeneous exterior gateway protocol (EGP)
  - Main goal: reachability
Autonomous Systems

- Common intradomain routing protocols
  - Routing Information Protocol (RIP)
    - From the early Internet
    - Part of Berkeley Software Distribution (BSD) Unix
    - Distance vector algorithm
    - Based on hop count (infinity set to 16 hops)
  - Open Shortest Path First (OSPF)
    - Internet Standard (RFC 2328)
    - Link state algorithm
    - Authenticates messages
    - Load balances across links
Autonomous Systems

- Standard interdomain routing protocols
  - General aspects
    - Very complex and difficult
    - Focuses on reachability rather than optimality
    - Must be loop free
    - Specify how reachability information should be exchanged
  - Exterior Gateway Protocol (EGP)
    - Defined on Internet with tree structure
    - Embodied (and enforced) tree structure
    - Had to be replaced eventually
    - Distance vector updates
  - Border Gateway Protocol (BGP)
    - Replaced EGP
BGP

**Border routers**
- Connects an AS to the Internet
- Used for default external route
Autonomous Systems

Share connectivity information across ASes

table at R1:

<table>
<thead>
<tr>
<th>dest</th>
<th>next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R2</td>
</tr>
</tbody>
</table>

you can reach net A via me

traffic to A

AS1

AS2

R1

BGP

R2

R3

A

border router
internal router
In more detail: 4 Parts

1. Provide internal reachability (IGP)
2. Learn routes to external destinations (eBGP)
In more detail: 4 Parts

3. Distribute externally learned routes internally (iBGP)
4. Select closest egress (IGP)
The ‘A’ in AS really means Autonomous

- Want to choose their own internal routing protocol
  - Different algorithms and metrics

- Want freedom to route based on policy
  - “My traffic can’t be carried over my competitor’s network”
  - “I don’t want to carry transit traffic through my network”
  - Not expressible as Internet-wide “shortest path”!

- Want to keep their connections and policies private
  - Would reveal business relationships, network structure
AS’s are Businesses

- Three kinds of common 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

- Policy implications
  - “When sending traffic, I prefer to route through customers over peers, and peers over providers”
  - “I don’t carry traffic from one provider to another provider”
AS-Level Topology

- Destinations are IP prefixes (e.g., 12.0.0.0/8)
- Nodes are Autonomous Systems (ASes)
- Links are connections & business relationships
Autonomous Systems

Challenges
- **Scale**
  - Border router must be able to forward any packet destined anywhere in the Internet
- **Autonomous routing in AS’s**
  - Impossible to calculate meaningful costs for paths that cross multiple AS’s
- **Trust**
  - One AS must trust the advertised routes of other AS’s

Goal
- Specify polices that lead to “good” paths (even if they are not optimal)
Routing Choices

- Key issues are *policy* and *privacy*

- Challenges
  - No universal metric
  - AS-specific Policy decisions

- Problems with link state
  - Metric used by routers not the same
    - Can’t use shortest path - loops
  - LS database too large - entire Internet
  - Flooding may expose internal topology and policies to other AS’s
Routing Choices

- Key issues are *policy* and *privacy*
- Challenges
  - No universal metric
  - AS-specific Policy decisions
- Problems with distance-vector
  - Does not reveal any connectivity information
  - But still uses shortest path
  - Slow to converge
Solution: Path Vector Routing

- Extension of distance-vector routing
  - Support flexible routing policies
  - Faster loop detection (no 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
Path Vectors

- Each routing update carries the entire path
- Loops are detected as follows
  - When AS gets route check if AS already in path
    - If yes, reject route
    - If no, add self and (possibly) advertise route further
- Advantage
  - Metrics are local
  - AS chooses path, protocol ensures no loops
Loop Detection

- Node can easily detect a loop
  - Look for its own node identifier in the path
  - e.g., node 1 sees itself in the path “3, 2, 1”

- Node can simply discard paths with loops
  - e.g., node 1 simply discards the advertisement

```
3
```
```
2
```
```
1
```
```
“d: path (2,1)”
```
```
“d: path (3,2,1)”
```
```
“d: path (1)”
```
Flexible Policies

- Each node can apply local policies
  - Path selection: Which path to use?
  - Path export: Which paths to advertise?

- Examples
  - Node 2 may prefer the path “2, 3, 1” over “2, 1”
  - Node 1 may not let node 3 hear the path “1, 2”
AS Categories

- **Stub**
  - An AS that has only a single connection to one other AS - carries only local traffic

- **Multi-homed**
  - An AS that has connections to more than one AS, but does not carry transit traffic

- **Transit**
  - An AS that has connections to more than one AS, and carries both transit and local traffic (under certain policy restrictions)
AS Categories

Stub

Multi-homed

Transit

AS Categories
Issues with Path-Vector Policy Routing

- Reachability
- Security
- Performance
- Lack of isolation
- Policy oscillations
Reachability

- Normal routing
  - If graph is connected, reachability is assured

- Policy routing
  - Does not always hold

---

Diagram:

- AS 1 (Provider)
- AS 2 (Customer)
- AS 3 (Provider)

There is a direct connection between AS 1 and AS 3, but AS 1 cannot reach AS 3 via AS 2 due to a lack of a direct path.
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
- Even worse: snoop on all traffic to almost any destination
  - Without anyone realizing that anything is wrong
- Fixable: make ASes “prove” they have a path
  - But not used in today’s Internet
Performance

- BGP designed for policy not performance
- “Hot Potato” routing common but suboptimal
  - AS wants to hand off the packet as soon as possible
- Even BGP “shortest paths” are not shortest
  - Fewest AS’s != Fewest number of routers
- 20% of paths inflated by at least 5 router hops
- Not clear this is a significant problem
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
Lack of Isolation: Dynamics

- Change in the path
  - Path must be re-advertised to every node using the path
  - “Route Flap Damping” supposed to help (but ends up causing more problems)
Lack of isolation: Routing Table Size

- Each BGP router must know path to every other IP prefix
  - But router memory is expensive and thus constrained

- Number of prefixes growing more than linearly

- Subject of current research

Number of prefixes in BGP table
Persistent Oscillations due to Policies

Depends on the interactions of policies

We are back to where we started!
Policy Oscillations

- Policy autonomy vs network stability
  - Focus of much recent research

- Not an easy problem
  - Difficult to decide whether given policies will eventually converge!

- However, if policies follow normal business practices, stability is guaranteed
Border Gateway Protocol (BGP)

- Interdomain routing protocol for the Internet
  - Prefix-based path-vector protocol
  - Policy-based routing based on AS Paths
  - Evolved during the past 15 years

- 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’s job: maintain routing table

```plaintext
ner-routes>show ip bgp
BGP table version is 6128791, local router ID is 4.2.34.165
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* i3.0.0.0</td>
<td>4.0.6.142</td>
<td>1000</td>
<td>50</td>
<td>0 701 80 i</td>
<td></td>
</tr>
<tr>
<td>* i4.0.0.0</td>
<td>4.24.1.35</td>
<td>0</td>
<td>100</td>
<td>0 i</td>
<td></td>
</tr>
<tr>
<td>* i12.3.21.0/23</td>
<td>192.205.32.153</td>
<td>0</td>
<td>50</td>
<td>0 7018 4264 6468 ?</td>
<td></td>
</tr>
<tr>
<td>* e128.32.0.0/16</td>
<td>192.205.32.153</td>
<td>0</td>
<td>50</td>
<td>0 7018 4264 6468 25 e</td>
<td></td>
</tr>
</tbody>
</table>
```
BGP Operations

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
  - Stores all of the routes in a routing table
  - 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 neighbors

- Withdrawal
  - If the active route is no longer available
  - … send a withdrawal message to the neighbors
BGP Route Processing

Receive BGP Updates

Apply Import Policies

Best Route Selection

Best Route Table

Apply Export Policies

Install forwarding Entries for best routes.

IP Forwarding Table

Apply Policy = filter routes & tweak attributes

Based on Attribute Values

Best Routes

Apply Policy = filter routes & tweak attributes

Open ended programming.
Constrained only by vendor configuration language

Transmit BGP Updates

Apply Policy = filter routes & tweak attributes

Based on Attribute Values

Best Routes

Apply Policy = filter routes & tweak attributes
Selecting the best route

- **Route Attributes**
  - Set/modified according to operator instructions

- **Route Choices**
  - Compared based on attributes using (mostly) standardized rules
Joining BGP and IGP Information

- **Border Gateway Protocol (BGP)**
  - Announces reachability to external destinations
  - 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
    - 192.0.2.1 reached via 10.1.1.1
Summary

- BGP is essential to the Internet
  - ties different organizations together

- Poses fundamental challenges....
  - leads to use of path vector approach

- ...and myriad details