Interdomain Routing

EECS 489 Computer Networks
http://www.eecs.umich.edu/~zmao/eecs489

Z. Morley Mao
Tuesday Sept 21, 2004
Distance Vector: Link Cost Changes

7 loop:
8 wait (until A sees a link cost change to neighbor V
9 or until A receives update from neighbor V)
10 if (D(A,V) changes by d)
11 for all destinations Y through V do
12 \[ D(A,Y) = D(A,Y) + d \]
13 else if (update D(V,Y) received from V)
14 \[ D(A,Y) = D(A,V) + D(V,Y) \];
15 if (there is a new minimum for destination Y)
16 send D(A,Y) to all neighbors
17 forever

“good news travels fast”
Distance Vector: Count to Infinity

Problem

1. **Loop:**
   - **Wait** (until $A$ sees a link cost change to neighbor $V$
   - or until $A$ receives update from neighbor $V$)
2. **If** ($D(A, V)$ changes by $d$)
3. **For all** destinations $Y$ through $V$ do
5. **Else if** (update $D(V, Y)$ received from $V$)
7. **If** (there is a new minimum for destination $Y$)
8. **Send** $D(A, Y)$ to all neighbors
9. **Forever**

**Node B**

<table>
<thead>
<tr>
<th></th>
<th>D</th>
<th>C</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>4</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>$C$</td>
<td>1</td>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>

**Node C**

<table>
<thead>
<tr>
<th></th>
<th>D</th>
<th>C</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>5</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>$B$</td>
<td>1</td>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>

**Link cost changes here:** recall from slide 24 that $B$ also maintains shortest distance to $A$ through $C$, which is 6. Thus $D(B, A)$ becomes 6!

“bad news travels slowly”
Distance Vector: Poisoned Reverse

- If C routes through B to get to A:
  - C tells B its (C’s) distance to A is infinite (so B won’t route to A via C)
  - Will this completely solve count to infinity problem?

**Node B**

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>

**Node C**

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>

**Link cost changes here**; B updates $D(B, A) = 60$ as C has advertised $D(C, A) = 8$

**Algorithm terminates**
Link State vs. Distance Vector

Per-node message complexity
- **LS**: $O(e)$ messages
  - $e$: number of edges
- **DV**: $O(d)$ messages, many times
  - $d$: node’s degree

Complexity/Convergence
- **LS**: $O(n^2)$ computation
- **DV**: convergence time varies
  - may be routing loops
  - count-to-infinity problem

Robustness: what happens if router malfunctions?
- **LS**:
  - node can advertise incorrect *link* cost
  - each node computes only its *own* table
- **DV**:
  - node can advertise incorrect *path* cost
  - each node’s table used by others; error propagate through network
Are We Done?

- We now know how to route scalably
- What more is there to do?
Issues We Haven’t Addressed

- Scaling
  - Addressing
  - Router table size

- Structure
  - Autonomy
  - Policy
Scaling

- Every router must be able to forward based on *any* destination IP address
  - Given address, it needs to know “next hop” (table)
- Naive: Have an entry for each address
  - There would be $10^8$ entries!
- Better: Have an entry for a range of addresses
  - But can’t do this if addresses are assigned randomly!
- Addresses allocation is a big deal
The Internet contains a large number of diverse networks.
Autonomous Systems (AS)

- Internet is not a single network!

- The Internet is a collection of networks, each controlled by different administrations

- An autonomous system (AS) is a network under a single administrative control
Implications

- ASs want to choose own local routing algorithm
  - AS takes care of getting packets to/from their own hosts
  - Interdomain routing and Intradomain routing

- ASs want to choose own nonlocal routing policy
  - Interdomain routing must accommodate this
  - BGP is the current interdomain routing protocol
Intradomain And Interdomain

Diagram showing network domains and protocols:
- IGRP A
- BGP
- OSPF
- RIP
- IntraDomain
- InterDomain

Nodes and connections indicate different domains and protocols.
Interconnection

- IP unifies network **technologies**
  - allows any network to communicate with another

- BGP unifies network **organizations**
  - ties them into a global Internet
OSPF (Open Shortest Path First)

- “open”: publicly available
- Uses Link State algorithm
  - LS packet dissemination
  - Topology map at each node
  - Route computation using Dijkstra’s algorithm

- OSPF advertisement carries one entry per neighbor router
- Advertisements disseminated to entire AS (via flooding)
  - Carried in OSPF messages directly over IP (rather than TCP or UDP)
OSPF “advanced” features (not in RIP)

- **Security**: all OSPF messages authenticated (to prevent malicious intrusion)
- **Multiple same-cost paths** allowed (only one path in RIP)
- For each link, multiple cost metrics for different **TOS** (e.g., satellite link cost set “low” for best effort; high for real time)
- **Integrated uni- and multicast** support:
  - Multicast OSPF (MOSPF) uses same topology data base as OSPF
- **Hierarchical** OSPF in large domains.
Hierarchical OSPF
Hierarchical OSPF

- **Two-level hierarchy:** local area, backbone.
  - Link-state advertisements only in area
  - each node has detailed area topology; only know direction (shortest path) to nets in other areas.

- **Area border routers:** “summarize” distances to nets in own area, advertise to other Area Border routers.

- **Backbone routers:** run OSPF routing limited to backbone.

- **Boundary routers:** connect to other AS’s.
Outline

- Addressing
- BGP
Assigning Addresses (Ideally)

- Host: gets IP address from its organization or ISP
- Organization: gets IP address block from ISP
- ISP: gets address block from routing registry:
  - ARIN: American Registry for Internet Numbers
  - RIPE: Reseaux IP Europeens
  - APNIC: Asia Pacific Network Information Center
  - Whois –h whois.arin.net <IP>
- Each AS is assigned a 16-bit number (65536 total)
  - Currently 10,000 AS’s in use
  - Most stub, so don’t really need own number
Original Addressing Scheme

- Class-based addressing schemes:
  - 32 bits divided into 2 parts:
    - Class A
      - Network: 0
      - Host: 8
      - 126 nets
      - ~16M hosts
    - Class B
      - Network: 1 0
      - Host: 16
      - ~16K nets
      - ~65K hosts
    - Class C
      - Network: 1 1 0
      - Host: 24
      - ~2M nets
      - 254 hosts

Original Vision:
- Route on network number
- All nodes with same net # are directly connected
Classless Interdomain Routing (CIDR)

Introduced to solve two problems:

- exhaustion of IP address space
- size and growth rate of routing table
#1: Address Space Exhaustion

- 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: networks assigned on arbitrary bit boundaries.
  - Requires explicit masks to be passed in routing protocols
  - Masks: identify the “network” portion of the address

- CIDR solution for example above: organization is allocated a single /23 address (equivalent of 2 class C’s).
CIDR Addressing

- Suppose fifty computers in a network are assigned IP addresses 128.23.9.0 - 128.23.9.49
  - They share the **prefix** 128.23.9

- **Range:** 01111111 00001111 00001001 00000000 to
  
  
  01111111 00001111 00001001 00110001

  - How to write 01111111 00001111 00001001 00X?

- **Convention:** 128.23.9.0/26
  - There are 32-26=6 bits for the 50 computers
  - $2^6 = 64$ addresses
Specify a range of addresses by a prefix: $X/Y$
- The common prefix is the first $Y$ bits of $X$.
- $X$: The first address in the range has prefix $X$
- $Y$: $2^{32-Y}$ addresses in the range

Example 128.5.10/23
- Common prefix is 23 bits:
  - 01000000 00000101 0000101
- Number of addresses: $2^9 = 512$

Prefix aggregation
- Combine two address ranges
  - 128.5.10/24 and 128.5.11/24 gives 128.5.10/23

Routers match to longest prefix
Problem #2: Routing Table Size

Without CIDR:

232.71.0.0
232.71.1.0
232.71.2.0
......
232.71.255.0

service provider

232.71.0.0
232.71.1.0
232.71.2.0
......
232.71.255.0

Global internet

With CIDR:

232.71.0.0
232.71.1.0
232.71.2.0
......
232.71.255.0

service provider

232.71.0.0/16

Global internet
Border Gateway Protocol

ignore the details
pay attention to the “why”
Internet inter-AS routing: BGP

- **BGP (Border Gateway Protocol):** *the de facto standard*

- **BGP provides each AS a means to:**
  1. Obtain subnet reachability information from neighboring ASs.
  2. Propagate the reachability information to all routers internal to the AS.
  3. Determine “good” routes to subnets based on reachability information and policy.

- **Allows a subnet to advertise its existence to rest of the Internet:** “*I am here*”
Who speaks BGP?

- Two types of routers
  - Border router (Edge), Internal router (Core)
Purpose of BGP

Share connectivity information across ASes
I-BGP and E-BGP

IGP: Interior Gateway Protocol. Example: OSPF

- **R** border router
- **internal router**
Issues

- What basic routing algorithm should BGP use?
- How are the routes advertised?
- How are routing policies implemented?
  - policy routing: not always shortest path
- Peering, transit, and reachability
Choice of Routing Algorithm

- **Constraints:**
  - scaling
  - autonomy (policy and privacy)

- **Link-state?**
  - requires sharing of complete network information
  - information exchanges don’t scale
  - can’t express policy

- **Distance Vector?**
  - scales and retains privacy
  - can’t implement policy
  - can’t avoid loops if shortest paths not taken
Path Vector Protocol

- Distance vector algorithm with extra information
  - For each route, store the complete path (ASs)
  - No extra computation, just extra storage

- Advantages:
  - can make policy choices based on set of ASs in path
  - can easily avoid loops
### BGP 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</td>
<td>701 80 i</td>
</tr>
<tr>
<td>* i4.0.0.0</td>
<td>4.24.1.35</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>i</td>
</tr>
<tr>
<td>* i12.3.21.0/23</td>
<td>192.205.32.153</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>7018 4264 6468 ?</td>
</tr>
<tr>
<td>* e128.32.0.0/16</td>
<td>192.205.32.153</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>7018 4264 6468 25 e</td>
</tr>
</tbody>
</table>
```

- Every route advertisement contains the entire AS path
- Can implement policies for choosing best route
- Can detect loops at an AS level
Advertising Routes

- One router can participate in many BGP sessions.
- *Initially* ... node advertises ALL routes it wants neighbor to know (could be > 50K routes)
- *Ongoing* ... only inform neighbor of changes
A, B, C are provider networks
X, W, Y are customer (of provider networks)
X is dual-homed: attached to two networks
- X does not want to route from B via X to C
- ... so X will not advertise to B a route to C
BGP routing policy (2)

- A advertises to B the path AW
- B advertises to X the path BAW
- Should B advertise to C the path BAW?
  - No way! B gets no “revenue” for routing CBAW since neither W nor C are B’s customers
  - B wants to force C to route to w via A
  - B wants to route *only* to/from its customers!
Why different Intra- and Inter-AS routing?

- **Policy:**
  - Inter-AS: admin wants control over how its traffic routed, who routes through its net.
  - Intra-AS: single admin, so no policy decisions needed

- **Scale:**
  - hierarchical routing saves table size, reduced update traffic

- **Performance:**
  - Intra-AS: can focus on performance
  - Inter-AS: policy may dominate over performance
Basic Messages in BGP

- **Open**: 
  - Establishes BGP session (uses TCP port #179)
  - BGP uses TCP

- **Notification**: 
  - Report unusual conditions

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

- **Keepalive**: 
  - Inform neighbor that connection is still viable
Routes Have Attributes

- When a route is “advertised” it is described in terms of attributes:
  - next hop, AS-path, etc.
  - We will discuss: Origin, MED, Local Preference

- Origin:
  - Who originated the announcement? Where was a prefix *injected* into BGP?
  - IGP, EGP or Incomplete (often used for static routes)
Multi-Exit Discriminator (MED)

- When AS’s interconnected via 2 or more links
- AS announcing prefix sets MED (AS2 in picture)
- AS receiving prefix uses MED to select link
- A way to specify how close a prefix is to the link it is announced on
Local Preference

- Used to indicate preference among multiple paths for the same prefix *anywhere* in the Internet.
- The higher the value the more preferred
- Exchanged between IBGP peers only. Local to the AS.
- Often used to select a specific exit point for a particular destination

**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>
Choosing Best Route

- Choose route with highest `LOCAL_PREF`
  - Preference-based routing
- Multiple choices: select route with shortest `hop-count`
- Multiple choices for same neighboring AS: choose path with min MED value
- Choose route based on lowest origin type
  - IGP < EGP < INCOMPLETE
- Among IGP paths, choose one with lowest cost
- Finally use router ID to break the tie.
Routing Process Overview

Routes received from neighbors

Import Policy Engine

Decision process

Choose best route

Routes used by router

BGP table

IP routing table

Export Policy Engine

forward, not forward set MEDs

Routes sent to neighbors

accept, deny, set preferences

Mao F04 44
Import and Export Policies

- **Inbound** filtering controls **outbound** traffic
  - Filters route updates received from other peers
  - Filtering based on IP prefixes, AS_PATH, community

- **Outbound** filtering controls **inbound** traffic
  - *Forwarding a route* means others may choose to reach the prefix through you
  - *Not forwarding a route* means others must use another router to reach the prefix
Transit vs. Nontransit AS

Transit traffic = traffic whose source and destination are outside the AS

**Nontransit AS**: does not carry transit traffic
- Advertise own routes only
- Do not propagate routes learned from other AS’s

**Transit AS**: does carry transit traffic
- Advertises its own routes PLUS routes learned from other AS’s
AS Relationships and Export Rules

- AS has customers, providers and peers
- Relationships between AS pairs:
  - Customer-provider
  - Peer-to-peer
- Type of relationship influences policies

- Exporting to provider:
  AS exports its routes & its customer’s routes, but not routes learned from other providers or peers
- Exporting to peer:
  (same as above)
- Exporting to customer:
  AS exports its routes plus routes learned from its providers, peers and other customers
- Assume that the small ISP is a customer of two large ISPs
- If customer ISP does not obey export rules
  - forwards advertisements from one large ISP to another
  - Carries huge volume of transit traffic between two large ISPs
Is Reachability Guaranteed?

- In normal routing, if graph is connected then reachability is assured

- With policy routing, not always
Peering & Transit

- **Peering**
  - The business relationship whereby ISPs reciprocally provide to each other connectivity to each others’ transit customers

- **Transit (customer)**
  - The business relationship whereby one ISP provides (usually sells) access to all destinations in its routing table
West and East Peer with USNet but they can’t reach each other
Transit

By EastNet purchasing transit, EastNet is announced by USNet to USNet Peering and Transit interconnections alike.

...for a (transit) fee of course.

Figure from William B. Norton, “Internet Service Providers and Peering”
Reachability?

- Depends on series of economic relationships
- Not a technical solution, but an economic one
BGP and Performance

- BGP designed for policy not performance
  - Hot Potato routing common but suboptimal
  - 20% of internet paths inflated by at least 5 router hops

- Susceptible to router misconfiguration
  - Blackholes: announce a route you cannot reach

- Incompatible policies
  - Solutions to limit the set of allowable policies
Research Aside

- What is one of the most important Internet design principles?

- In what way is BGP robust?
  - Can it protect against node failures?
  - Can it protect against spoofing?
  - Can it protect against lying?

- Semantic failures versus syntactic failures
  - BGP can check who you are, but not what you say