Lecture 6

Link-State and Distance-Vector

- Attend section!
    - Review Dijkstra's
    - DV data-structures in detail
    - When poison-reverse fails

Routing in the Internet

- **Autonomous Systems (AS) or Domain**: Region of a network under a single administrative entity
- **Border Router**: Has connection with another AS
- **Interior Routers**: All connections are within the domain

Autonomous Systems (AS)

- A network under a single administrative control
    - Currently over 30,000 ASes
    - AT&T, France Telecom, UCB, IBM, etc.
- Each AS is assigned a unique identifier
    - 16 bit AS Number (ASN)
    - E.g., ASN 25 is UCB

Intradomain routing: within an AS

- Link-State (OSPF) and Distance-Vector (RIP, IGRP)
- Focus
    - "Least cost" paths

    - Convergence

Interdomain 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

Scaling

- A router must be able to reach **any** destination
    - Given packet's destination address, lookup "next hop"
- Naive: Have an entry for each destination
    - There would be over 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!

- How addresses are allocated will matter!!
- **How addressing is key to scaling**

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 “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.

Choice of Routing Algorithm

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

- LS offers no privacy: broadcasts all network information
- LS limits autonomy: need agreement on metric, algorithm
- DV is a decent starting point
    - Per-destination updates by intermediate nodes give us a hook
    - But wasn’t designed to implement policy
    - And is vulnerable to loops if shortest paths not taken
- **The “Border Gateway Protocol” (BGP extends distance-vector ideas to accommodate policy**

Outline

- Addressing
- BGP
    - context and basic ideas: today
    - details and issues: next lecture

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
- 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 routing scalability

IP Addresses (IPv4)

- Unique 32-bit number associated with a host

- 00001100 00100010 10011110 00000101
- Represented with the ”dotted quad” notation
  - e.g. 12.34.158.5

Hierarchy in IP Addressing

- 32 bits are partitioned into a prefix and suffix components
- Prefix is the **network component**; suffix is **host component**
- Network (23 bits) + Host (9 bits)
- Interdomain routing operates on the network prefix

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
- Maximum waste: 50%

CIDR (example)

- Suppose a network has fifty computers
  - Allocate 6 bits for host addresses ($2^5 < 50 < 2^6$)
  - Remaining 32 - 6 = 26 bits as network prefix
- Flexible boundary means the boundary must be explicitly specified with the network address!
  - informally , **slash 26** -> 128.23.9/26
  - formally, prefix represented with a 32-bit **mask**: 255.255.255.192 where all network prefix bits set to **1** and host

suffix bits to **0**

Allocation Done Hierarchically

- Internet Corporation for Assigned Names and Numbers (ICANN) gives large blocks to…
- Regional Internet Registries, such as the American Registry for Internet Names (ARIN), which give blocks to…
- Large Institutions (ISPs), which give addresses to…
- Individuals and smaller institutions
- FAKE example:
    - ICANN -> ARIN -> AT&T -> UCB -> EECS

FAKE Example in More Detail

- CANN 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: 0000110011000101
- UCB gives EECS a /24, 12.197.45/24
    - Network Prefix: 000011001100010100101101
- EECS gives me a specific address 12.197.45.23
- Address: 00001100110001010010110100010111

IP addressing -> scalable routing?

- Hierarchical address allocation only 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

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