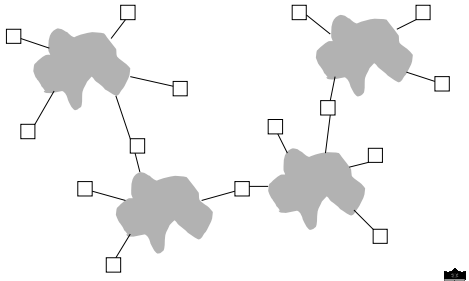
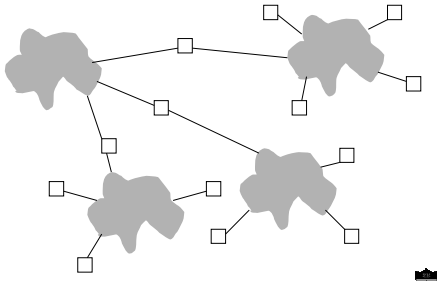
Routing



Typical Structure



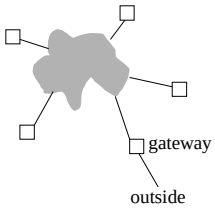
Routing: Goals

- avoid long paths
- avoid congested paths
- adapt to changes in topology
- avoid centralized computation
- keep memory requirements low

Routing Tables

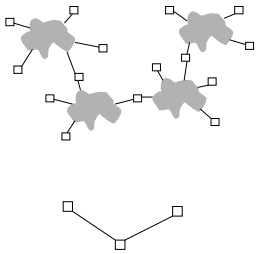
- each host has a routing table
- table says where to forward packets destined for each network
  - forward only to directly connected hosts
- Where do routing tables come from?

Simple Gateways



- ordinary hosts send all packets destined for outside to the gateway machine
- only gateway participates in the routing protocols

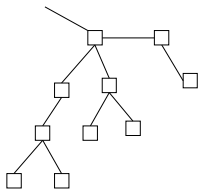
Routing as a Graph Problem



- each router becomes a vertex
- each LAN link between routers becomes an edge
- can label edges with "cost" of transmitting along that link

Hierarchical Routing

- if part of graph is "tree-like"
  - each router needs to know only about the subtree below it
  - all traffic in or out of the tree is sent through the root
  - rest of network sees tree as collapsed to a single node
- can make graph tree-link by deciding to ignore some links



Link-State Routing

- three stages:
  - estimate "cost" for each link
    - » based on bandwidth, congestion, reliability
  - communicate graph structure to all routers
  - each router solves shortest-path problem on graph
- good: simple, gets the right answer
- bad: scales poorly as routers are added

Distance-Vector Routing

- iterative, distributed algorithm
- nodes communicate with their neighbors, advertising available routing options
- useful information diffuses through the graph

destination	cost	next hop
A	13	A
B	19	A
C	3	D
D	1	D
E	inf	-

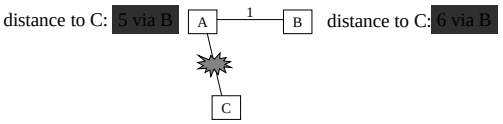
Distance-Vector Algorithm

- start with cost and next-hop for direct links
  - all other costs set to infinity
- periodically, (dest, cost) pairs to neighbors
- on receiving (dest, cost) from a neighbor, check whether it is a better path than the current one
  - it's better if (cost to neighbor)+(neighbor's cost to dest) < (my cost to dest)

Behavior of Distance-Vector

- if network never changes, algorithm converges to optimum
  - time goes as O(longest path in hops)
- if costs change or nodes are added, algorithm re-converges to new optimum
- if nodes or links fail, bad things can happen
  - "count to infinity"

Count-to-Infinity Problem



solution: ~~detect back-and-forth pattern~~
misses cycles with 3 or more links

Routing in the Real Internet

- two-level approach
- divide Net into Autonomous Systems (AS)
  - each AS within one organization
- routing within AS done locally
  - any algorithm is OK
- routing between ASes uses BGP protocol

Routing within an AS

- many approaches are viable
  - static routing tables
  - RIP (Routing Information Protocol) uses distance-vector approach
  - OPSF (Open Shortest Path First) uses link-state approach, with some optimizations

Inter-AS Routing

- use Boundary Gateway Protocol (BGP)
  - each AS designates a single “BGP speaker” to represent it in the protocol
  - speakers advertise which ASes they can reach
    - » also include path
  - like distance vector, but everybody knows the paths
    - » each AS can choose paths by its own criteria
    - » routing cycles avoided

Classless Routing

- Inter-AS routing requires state for each network
  - uses too much space as more networks added
- solution: aggregate routing information
- example: if all nets whose numbers are 10110011010111xx use the same route, store only one record

Classless Routing

- generalization: treat network number as having a variable number of bits
- saves space in routing tables
- allows variable number of hosts in a network, so uses network numbers more efficiently
- but we’ll still run out of IP addresses before too long

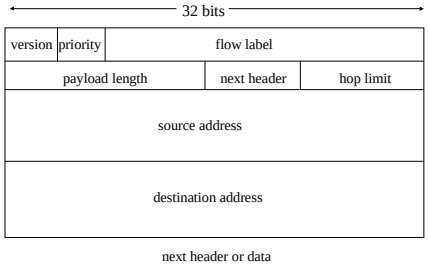
IPv6

- upcoming version of IP
- features
  - 128-bit addresses
  - better control over routing offered to end hosts
  - security (crypto)
  - clean up mistakes in IPv4
  - extensible (can add options)
  - can coexist with IPv4
    - » expect ten-year transition period

IPv6 Addressing

- 128-bit host addresses
- allocation of possible addresses
  - 72% reserved for later use
  - provider-allocated range
  - geographically-allocated range
  - local-use-only addresses
  - wrapped IPv4 addresses
  - multicast addresses

IPv6 Header Format



IPv6 Header Fields

- version: set to 6
- priority: 0-15
  - higher priority gets preferential service
  - 7 or lower means endpoints can slow down
- flow label
  - experimental feature allows different treatment of different data flows between a pair of hosts

IPv6 Header Fields

- payload length: excludes header
- hop limit: decremented when packet is forwarded; packet dropped if this reaches 0
- next header: either
  - which of the defined extension headers comes next, or
  - which higher-level protocol should get this packet

Header Extensions

- six currently defined (hop-by-hop options, routing, fragmentation, authentication, encrypted security payload, destination options)
  - more will be defined later (probably)
- must appear in a fixed order
- each has its own format

Differences with IPv4

- fragmentation done by sender only
  - too-large fragment causes ICMP error message back to sender; sender can refragment and try again
- no checksum
  - higher level protocol should have one
- support for crypto and authentication

Transition: From IPv4 to IPv6

- people will switch little by little
  - might take ten years
- during transition, the two protocols will coexist
  - hosts and routers can run two-headed IP software
  - IPv6 packets can “tunnel” through IPv4 regions