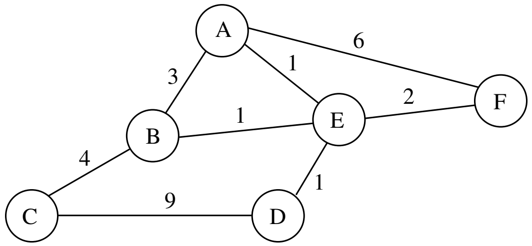
Routing

- Outline
  - Algorithms
  - Scalability

Overview

- Forwarding versus Routing
  - forwarding: to select an output port based on destination address and routing table
  - routing: process by which routing table is built
- Network as a Graph



- Problem: Find the lowest cost path between any two nodes
- Factors
  - static: topology
  - dynamic: load

Distance Vector

- Each node maintains a set of triples:

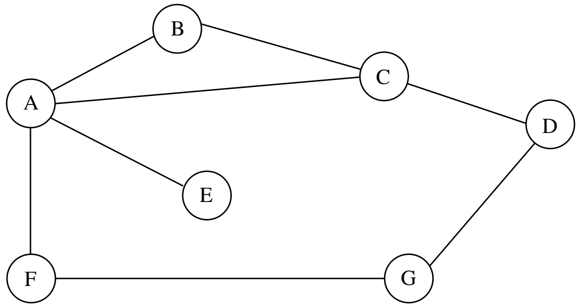
  `(Destination, Cost, NextHop)`

- Each node sends updates to (and receives updates from) its directly connected neighbors
  - periodically (on the order of several seconds)
  - whenever its table changes (called *triggered* update)
- Each update is a list of pairs:

  `(Destination, Cost)`

- Update local table if receive a "better" route
  - smaller cost
  - came from next-hop
- Refresh existing routes; delete if they time out

Example



Routing table at node B

Destination	Cost	NextHop
A	1	A
C	1	C
D	2	C
E	2	A
F	2	A
G	3	A

Routing Loops

- Example 1
  - F detects that link to G has failed
  - F sets distance to G to infinity and sends update to A
  - A sets distance to G to infinity since it uses F to reach G
  - A receives periodic update from C with 2-hop path to G
  - A sets distance to G to 3 and sends update to F
  - F decides it can reach G in 4 hops via A
- Example 2
  - Link from A to E fails
  - A advertises distance of infinity to E
  - B and C advertise a distance of 2 to E
  - B decides it can reach E in 3 hops; advertises this to A
  - A decides it can reach E in 4 hops; advertises this to C
  - C decides that it can reach E in 5 hops......
- Heuristics to break routing loops
  - set infinity to 16
  - split horizon
  - split horizon with poison reverse

Link State

- Strategy: Send to all nodes (not just neighbors) information about directly connected links (not entire routing table).
- Link State Packet (LSP)
  - id of the node that created the LSP
  - cost of link to each directly connected neighbor
  - sequence number (SEQNO)
  - time-to-live (TTL) for this packet
- Reliable Flooding
  - store most recent LSP from each node
  - forward LSP to all nodes but one that sent it
  - generate new LSP periodically; increment SEQNO
  - start SEQNO at 0 when reboot
  - decrement TTL of each stored LSP; discard when TTL=0

Route Calculation

- Dijkstra's shortest path algorithm
- Let
  - N denotes set of nodes in the graph
  - $l(i,j)$ denotes non-negative cost (weight) for edge (i,j)
  - $s \in N$ denotes this node
  - M denotes the set of nodes incorporated so far
  - $C(n)$ denotes cost of the path from s to node n
- Then

$M = \{s\}$
`for each` n `in` $N - \{s\}$
  $C(n) = l(s,n)$
`while` $(N \neq M)$
  $M = M$ `union` $\{w\}$ `such that` $C(w)$
  `is the minimum for all` w `in` $(N - M)$
  `for each` n `in` $(N - M)$
    $C(n) =$ `MIN`$(C(n), C(w) + l(w,n))$

Metrics

- Original ARPANET metric
  - measured number of packets enqueued on each link
  - took neither latency or bandwidth into consideration
- New ARPANET metric
  - stamp each incoming packet with its arrival time (`AT`)
  - record departure time (`DT`)
  - when link-level ACK arrives, compute

```
Delay = (DT - AT) + Transmit + Latency
```

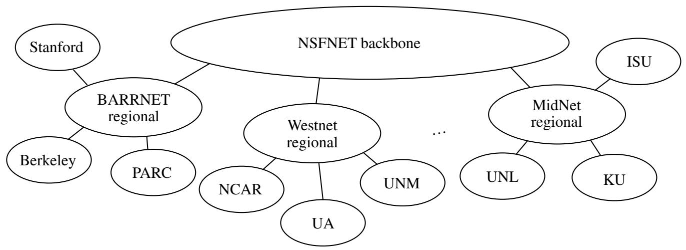
  - if timeout, reset `DT` to departure time for retransmission
  - link cost = average delay over some time period
- Fine Tuning
  - compressed dynamic range
  - replaced delay with link utilization

How to Make Routing Scale

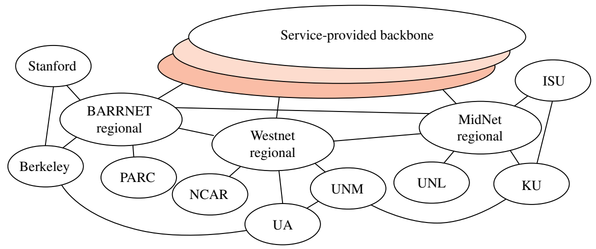
- Flat versus Hierarchical Addresses
- Inefficient use of Hierarchical Address Space
  - class C network with 2 hosts (2/255 = 0.78% efficient)
  - class B network with 256 hosts (256/65535 = 0.39% efficient)
- Still Too Many Networks
  - routing tables do not scale
  - route propagation protocols do not scale

Internet Structure

Recent Past

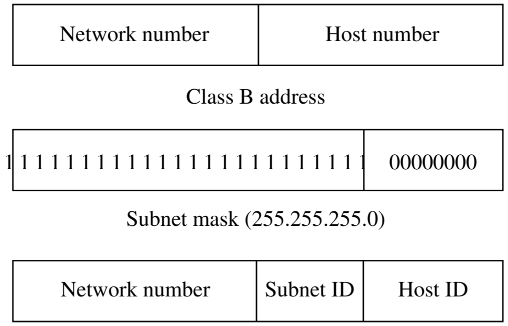


Today



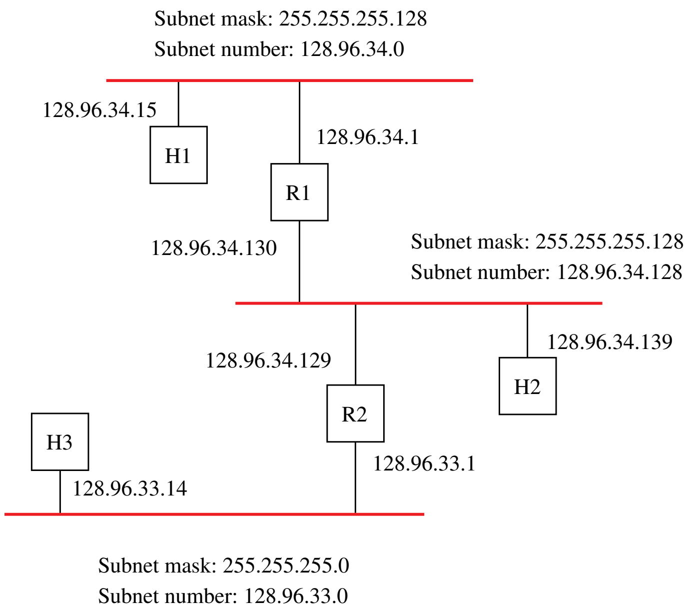
Subnetting

- Add another level to address/routing hierarchy: *subnet*
- *Subnet masks* define variable partition of host part of class A and B addresses
- Subnets visible only within site



Subnetted address

Subnet Example



Forwarding table at router R1

Subnet Number	Subnet Mask	Next Hop
128.96.34.0	255.255.255.128	interface 0
128.96.34.128	255.255.255.128	interface 1
128.96.33.0	255.255.255.0	R2

Forwarding Algorithm

```
D = destination IP address
for each entry ⟨SubnetNum, SubnetMask, NextHop⟩
   D1 = SubnetMask & D
   if D1 = SubnetNum
      if NextHop is an interface
         deliver datagram directly to destination
      else
         deliver datagram to NextHop (a router)
```

- Use a default router if nothing matches
- Not necessary for all ones in subnet mask to be contiguous
- Can put multiple subnets on one physical network
- Subnets not visible from the rest of the Internet

Supernetting

- Assign block of contiguous network numbers to near-by networks
- Called CIDR: Classless Inter-Domain Routing
- Represent blocks with a single pair

  ⟨ `first_network_address, count` ⟩

- Restrict block sizes to powers of 2
- Use a bit mask (CIDR mask) to identify block size
- All routers must understand CIDR addressing

Route Propagation

- Know a smarter router
  - hosts know local router
  - local routers know site router
  - site routers know core router
  - core routers know everything
- Autonomous System (AS)
  - corresponds to an administrative domain
  - examples: University, company, backbone network
  - assign each AS a 16-bit number
- Two-level route propagation hierarchy
  - interior gateway protocol (each AS selects its own)
  - exterior gateway protocol (Internet-wide standard)

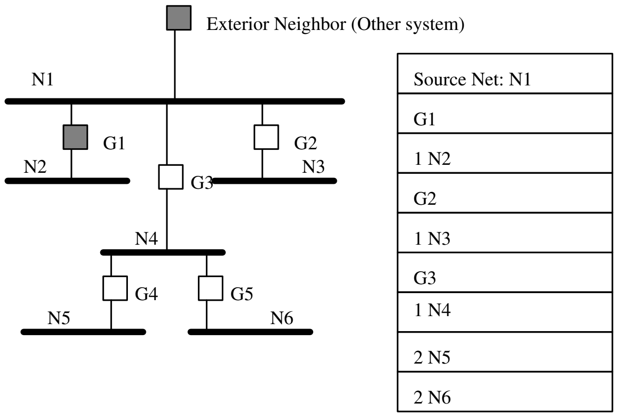
Popular Interior Gateway Protocols

- RIP: Route Information Protocol
  - developed for XNS
  - distributed with Unix
  - distance-vector algorithm
  - based on hop-count
- OSPF: Open Shortest Path First
  - recent Internet standard
  - uses link-state algorithm
  - supports load balancing
  - supports authentication

EGP: Exterior Gateway Protocol

- Overview
  - designed for tree-structured Internet
  - concerned with *reachability*, not optimal routes
- Protocol messages
  - neighbor acquisition: one router requests that another be its peer; peers exchange reachability information
  - neighbor reachability: one router periodically tests to see if the other router is still reachable; exchange HELLO/ACK messages; uses a k-out-of-n rule
  - routing updates: peers periodically exchange their routing tables (distance-vector)

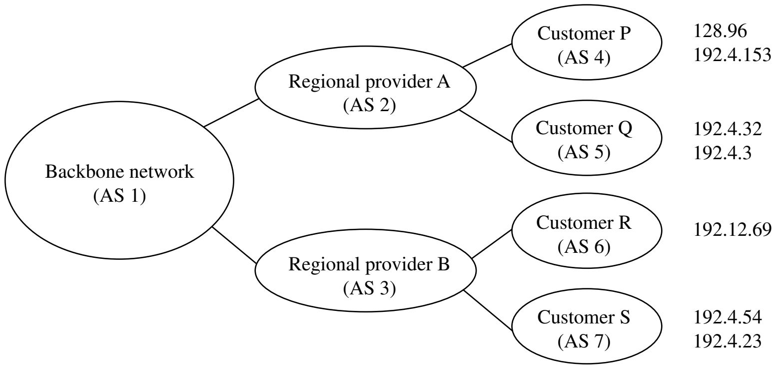
EGP Example



BGP-4: Border Gateway Protocol

- AS Types
  - Stub AS: an AS that has only a single connection to one other AS; carry local traffic only.
  - Multihomed AS: an AS that has connections to more than one other AS, but refuses to carry transit traffic.
  - Transit AS: an AS that has connections to more than one other AS, and is designed to carry both transit and local traffic.
- Each AS has:
  - One or more border routers
  - One BGP *speaker* that advertises:
    - local networks
    - other reachable networks (transit AS only)
    - gives *path* information

BGP Example



- Speaker for AS 2 advertises reachability to P and Q

  Network 128.96, 192.4.153, 192.4.32, and 192.4.3, can be reached directly from AS 2.

- Speaker for backbone network then advertises

  Networks 128.96, 192.4.153, 192.4.32, and 192.4.3 can be reached along the path ⟨AS 1, AS 2⟩.

- Speaker can also cancel previously advertised paths

IP Version 6

- Features
  - 128-bit addresses (classless)
  - Multicast
  - Real-time service
  - Authentication and security
  - Autoconfiguration
  - End-to-end fragmentation
  - Protocol extensions
- Header
  - 40-byte “base” header
  - Extension headers (fixed order, mostly fixed length)
    - fragmentation
    - source routing
    - authentication and security
    - other options