lecture 9

LAN/AS

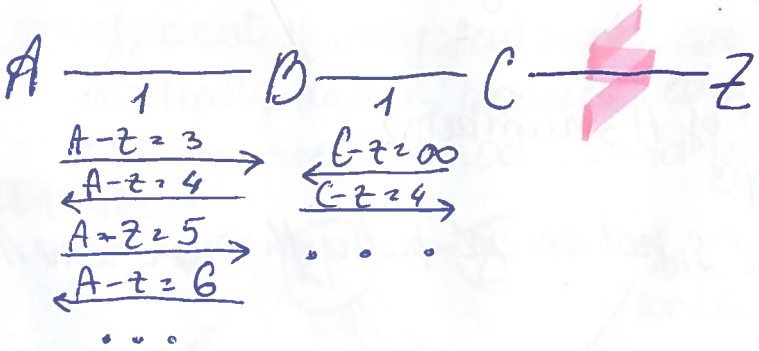
Distance Vector algorithm

- Upon receiving update from neighbour
  - For each destination:
    - add link cost to cost reported by neighbour
    - if it is better than cost we know → update table

$$D_i^{h+1} = \min_j \left[d_{ij} + D_j^h \right]$$

h - iteration counter
d_{ij} - cost of link between i & j
D_{ij} - cost of best path to node 1 known by i at h

Counting to infinity problem



- A does not know about Z's failure
- A says to B that it can still reach Z
- B says to A that it can still reach Z

Partial Solutions

- Choose a low value of infinity
- Split horizon (A will not tell B about t, since it needs B to go to Z)
- Split horizon with poison reverse (A tells B that A→Z is ∞, since via B)
- Send entire path in routing updates (less efficient, used by BGP)
- Postpone "good news" until "bad news" have chance to spread

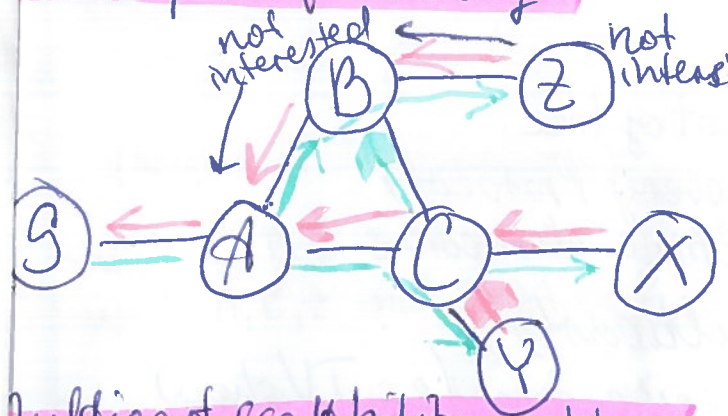
Routing in the Internet

- Internet is divided into Autonomous Systems (AS)
  - Stub AS: only one connection to the rest (small ISP)
  - Multihomed AS: more than 1 connection to the rest; but doesn't carry traffic for others
  - Transit AS: also carries traffic for other ASs (SURFnet)
- unique 16-bit number

every AS can use its own preferred protocol for routing
routing among ASs is done using Border Gateway Protocol

- Routing within AS
  - static routing
  - bridges/switches with rapid spanning tree protocol
  - RIP (Routing Information Protocol)
    - based on DVR
    - exchange routing tables every 30s
    - max. hops = 15
  - OSPF (Open Shortest Path First)
    - based on link state
    - can subdivide the AS's network further into "routing areas"
- Routing between ASs: BGP
  - based on DVR
  - advertises complete path (list of AS numbers)
  - path info helps to detect loops
  - admins can install policies (e.g. not accept paths through some AS)
  - 3 types of neighbours:
    - providers – get money for traffic
    - peers – no payment
    - customers – give money for traffic
- Network-supported multicast
  - Src sends pkt only once
  - Routers send out multiple copies of pkt when needed
  - Problem: How do routers know who should get a copy?
  - Solution 1: List all dests in pkt header → inefficient; src may not even know dest.
  - Solution 2: address indirection
    - Pkt contains addr that belong to group of dests
    - Router knows which hosts belong to which groups and copy accordingly
  - How do routers know where to send multicast?
    → IGMP (Internet Group Management Protocol)
    Hosts tell their local router in which groups they are interested
  - Rest of network:
    - several network-layer multicast routing algorithms (DVMRP, PIM-SM, PIM-SSM etc.)

Reverse path forwarding



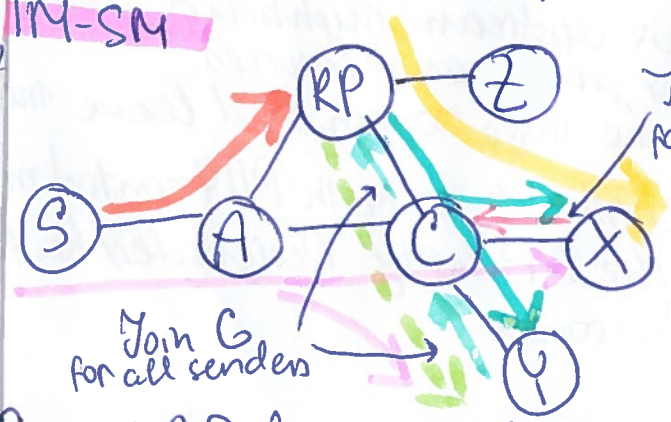
- Path from every node to S together form a spanning tree rooted at S
- For multicast, the tree is "pruned" by non-interested nodes (routers) telling their upstream neighbours

Multicast scalability problems

Link-state based algorithms
- amount of information and computation too large

Reverse path forwarding
- inefficient if only few nodes are interested
  - initially data is broadcasted over entire network
  - non-interested nodes send prune msgs

PIM-SM



C and RP learn that they should send packets for Group G toward Y (regardless of sender).
C learns that it should also send packets for Group G towards X (regardless of sender)
The shared tree (rooted at RP) is now complete.
S can send data to the multicast group by encapsulating it in unicast packet to RP
To remove the detour via RP, X and Y may send a src-specific join request to S, to create a src-specific tree

Note the tradeoff between optimality and scalability

- Protocol Independent Multicast (PIM) variants
  - PIM-SM (Sparse Mode)
    - use Rendez-Vous point as root of tree.
  - MSDP (Multicast Source Discovery Protocol)
    - extension for sources in multiple domains
  - PIM-SSM (Source-specific Multicast)
    - no RP, create only source specific tree (e.g. TV show)
  - BIDIR-PIM (Bi-directional tree)
    - create multicast tree where source is just another member (e.g. conferencing)
- More on variants:
  - PIM-SM:
    - ~~not~~ assumed that most subnets in the network will not want any given multicast pkt.
    - routers must explicitly tell their upstream neighbours about their interest in particular groups and sources.
    - routers use PIM Join and Prune msgs to join and leave
    - to send to RP sources must encapsulate data in PIM control and send it by unicast to the RP using Designated R
      DR - source's local network's router

- Multicast in the current internet
  - most internet connections do not yet have multicast routing
    → use of tunnels to cross non-multicast-capable networks
  - multicast within specific networks.
  - alternative: application-layer multicast. Apps send copies of pkts
- Mobility
  - non-static
  - A host (e.g. smartphone) moves from one network to another
    - gets a new IP addr
    - existing TCP connections are lost
    - solution: Mobile-IP (not used much)
  - A bunch of nodes move into and out of each other's radio range but still want to remain connected
    - ad-hoc networking
    - needs special routing algorithm (AODV)
- non-solutions:
  - new IP addr for moved host → works for new connections, but existing are lost
  - specific route for this host in networks forwarding tables → not scalable
- Routing in ad-hoc networks
  - Ad-hoc networks: nodes are mobile, connectivity all the time
  - Link-state and DVR are unsuitable: too much data exchanged
  - solution: only exchange routing info when needed, "on demand".

~~Addressing~~

- Mobile IP
  - mobile node's home addr always identifies the mobile node
  - uses registration mechanism to register the care-of address with home agent
  - home agent redirects datagrams from home network to the care-of addr by new IP header (dest is node's care-of addr)
    ↓
    tunneling

- Ad-hoc On-Demand Distance Vector
  - host-specific
  - smaller networks
- Source broadcasts Route Request (RREQ)
- Node replies to RREQ if:
  - it is the destination
  - has fresh enough route to the destination
- Otherwise, it rebroadcasts the RREQ and updates its Route Table with src. node information.
- Loops in flooding are prevented using a unique identifier in each RREQ.
- Destination or intermediate node replies with Route Reply (RREP)
- Nodes along the path record route to destination in their Route Table.
- When the nodes many changes connectivity?
  - Source moves: can restart route discovery
  - Destination/node on path moves:
    upstream neighbours sends RREP with cost = infinite, to remove outdated routes
- RREPs and RREQs have sequence numbers to ensure latest is used.