CMSC 417 Spring 2016 Lecture #6 2/17/2016

Agenda

⇒ project 1
  - initial public tests are up
  - updated pingmap-router.py
  - updated assumptions

⇒ Link State Routing

⇒ Maybe IP

CMSC 417 Spring 2016 Lecture #6 2/17/2016

Link State Routing)

⇒ Key Idea: If everyone had a full view of the topology, they could just directly compute routes and install forwarding entries

⇒ Challenges

- □ how to get everyone the full topology (Reliable Flooding)
- □ how to make sure everyone's view is consistent
- □ how to make sure everyone computes routes in compatible ways

CMSC417 Spring 2016 Lecture #6 2/17/2016

Link State Routing)

0.) Reliably flood list of neighbors and costs
1.) Run Dijkstra's shortest path to find routes
2.) Install forwarding rules based on the routes

Reliable Flooding)

Link State Packet (LSP) Actually an LSA in OSPF for Link State Advertisement

⇒ node ID
⇒ list of (neighbor node ID, cost) tuples
⇒ sequence #
⇒ age (a TTL that counts up to exipiration)

at each router

0.) reliably send LSP to all neighbors
1.) on recving an LSP, forward to all neighbors but the one who sent it to you if it's newer (according to its seq #)
2.) resend LSP on a timer or tupolugy update (in practice, updates on the order of hours)

CMSC 417 Spring 2016 Lecture #6 2/17/2016

Link State Routing : Path Computation

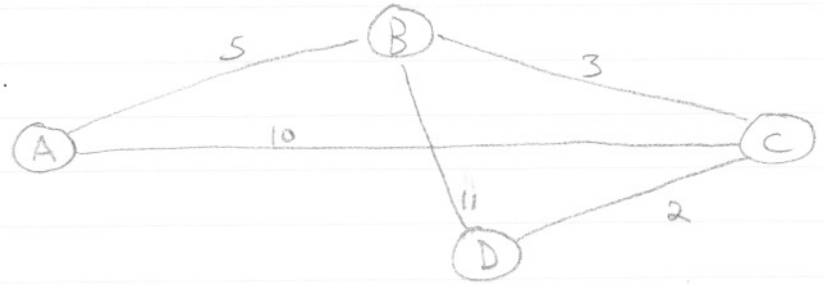
just use Dijkdra (at node S)

```
$M \leftarrow \{S\}$

$\forall n \in N - M$                  // N is all known nodes
    $C(n) = l(S, n)$               // C(n) is cost to get to n
                                   // l(S, n) is the direct link cost
                                   //        from S to n

while ( M != N )
    $w \leftarrow \underset{x \in N-M}{\text{argmin}}\ C(x)$

    $M = M \cup \{w\}$
    $\forall n \in N - M$
        $C(n) \leftarrow \min(C(n), C(w) + l(w, n))$
```



at D

0) (D, 0, -) → expand D's LSP and see lowest cost is C
1) (C, 2, C) → " C's " B
2) (B, 5, C) → " B's " A
3) (A, 10, C) → done

CMSC417 Spring 2016 Lecture #6 2/17/2016

Link State Quirks)

expiring old LSPs

⇒ if a node goes down, you can assume its neighbor will tell you, so expiring isn't important for topological correctness

⇒ when a node reboots, it might have forgotten it's seq #,
- □ other nodes will ignore its messages until the seq # increases to where it left off ... which could be days
- □ two solutions:
  - if a node recieves its own LSP, read the seq# and pick up from there
  - LSPs age out at other nodes, so they'll accept a new one eventually

⇒ when an LSP expires, send the LSP with age/TTL ~~set~~ to ∅ to tell others to delete it as well

DV vs. LS)

In distance vector you tell only your neighors, but tell them everything you know.

I link state you tell everyone, but tell them only about your neighbors.

CMSC 417 Spring 2016 Lecture #6 2/17/2016

Link State Routing Summary

Pros

- + fast
- + no flapping
- + low overhead
- + no count to ∞

Cons

- - high(er) storage cost
- - high(er) compute cost