Routing

Digital Communications II

Michaelmas Term 2007
Based on Prof. Jon Crowcroft’s notes, and thus transitively on
S. Keshav’s “An Engineering Approach to Computer Networking”

What is it?

- Process of finding a path from a source to every destination in
  the network
- Suppose you want to connect to Antarctica from your desktop
  - What route should you take?
  - Does a shorter route exist?
  - What if a link along the route goes down?
  - What if you’re on a mobile wireless link?
- Routing deals with these types of issues

Moving from DigiComms I

- DigiComms I set up the key principles:
  - Routing is the binding of addresses to paths
  - Can be static or dynamic
  - Can be centralised or distributed
- Our focus in DigiComms II
  - What technologies do routing today?
    - (with simplifications!!)
  - How do these technologies work?
    - (or not)

Basics

- A routing protocol sets up a routing table in routers and switch controllers

- A node makes a local choice depending on global topology:
  - this local/global link is the fundamental problem
Key problem

- How to make correct local decisions?
  - Each router must know something about global state
- Global state
  - Inherently large
  - Dynamic
  - Hard to collect
- A routing protocol must intelligently summarize relevant information

Requirements

- Minimise routing table space
  - Want entries to be quick to look up
  - Want less to exchange with peers
- Minimise number and frequency of control messages
- Robustness: want to avoid
  - Black holes – traffic vanishes never to be seen again
  - Loops – traffic gets stuck within the network
  - Oscillations – paths cycle between alternatives
- Use optimal path

Choices

- Centralised vs. distributed routing
  - Centralised is simpler, but prone to failure and congestion
- Source-based vs. hop-by-hop
  - How much is placed in the packet header?
  - Intermediate: loose-source routing
- Stochastic vs. deterministic
  - Stochastic spreads load, avoiding oscillations, but misorders
- Single vs. multiple path
  - Primary and alternative paths (compare with stochastic)
- State-dependent vs. state-independent
  - Do routes depend on current network state (e.g. delay)

Outline

- Routing in telephone networks
- Distance-vector routing
- Link-state routing
- Choosing link costs
- Hierarchical routing
- Internet routing protocols
- Routing within a broadcast LAN
- Multicast routing
- Routing with policy constraints
- Routing for mobile hosts
USA telephone network topology

- 3-level hierarchy, with a fully-connected core
- AT&T: 135 core switches with nearly 5 million circuits
- Local Exchange Carriers (LECs) may connect to multiple cores

Routing algorithm

- If endpoints are within the same central office (CO), directly connect.
- If call is between COs in the same LEC, use one-hop path between COs.
- Otherwise send call to one of the cores.
- Only major decision is at the toll switch:
  - one-hop or two-hop path to the destination toll switch.
  - (why don’t we need longer paths?)
- Essence of telephone routing problem:
  - which two-hop path to use if one-hop path is full.

Features of telephone network routing

- Stable load:
  - can predict pair-wise load throughout the day
  - can choose optimal routes in advance
- Extremely reliable switches:
  - downtime is less than a few minutes per year
  - can assume that a chosen route is available
  - can’t do this in the Internet
- Single organization controls entire core:
  - can collect global statistics and implement global changes
- Very highly connected network:
- Connections require resources (but all need the same).

The cost of simplicity

- Simplicity of routing a historical necessity.
- But requires:
  - reliability in every component
  - logically fully-connected core
- Can we build an alternative that has the same features as the telephone network, but is cheaper because it uses more sophisticated routing?
  - Yes: that is one of the motivations for ATM.
  - But 80% of the cost is in the local loop:
    - not affected by changes in core routing
  - Moreover, many of the software systems assume topology:
    - too expensive to change them.
**Dynamic Non-Hierarchical Routing (DNHR)**
- Simplest core routing protocol
  - accept call if one-hop path is available, else drop
- **DNHR**
  - divides day into around 10-periods
  - in each period, each toll switch is assigned a primary one-hop path and a list of alternatives
  - can overflow to alternative if needed
  - drop only if all alternate paths are busy
    - Crankback to previous node in hierarchy
- Problems
  - does not work well if actual traffic differs from prediction

**Metastability**
- Burst of activity can cause network to enter metastable state
  - high blocking probability even with a low load
- Removed by trunk reservation
  - prevents spilled traffic from taking over direct path

**Trunk status map routing (TSMR)**
- Dynamic non-hierarchical routing measures traffic once a week
- TSMR updates measurements once an hour or so
  - only if it changes “significantly”
- List of alternative paths is more up to date

**Real-time network routing (RTNR)**
- No centralised control
- Each toll switch maintains a list of lightly loaded links
- Intersection of source and destination lists gives set of lightly loaded paths
- Example
  - At A, list is C, D, E → links AC, AD, AE lightly loaded
  - At B, list is D, F, G → links BD, BF, BG lightly loaded
  - A asks B for its list
  - Intersection = D → AD and BD lightly loaded, ADB lightly loaded → it is a good alternative path
  - Very effective in practice: only about a couple of calls blocked in core out of about 250 million calls attempted every day
Outline

- Routing in telephone networks
- Distance-vector routing
- Link-state routing
- Choosing link costs
- Hierarchical routing
- Internet routing protocols
- Routing within a broadcast LAN
- Multicast routing
- Routing with policy constraints
- Routing for mobile hosts

Two key routing algorithms

- Environment
  - Links and routers unreliable
  - Alternative paths scarce
  - Traffic patterns can change rapidly
- Two key algorithms
  - **Distance vector**: tell your neighbours about reaching everyone
  - **Link-state**: tell everyone about reaching your neighbours
- Both assume router knows
  - Address of each neighbour
  - Cost of reaching each neighbour
- Both allow a router to determine global routing information by talking to its neighbours

Distance vector routing: Basic idea

- Node tells its neighbours its best idea of distance to every other node in the network
- Node receives these *distance vectors* from its neighbours
- Updates its notion of best path to each destination, and the next hop for this destination
- Features
  - Distributed
  - Adapts to traffic changes and link failures
  - Suitable for networks with multiple administrative entities

Example

*Computation at A when DV from B arrives*  
\[ \text{Cost to go to B} + \text{Cost to dest. from B} = \text{Cost to dest. via B} \]

*Current cost from A*  
\[ \text{Min} \]

*New cost = new DV for A*  
\[ \text{Next hop} \{ \text{B, C} \} \]
Why it works

- Each node knows its true cost to its neighbours
- This information is spread to its neighbours the first time it sends out its distance vector
- Each subsequent dissemination spreads the truth one hop
- Eventually, it is incorporated into routing table everywhere in the network
- Proof: Bellman and Ford, 1957

Iterate to reach global fixed point.
But what if there isn’t a global fixed point?
- The Internet is a rather dynamic environment after all…

Problems with distance vector

- Count to infinity

Dealing with the problem

- Path vector
  - DV carries path to reach each destination
- Split horizon
  - Never tell neighbour cost to X if neighbour is next hop to X
  - Doesn’t work for 3-way count to infinity
- Triggered updates
  - Exchange routes on change, instead of on timer
  - Faster count up to infinity
- More complicated
  - Source tracing
    - DUAL

Outline

- Routing in telephone networks
- Distance-vector routing
- Link-state routing
- Choosing link costs
- Hierarchical routing
- Internet routing protocols
- Routing within a broadcast LAN
- Multicast routing
- Routing with policy constraints
- Routing for mobile hosts
Link state routing

- In distance vector, router knows only cost to each destination
  - Hides information, causing problems
- In link state, router knows entire network topology, and computes shortest path by itself
  - Independent computation of routes
  - Potentially less robust
- Key elements
  - Topology dissemination
  - Computing shortest routes

Sequence numbers

- How do we know an LSP is new?
- Use a sequence number in LSP header
- Greater sequence number is newer
- What if sequence number wraps around?
  - Smaller sequence number is now newer!
  - (Hint: use a large sequence space)
- On boot up, what should be the initial sequence number?
  - Have to somehow purge old LSPs
  - Two solutions
    - Aging
    - Lollipop sequence space

Link state: topology dissemination

- A router describes its neighbours with a link state packet (LSP)

```
A --1-- B
  |     |
  | 1   |
  V     V
C --2-- D
```

- LSPs created by A
- Use controlled flooding to distribute this everywhere
  - Store an LSP in an LSP database
  - If new, forward to every interface other than incoming one
  - A network with E edges will copy at most 2E times

Aging

- Creator of LSP puts timeout value in the header
- Router removes LSP when it times out
  - Also floods this information to the rest of the network (why?)
- So, on booting, router just has to wait for its old LSPs to be purged
- But what age to choose?
  - If too small
    - Purged before fully flooded (why?)
    - Needs frequent updates
  - If too large
    - Router waits idle for a long time on rebooting
A better solution

- Need a unique start sequence number
- a is older than b if:
  - a < 0 and a < b
  - a > 0, a < b, and b-a < N/4
  - a > 0, b > 0, a > b, and a-b > N/4

More on lollipops

- If a router gets an older Link State Packet, it tells the sender about the newer LSP
- So, newly booted router quickly finds out its most recent sequence number
- It jumps to one more than that
- -N/2 is a trigger to evoke a response from community memory

Recovering from a partition

- On partition, LSP databases can get out of synchronisation
- Databases described by database descriptor records
- Routers on each side of a newly restored link talk to each other to update databases (determine missing and out-of-date LSPs)

Router failure

- How to detect?
  - Use some sort of heartbeat protocol
  - Send periodic “Hello” packets
    - Much smaller than routing information
- Hello packet may be corrupted
  - Traffic might appear to come from the dead router
  - … so age anyway
  - On a timeout, flood information about the failed router
- (Not to be confused with the HELLO protocol (RFC 891)
  - Time-based Distance Vector routing algorithm)
Securing LSP databases

- LSP databases *must* be consistent to avoid routing loops
- Malicious agent may inject spurious LSPs
- Routers must actively protect their databases
  - Checksum LSPs
  - ACK LSP exchanges
  - Passwords

Computing shortest paths

- Nothing you don’t know how to do already
- Dijkstra’s algorithm
  - Maintain a set of nodes P to whom we know shortest path
  - Consider every node one hop away from nodes in P = T
  - Find every way in which to reach a given node in T, and choose shortest one
  - Then add this node to P

Example

Link state versus distance vector

- Criteria
  - Stability
  - Multiple routing metrics
  - Convergence time after a change
  - Communication overhead
  - Memory overhead
- Both are evenly matched
- Both widely used
Outline

- Routing in telephone networks
- Distance-vector routing
- Link-state routing
- Choosing link costs
- Hierarchical routing
- Internet routing protocols
- Routing within a broadcast LAN
- Multicast routing
- Routing with policy constraints
- Routing for mobile hosts

Choosing link costs

- Shortest path uses link costs
- Can use either static or dynamic costs
- In both cases: cost determines amount of traffic on the link
  - Lower the cost, more the expected traffic
  - If dynamic cost depends on load, can have oscillations (why?)

Static metrics

- Simplest: set all link costs to 1 → min hop routing
  - ... but 56k modem link is not the same as an E3 link!
- Give links weight proportional to capacity
  - (low weight value links will be preferred)
- What happens to B-A-C traffic when A-B link is congested?

```
  A  E3  B
  E3  E1  E3
  C  E1  D
```

Weights

- E3 = 1
- E1 = 10

Dynamic metrics

- A first cut (ARPAnet original)
- Cost proportional to length of router queue
  - Independent of link capacity
- Many problems when network is loaded
  - Queue length averaged over a small time → transient spikes caused major rerouting
  - Wide dynamic range → network completely ignored paths with high costs
  - Queue length assumed to predict future loads → opposite is true (why?)
  - No restriction on successively reported costs → oscillations
  - All tables computed simultaneously → low cost link flooded
Modified metrics

<table>
<thead>
<tr>
<th>Problem</th>
<th>Potential solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue length averaged over a small time</td>
<td>Queue length averaged over a longer time</td>
</tr>
<tr>
<td>Wide dynamic range queue</td>
<td>Dynamic range restricted</td>
</tr>
<tr>
<td>Queue length assumed to predict future loads</td>
<td>Cost also depends on intrinsic link capacity</td>
</tr>
<tr>
<td>No restriction on successively reported costs</td>
<td>Restriction on successively reported costs</td>
</tr>
<tr>
<td>All tables computed simultaneously</td>
<td>Attempt to stagger table computation</td>
</tr>
</tbody>
</table>

Routing dynamics

Outline

- Routing in telephone networks
- Distance-vector routing
- Link-state routing
- Choosing link costs
- Hierarchical routing
- Internet routing protocols
- Routing within a broadcast LAN
- Multicast routing
- Routing with policy constraints
- Routing for mobile hosts

Hierarchical routing

- Large networks need large routing tables
  - More computation to find shortest paths
  - More bandwidth wasted on exchanging DVs and LSPs
- Solution:
  - Hierarchical routing
- Key idea
  - Divide network into a set of domains
  - Gateways connect domains
  - Computers within domain unaware of outside computers
  - Gateways know only about other gateways
Example

- **Features**
  - Only a few routers in each level
  - Not a strict hierarchy
  - Gateways participate in multiple routing protocols
  - Non-aggregable routers increase core table space

Hierarchy in the Internet

- Three-level hierarchy in addresses
  - Network number
  - Subnet number
  - Host number
- Core advertises routes only to networks, not to subnets
  - e.g. 135.104.*, 192.20.225.*
- Even so, in 1996, about 80,000 networks in core routers
  - Post .com boom many core router manufacturers went bust
  - Likely to be many more networks
  - Much more organisational autonomy also
- Gateways talk to backbone to find best next-hop to every other network in the Internet

External and summary records

- If a domain has multiple gateways
  - *External* records tell hosts in a domain which one to pick to reach a host in an external domain
    - e.g. allows 6.4.0.0 to discover shortest path to 5.* is through 6.0.0.0
  - *Summary* records tell backbone which gateway to use to reach an internal node
    - e.g. allows 5.0.0.0 to discover shortest path to 6.4.0.0 is through 6.0.0.0
- External and summary records contain distance from gateway to external or internal node
  - Unifies distance vector and link state algorithms

Interior and exterior protocols

- Internet has three levels of routing
  - Highest is at *backbone* level, connecting *autonomous systems* (AS)
  - Next level is within AS
  - Lowest is within a LAN
- Protocol between AS gateways: *exterior gateway protocol*
- Protocol within AS: *interior gateway protocol*
Exterior gateway protocol

- Between untrusted routers
  - Mutually suspicious
- Must tell a border gateway who can be trusted and what paths are allowed
- Transit over backdoors is a problem

Interior protocols

- Much easier to implement
- Typically partition an AS into areas
- Exterior and summary records used between areas

Issues in interconnection

- May use different schemes (DV vs. LS)
- Cost metrics may differ
- Need to:
  - Convert from one scheme to another (how?)
  - Use the lowest common denominator for costs
  - Manually intervene if necessary

Outline

- Routing in telephone networks
- Distance-vector routing
- Link-state routing
- Choosing link costs
- Hierarchical routing
- Internet routing protocols
- Routing within a broadcast LAN
- Multicast routing
- Routing with policy constraints
- Routing for mobile hosts
Common routing protocols

- Interior
  - RIP
  - OSPF
- Exterior
  - EGP
  - BGP
- ATM
  - PNNI

Routing Information Protocol (RIP)

- Distance vector
- Cost metric is hop count
- Infinity = 16
- Exchange distance vectors every 30 seconds
- Split horizon
- Useful for small subnets
  - Easy to install

Open Shortest Path First (OSPF)

- Link-state
- Uses areas to route packets hierarchically within AS
- Complex
  - LSP databases to be protected
- Uses designated routers to reduce number of endpoints

Exterior Gateway Protocol (EGP)

- Original exterior gateway protocol
- Notionally distance-vector
- Costs are either 128 (reachable) or 255 (unreachable)
  - DV simplifies reachability protocol
  - Backbone topology must be loop free already
- Allows administrators to pick neighbours to peer with
- Allows backdoors (by setting backdoor cost < 128)
Border Gateway Protocol (BGP)

- Path-vector
  - Distance vector annotated with entire path
  - Also with policy attributes
  - Guaranteed loop-free
- Can use non-tree backbone topologies
- Uses TCP to disseminate DVs
  - Reliable
  - ... but subject to TCP flow control
- Policies are complex to set up

Private Network-Network Interface (PNNI)

- Link-state
- Many levels of hierarchy
- Switch controllers at each level form a peer group
- Group has a group leader
- Leaders are members of the next higher level group
- Leaders summarise information about group to tell higher level peers
- All records received by leader are flooded to lower level
- LSPs can be annotated with per-link QoS metrics
- Switch controller uses this to compute source routes for call-setup packets

Outline

- Routing in telephone networks
- Distance-vector routing
- Link-state routing
- Choosing link costs
- Hierarchical routing
- Internet routing protocols
- Routing within a broadcast LAN
- Multicast routing
- Routing with policy constraints
- Routing for mobile hosts

Routing within a broadcast LAN

- What happens at an endpoint?
- On a point-to-point link, no problem
- On a broadcast LAN
  - Is packet meant for destination within the LAN?
  - If so, what is the datalink address?
  - If not, which router on the LAN to pick?
  - What is the router’s datalink address?
### Internet solution

- All hosts on the LAN have the same subnet address
- So, easy to determine if destination is on the same LAN
- Destination’s datalink address determined using ARP
  - Broadcast a request
  - Owner of IP address replies
- To discover routers (if not pre-specified statically or via DHCP)
  - Routers periodically sends router advertisements
    - With preference level and time to live
  - Pick most preferred router
  - Delete overage records
  - Can also force routers to reply with *solicitation message*

### Redirection (ICMP)

- How to pick the best router?
- Send message to arbitrary router
- If that router’s next hop is another router on the same LAN, host gets a *redirect* message
- It uses this for subsequent messages

### Outline

- Routing in telephone networks
- Distance-vector routing
- Link-state routing
- Choosing link costs
- Hierarchical routing
- Internet routing protocols
- Routing within a broadcast LAN
- Multicast routing
- Routing with policy constraints
- Routing for mobile hosts

### Multicast routing

- Unicast: single source sends to a single destination
- Multicast: hosts are part of a *multicast group*
  - Packet sent by *any* member of a group are received by *all*
- Useful for
  - Multiparty videoconference
  - Distance learning
  - Resource location
  - Imaging machines on a LAN
  - Internet TV
Multicast group

- Associates a set of senders and receivers with each other
  - But independent of them
  - Created either when a sender starts sending from a group
  - Or a receiver expresses interest in receiving
  - Even if no one else is there!
- Sender does not need to know receivers' identities
  - *Rendezvous point*

Addressing

- Multicast group in the Internet has its own Class D address
  - Looks like a host address, but isn't
- Senders send to the address
- Receivers anywhere in the world request packets from that address
- “Magic” is in associating the two: *dynamic directory service*
- Four problems
  - Which groups are currently active
  - How to express interest in joining a group
  - Discovering the set of receivers in a group
  - Delivering data to members of a group

Expanding ring search

- A way to use multicast groups for resource discovery
- Routers decrement TTL when forwarding
- Sender sets TTL and multicasts
  - Reaches all receivers ← TTL hops away
- Discovers local resources first
- Since heavily loaded servers can keep quiet, automatically distributes load

Multicast flavours

- Unicast: point to point
- Multicast:
  - Point to multipoint
  - Multipoint to multipoint
- Can simulate point to multipoint by a set of point to point unicasts
- Can simulate multipoint to multipoint by a set of point to multipoint multicasts
- The difference is efficiency
Example

- Suppose A wants to talk to B, G, H, I; and B to A, G, H, I
- With unicast, 4 messages sent from each source
  - Links AC, BC carry a packet in triplicate
- With point to multipoint multicast, 1 message sent from each source
  - But requires establishment of two separate multicast groups
- With multipoint to multipoint multicast, 1 message sent from each source,
  - Single multicast group

Issues in wide-area multicast

- Difficult because
  - Sources may join and leave dynamically
    - Need to dynamically update shortest-path tree
  - Leaves of tree are often members of broadcast LAN
    - Would like to exploit LAN broadcast capability
  - Would like a receiver to join or leave without explicitly notifying sender
    - Otherwise it will not scale

Shortest path tree

- Ideally, want to send exactly one multicast packet per link
  - Forms a multicast tree rooted at sender
- Optimal multicast tree provides shortest path from sender to every receiver
  - Shortest-path tree rooted at sender

Multicast in a broadcast LAN

- Wide area multicast can exploit a LAN’s broadcast capability
- E.g. Ethernet will multicast all packets with multicast bit set on destination address

- Two problems:
  - What multicast MAC address corresponds to a given Class D IP address?
  - Does the LAN have contain any members for a given group (why do we need to know this?)
Class D to MAC translation

- Multiple Class D addresses map to the same MAC address
- Well-known translation algorithm => no need for a translation table

Internet Group Management Protocol

- Detects if a LAN has any members for a particular group
  - If no members, then we can prune the shortest path tree for that group by telling parent
- Router periodically broadcasts a query message
- Hosts reply with the list of groups they are interested in
- To suppress traffic
  - Reply after random timeout
  - Broadcast reply
  - If someone else has expressed interest in a group, drop out
- To receive multicast packets:
  - Translate from class D to MAC and configure adapter

Wide area multicast

- Assume
  - Each endpoint is a router
  - A router can use IGMP to discover all the members in its LAN that want to subscribe to each multicast group
- Goal
  - Distribute packets coming from any sender directed to a given group to all routers on the path to a group member

Simplest solution

- Flood packets from a source to entire network
- If a router has not seen a packet before, forward it to all interfaces except the incoming one
- Pros
  - Simple
  - Always works!
- Cons
  - Routers receive duplicate packets
  - Detecting that a packet is a duplicate requires storage, which can be expensive for long multicast sessions
A clever solution

- Reverse path forwarding
- Rule
  - Forward packet from S to all interfaces if and only if packet arrives on the interface that corresponds to the shortest path to S
  - No need to remember past packets
  - C need not forward packet received from D

Cleverer

- Don’t send a packet downstream if you are not on the shortest path from the downstream router to the source
- C need not forward packet from A to E

- Potential confusion if downstream router has a choice of shortest paths to source (see figure on previous slide)

Pruning

- RPF does not completely eliminate unnecessary transmissions

- B and C get packets even though they do not need it
- Pruning → router tells parent in tree to stop forwarding
- Can be associated either with a multicast group or with a source and group
  - Trades selectivity for router memory

Rejoining

- What if host on C’s LAN wants to receive messages from A after a previous prune by C?
  - IGMP lets C know of host’s interest
  - C can send a join(group, A) message to B, which propagates it to A
  - Or, periodically flood a message; C refrains from pruning
A problem

- Reverse path forwarding requires a router to know shortest path to a source
  - Known from routing table
- Doesn’t work if some routers do not support multicast
  - Virtual links between multicast-capable routers
  - Shortest path to A from E is not C, but F

![Diagram](image1.png)

A problem (contd.)

- Two problems
  - How to build virtual links
  - How to construct routing table for a network with virtual links

Tunnels

- Why do we need them?

  - Consider packet sent from A to F via multicast-incapable D
  - If packet’s destination is Class D, D drops it
  - If destination is F’s address, F doesn’t know multicast address!
  - So, put packet destination as F, but carry multicast address internally
  - Encapsulate IP in IP -> set protocol type to IP-in-IP

![Diagram](image2.png)

Multicast routing protocol

- Interface on “shortest path” to source depends on whether path is real or virtual

  - Shortest path from E to A is not through C, but F
    - so packets from F will be flooded, but not from C
  - Need to discover shortest paths only taking multicast-capable routers into account
    - DVMRP

![Diagram](image3.png)
DVMRP

- Distance-vector Multicast routing protocol
- Very similar to RIP
  - Distance vector
  - Hop count metric
- Used in conjunction with
  - Flood-and-prune (to determine memberships)
    - Prunes store per-source and per-group information
  - Reverse-path forwarding (to decide where to forward a packet)
  - Explicit join messages to reduce join latency (but no source info, so still need flooding)

MOSPF

- Multicast extension to OSPF
- Routers flood group membership information with LSPs
- Each router independently computes shortest-path tree that only includes multicast-capable routers
  - No need to flood and prune
- Complex
  - Interactions with external and summary records
  - Need storage per group per link
  - Need to compute shortest path tree per source and group

Core-based trees

- Problems with DVMRP-oriented approach
  - Need to periodically flood and prune to determine group members
  - Need to source per-source and per-group prune records at each router
- Key idea with core-based tree
  - Coordinate multicast with a core router
  - Host sends a join request to core router
  - Routers along path mark incoming interface for forwarding

Example

- Pros
  - Routers not part of a group are not involved in pruning
  - Explicit join/leave makes membership changes faster
  - Router needs to store only one record per group
- Cons
  - All multicast traffic traverses core, which is a bottleneck
  - Traffic travels on non-optimal paths

A sends join to B
B adds D to G
B marks AB to forward data from G
D joins G
Core forwards to A
Core marks Core-C to forward data from G
S sends to core with destination G
D sends G to B
B adds D to G
Core forwards to A
Core marks Core-C to forward data from G
S sends to core with destination G
Protocol independent multicast (PIM)

- Tries to bring together best aspects of CBT and DVMRP
- Choose different strategies depending on whether multicast tree is dense or sparse
  - Flood and prune good for dense groups
  - Only need a few prunes
  - CBT needs explicit join per source/group
  - CBT good for sparse groups
- Dense mode PIM == DVMRP
- Sparse mode PIM is similar to CBT
  - ... but receivers can switch from CBT to a shortest-path tree

PIM (contd.)

- In CBT, E must send to core
- In PIM, B discovers shorter path to E (by looking at unicast routing table)
  - Sends join message directly to E
  - Sends prune message towards core
- Core no longer bottleneck
- Survives failure of core

More on core

- Renamed a rendezvous point
  - because it no longer carries all the traffic like a CBT core
- Rendezvous points periodically send “I am alive” messages downstream
- Leaf routers set timer on receipt
- If timer goes off, send a join request to alternative rendezvous point
- Problems
  - How to decide whether to use dense or sparse mode?
  - How to determine “best” rendezvous point?

Outline

- Routing in telephone networks
- Distance-vector routing
- Link-state routing
- Choosing link costs
- Hierarchical routing
- Internet routing protocols
- Routing within a broadcast LAN
- Multicast routing
  - Routing with policy constraints
- Routing for mobile hosts
Routing vs. policy routing

- In standard routing, a packet is forwarded on the ‘best’ path to destination
  - Choice depends on load and link status
- With policy routing, routes are chosen depending on policy directives regarding things like
  - Source and destination address
  - Transit domains
  - Quality of service
  - Time of day
  - Charging and accounting
- The general problem is still open
  - Fine balance between correctness and information hiding

Multiple metrics

- Simplest approach to policy routing
- Advertise multiple costs per link
- Routers construct multiple shortest path trees

![Diagram of multiple metrics](image)

- Cost per packet = 10p
- Delay = 100ms
- Cost per packet = 5p
- Delay = 200ms

Problems with multiple metrics

- All routers must use the same rule in computing paths
  - Otherwise we risk forming routing loops
- Remote routers may misinterpret policy
  - Source routing may solve this
  - But introduces other problems
    - Increased routing data sent (OK if connection oriented)
    - Source needs up-to-date routing data to form paths…

Provider selection

- Another simple approach
- Assume that a single service provider provides almost all the path from source to destination
  - e.g. AT&T or MCI
- Then, choose policy simply by choosing provider
  - This could be dynamic (agents!)
- In Internet, can use a loose source route through service provider’s access point
- Or, multiple addresses/names per host
Crankback

- Consider computing routes with QoS guarantees
- Router returns packet if no next hop with sufficient QoS can be found
- In ATM networks (PNNI) used for the call-setup packet
- In Internet, may need to be done for _every_ packet!
  - Will it work?

Mobile routing

- How to find a mobile host?
- Two sub-problems
  - Location (where is the host?)
  - Routing (how to get packets to it?)
- We will study mobile routing in the Internet and in the telephone network

Outline

- Routing in telephone networks
- Distance-vector routing
- Link-state routing
- Choosing link costs
- Hierarchical routing
- Internet routing protocols
- Routing within a broadcast LAN
- Multicast routing
- Routing with policy constraints
- Routing for mobile hosts

Mobile routing in the telephone network

- Each cell phone has a global ID that it tells remote MTSO when turned on (using slotted ALOHA up-channel)
- Remote MTSO tells home MTSO
- To phone: call forwarded to remote MTSO to closest base
- From phone: call forwarded to home MTSO from closest base
- New MTSOs can be added as load increases
Mobile routing in the Internet

- Very similar to mobile telephony
  - but outgoing traffic does not go through home
  - and need to use tunnels to forward data
- Use registration packets instead of slotted ALOHA
  - Passed on to home address agent
- Old care-of-agent forwards packets to new care-of-agent until home address agent learns of change

Problems with mobile routing

- Security
  - Mobile and home address agent share a common secret
  - Checked before forwarding packets to COA
- Loops

1. MH moves to BA redirects to B
2. MH moves back to A
3. Old, duplicate redirect reaches A, forming loop