Interdomain Routing

EE 122, Fall 2013
Sylvia Ratnasamy

http://inst.eecs.berkeley.edu/~ee122/

Material thanks to Ion Stoica, Scott Shenker, Jennifer Rexford, and many other colleagues
Internet Routing

- So far, only considered routing within a domain.

- Many issues can be ignored in this setting because there is central administrative control over routers.
  - Issues such as autonomy, privacy, policy.

- But the Internet is more than a single domain.
Recall from Lecture 3

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

An “end-to-end” route

“Border Routers”

“Interior Routers”
Autonomous Systems (AS)

- AS is a network under a single administrative control
  - currently over 30,000 ASes
  - Think AT&T, France Telecom, UCB, IBM, etc.

- ASes are sometimes called “domains”.
  - Hence, “interdomain routing”

- Each AS is assigned a unique identifier
  - 16 bit AS Number (ASN)
Routing between ASes

Two key challenges

- Scaling

- Administrative structure
  - Issues of autonomy, policy, privacy
Recall From Lecture#4

- Assume each host has a unique ID
- No particular structure to those IDs
Recall Also...

Forwarding Table

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCB</td>
<td>4</td>
</tr>
<tr>
<td>UW</td>
<td>5</td>
</tr>
<tr>
<td>MIT</td>
<td>2</td>
</tr>
<tr>
<td>NYU</td>
<td>3</td>
</tr>
</tbody>
</table>
Scaling

- Every router must be able to forward packets to any destination
  - Given address, it needs to know “next hop” (table)

- Naive: Have an entry for each address
  - There would be over $10^8$ entries!
  - And routing updates per destination!

- Any ideas on how to improve scalability?
Scaling

- Every router must be able to forward based on *any* destination address
  - Given address, it needs to know “next hop” (table)
- Naive: Have an entry for each address
  - There would be $10^8$ entries!
  - And routing updates per destination!
- Better: Have an entry for a range of addresses
  - But can’t do this if addresses are assigned randomly

**Host addressing is key to scaling**
Two Key Challenges

- Scaling
- Administrative structure
  - Issues of autonomy, policy, privacy
Administrative structure shapes Interdomain 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”
  - Not expressible as Internet-wide “shortest path”!

- 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.
Choice of Routing Algorithm

Link State (LS) vs. Distance Vector (DV)?

- LS offers no privacy -- global sharing of all network information (neighbors, policies)
- LS limits autonomy -- need agreement on metric, algorithm
- DV is a decent starting point
  - per-destination advertisement gives providers a hook for finer-grained control over whether/which routes to advertise

The “Border Gateway Protocol” (BGP) extends distance-vector ideas to accommodate policy
Today

- Addressing

- BGP
  - today: context and key ideas
  - next lecture: details and issues
Addressing Goal: **Scalable Routing**

- **State:** Small forwarding tables at routers
  - Much less than the number of hosts

- **Churn:** Limited rate of change in routing tables
  - Traffic, inconsistencies, complexity

Ability to aggregate addresses is crucial for both
(one entry to *summarize* many addresses)
Aggregation only works if:

- Groups of destinations reached via the same path
- These groups are assigned contiguous addresses
- These groups are relatively stable
- Few enough groups to make forwarding easy
Hence, IP Addressing: Hierarchical

- Hierarchical address structure
- Hierarchical address allocation
- Hierarchical addresses and topology
IP Addresses (IPv4)

- Unique 32-bit number associated with a host
- Represented with the *dotted-quad* notation, e.g., 12.34.158.5:

```
12  34  158  5
  |   |   |   |
00001100 00100010 10011110 00000101
```
Examples

- What address is this? 80.19.240.51

  01010000 00010011 11110000 00110011

- How would you represent 68.115.183.7?

  01000100 01110011 10110111 00000111
Hierarchy in IP Addressing

- 32 bits are partitioned into a prefix and suffix components.
- Prefix is the network component; suffix is host component.

Interdomain routing operates on the network prefix.

Notation and terminology: 12.34.158.0/23 represents a “slash 23” network with a 23 bit prefix and $2^9$ host addresses.
History of Internet Addressing

- Always dotted-quad notation
- Always network/host address split
- But nature of that split has changed over time
Original Internet Addresses

- First eight bits: network address (/8)
- Last 24 bits: host address

Assumed 256 networks were more than enough!
Next Design: “Classful” Addressing

- Three main classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Network</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>Class B</td>
<td>1 10</td>
<td>16</td>
</tr>
<tr>
<td>Class C</td>
<td>1 110</td>
<td>24</td>
</tr>
</tbody>
</table>

- Class A: 126 networks, ~16M hosts
- Class B: ~16K networks, ~65K hosts
- Class C: ~2M networks, 254 hosts

Problem: Networks only come in three sizes!
Today’s Addressing: CIDR

- CIDR = Classless Interdomain Routing
- Idea: Flexible division between network and host addresses
- Motivation: offer a better tradeoff between size of the routing table and efficient use of the IP address space
CIDR (example)

- Suppose a network has fifty computers
  - allocate 6 bits for host addresses (since $2^5 < 50 < 2^6$)
  - remaining $32 - 6 = 26$ bits as network prefix
  - E.g., 128.23.9/26 is a “slash 26” network

- Flexible boundary between network and host bits means the boundary must be explicitly specified with the network address
  - informally, “slash 26” $\rightarrow$ 128.23.9/26
  - formally, represent length of prefix with a 32-bit mask: 256.256.256.192 where all network prefix bits set to “1” and host suffix bits to “0”
Classful vs. Classless addresses

- Example: an organization needs 500 addresses.
  - A single class C address not enough (254 hosts).
  - Instead a class B address is allocated. (~65K hosts)
  - That’s overkill, a huge waste!

- CIDR allows an arbitrary prefix-suffix boundary
  - Hence, organization allocated a single /23 address (equivalent of 2 class C’s)

- Maximum waste: 50%
Hence, IP Addressing: Hierarchical

- Hierarchical address structure
- Hierarchical address allocation
- Hierarchical addresses and routing scalability
Allocation Done Hierarchically

- Internet Corporation for Assigned Names and Numbers (ICANN) gives large blocks to...

- Regional Internet Registries (e.g., ARIN), which give blocks to
  - ARIN ➔ American Registry for Internet Numbers

- Large institutions (ISPs), which give addresses to...

- Individuals and smaller institutions

- FAKE Example:
  ICANN ➔ ARIN ➔ AT&T ➔ UCB ➔ EECS
CIDR: Addresses allocated in contiguous prefix chunks

Recursively break down chunks as get closer to host

12.0.0.0/8
  : 12.0.0.0/15
  : 12.2.0.0/16
  : 12.3.0.0/16
  : 12.253.0.0/19
  : 12.253.32.0/19
  : 12.253.64.0/19
  : 12.253.64.108/30
  : 12.253.96.0/18
  : 12.253.128.0/17

12.0.0.0/15
  : 12.3.0.0/22
  : 12.3.4.0/24
  : 12.3.254.0/23

12.2.0.0/16
  : 12.3.0.0/22
  : 12.3.4.0/24
  : 12.3.254.0/23

12.3.0.0/16
  : 12.3.0.0/22
  : 12.3.4.0/24
  : 12.3.254.0/23

12.253.0.0/16
ICANN gives ARIN several /8s
ARIN gives AT&T one /8, **12.0/8**
  - Network Prefix: **00001100**
AT&T gives UCB a /16, **12.197/16**
  - Network Prefix: **000011001100010100010111**
UCB gives EECS a /24, **12.197.45/24**
  - Network Prefix: **00001100110001010001011100101101**
EECS gives me a specific address **12.197.45.23**
  - Address: **00001100110001010010110100010111**
Hence, IP Addressing: Hierarchical

- Hierarchical address structure
- Hierarchical address allocation
- Hierarchical addresses and routing scalability
IP addressing → scalable routing?

- Hierarchical address allocation helps routing scalability if allocation matches topological hierarchy.
IP addressing → scalable routing?

France Telecom

a.b.*.* is this way

AT&T
a.0.0.0/8

a.c.*.* is this way

LBL
a.b.0.0/16

UCB
a.c.0.0/16
IP addressing → scalable routing?

Can add new hosts/networks without updating the routing entries at France Telecom

France Telecom

AT&T
a.0.0.0/8

LBL
a.b.0.0/16

UCB
a.c.0.0/16

foo.com
a.d.0.0/16

a.*.*.* is this way
ESNet must maintain routing entries for both a.*.*.* and a.c.*.*.
IP addressing → scalable routing?

- Hierarchical address allocation helps routing scalability if allocation matches topological hierarchy.

- Problem: may not be able to aggregate addresses for “multi-homed” networks.

- Two competing forces in scalable routing:
  - Aggregation reduces number of routing entries.
  - Multi-homing increases number of entries.

- Initial growth super-linear; no aggregation
- Advent of CIDR allows aggregation: linear growth
- Dot-com implosion; Internet bubble bursts
- Internet boom: multihoming drives superlinear growth
- Back in business
Same Table, Extended to Present

Stock Market Crash of 2008

What Happened Here?

Linear growth

Superlinear growth

What Happened Here?

Crash of 2008
Summary of Addressing

- **Hierarchical** addressing
  - Critical for **scalable** system
  - Don’t require everyone to know everyone else
  - Reduces amount of updating when something changes

- **Non-uniform** hierarchy
  - Useful for heterogeneous networks of different sizes
  - Class-based addressing was far too coarse
  - Classless InterDomain Routing (CIDR) more flexible

- A later lecture: impact of CIDR on router designs
Outline

- Addressing

- Border Gateway Protocol (BGP)
  - today: context and key ideas
  - next lecture: details and issues
BGP (Today)

- The role of policy
  - what we mean by it
  - why we need it

- Overall approach
  - four non-trivial changes to DV
  - how policy is implemented (detail-free version)
Administrative structure shapes Interdomain routing

- ASes want freedom to pick routes based on policy
- ASes want autonomy
- ASes want privacy
Topology and policy is shaped by the business relationships between ASes

- 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
Business Relationships

Relations between ASes
- provider <-> customer
- peer <-> peer

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

Relations between ASes
- provider ↔ customer
- peer ↔ 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
Routing Follows the Money!

ASes provide “transit” between their customers

Peers do not provide transit between other peers

traffic allowed  traffic not allowed
Routing Follows the Money!

- An AS only carries traffic to/from its own customers over a peering link
Routing Follows the Money!

- Routes are “valley free” (will return to this later)
In Short

- AS topology reflects business relationships between ASes
- Business relationships between ASes impact which routes are acceptable
- BGP Policy: Protocol design that allows ASes to control which routes are used
- Next lecture: more formal analysis of the impact of policy on reachability and route stability
BGP (Today)

- The role of policy
  - what we mean by it
  - why we need it

- Overall approach
  - four non-trivial changes to DV
  - how policy is implemented (detail-free version)
Interdomain 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: Basic Idea

An AS advertises ("exports") its best routes to one or more IP prefixes

Each AS selects the "best" route it hears advertised for a prefix

You’ve heard this story before!
BGP inspired by Distance Vector

- Per-destination route advertisements
- No global sharing of network topology information
- Iterative and distributed convergence on paths
- With four crucial differences!
**Differences between BGP and DV**

1. **not picking shortest path routes**

- BGP selects the best route based on policy, not shortest distance (least cost)

**Node 2 may prefer “2, 3, 1” over “2, 1”**

- How do we avoid loops?
Differences between BGP and DV

(2) path-vector routing

- Key idea: advertise the entire path
  - Distance vector: send *distance metric* per dest $d$
  - Path vector: send the *entire path* for each dest $d$
Differences between BGP and DV (2) path-vector routing

- Key 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
Loop Detection w/ 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”
  - E.g., node 1 simply discards the advertisement
Differences between BGP and DV (2) path-vector routing

- Key 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
Differences between BGP and DV

(3) Selective route advertisement

- For policy reasons, an AS may choose not to advertise a route to a destination
- Hence, reachability is not guaranteed even if graph is connected

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

(4) BGP may **aggregate** routes

- For scalability, BGP may aggregate routes for different prefixes
BGP (Today)

- The role of policy
  - what we mean by it
  - why we need it

- Overall approach
  - four non-trivial changes to DV
  - how policy is implemented (detail-free version)
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
  - make/save money (send to customer > peer > provider)
  - maximize performance (smallest AS path length)
  - minimize use of my network bandwidth (“hot potato”)
  - ...
  - ...

- BGP uses something called route “attributes” to implement the above (next lecture)
Typical Export: Peer-Peer Case

- Peers exchange traffic between their customers
  - AS exports only customer routes to a peer
  - AS exports a peer’s routes only to its customers
Typical Export: Customer-Provider

- Customer pays provider for access to Internet
  - Provider exports its customer routes to everybody
  - Customer exports provider routes only to its customers

Traffic to customer

Traffic from customer
Next Time.

- Gautam will introduce Project#1!
- Wrap up BGP
  - protocol details
  - pitfalls