CS118 Discussion 1A, Week 8

Zengwen Yuan (zyuan [at] cs.ucla.edu)
Boelter Hall 9436, Friday 12:00—1:50 p.m.
Outline

• Project 2

• Lecture review:
  • Routing
    • Link state routing, Distance vector routing, Hierarchical routing, BGP
  • Introduction to link layer
Course Project 2: questions?

- Demo time survey! https://goo.gl/forms/LbNF2dT1GGvFPiPP2
- We want to implement **byte-stream** reliable data transfer
  - WND is in unit of bytes, not packets
- How to realize timeout?
  - **Option 1**: select() with countdown timer
  - **Option 2**: alarm() + signal()
- Should we implement RTT estimation?
  - Depends on if you plan to realize congestion control
Routing: concepts

- Global or decentralized information?
  - global: all routers have complete topology, link cost info
  - algorithm?
Routing: concepts

• Global or decentralized information?

  • global: all routers have complete topology, link cost info

  • “link state” algorithms
Routing: concepts

• Global or decentralized information?
  • global: all routers have complete topology, link cost info
  • “link state” algorithms
  • decentralized: router knows physically-connected neighbors, link costs to neighbors; iterative process of computation, exchange of info with neighbors
  • algorithm?
Routing: concepts

• Global or decentralized information?
  • global: all routers have complete topology, link cost info
  • “link state” algorithms
  • decentralized: router knows physically-connected neighbors, link costs to neighbors; iterative process of computation, exchange of info with neighbors
  • “distance vector” algorithms
Link state routing

- Dijkstra’s algorithm
  - net topology, link costs known to all nodes
  - computes least cost paths from one node (‘source”) to all other nodes
  - iterative: after k iterations, know least cost path to k destinations
Link state routing: algorithm

1. **Initialization:**
   2. \( N' = \{u\} \)
   3. for all nodes \( v \)
      4. if \( v \) adjacent to \( u \)
         5. then \( D(v) = c(u,v) \)
      6. else \( D(v) = \infty \)
   7. 
   8. **Loop**
      9. find \( w \) not in \( N' \) such that \( D(w) \) is a minimum
     10. add \( w \) to \( N' \)
     11. update \( D(v) \) for all \( v \) adjacent to \( w \) and not in \( N' \):
         12. **[Link cost update heuristic from Dijkstra algo.]**
     13. until all nodes in \( N' \)

\( c(x, y) \): link cost from node \( x \) to \( y \); \( c(x, y) = \infty \) if not direct neighbors
\( D(v) \): current value of cost of path from source to destination \( v \)
\( p(v) \): predecessor node along path from source to \( v \)
\( N' \): set of nodes whose least cost path definitively known
Link state routing: algorithm

1 Initialization:
2 \[ N' = \{u\} \]
3 for all nodes v
4   if v adjacent to u
5     then \[ D(v) = c(u,v) \]
6   else \[ D(v) = \infty \]

8 Loop
9   find w not in N' such that \[ D(w) \] is a minimum
10   add w to N'
11   update \[ D(v) \] for all v adjacent to w and not in N':
12 \[ D(v) = \min( D(v), D(w) + c(w,v) ) \]
13 until all nodes in N'

c(x, y): link cost from node x to y; c(x, y) = \infty if not direct neighbors
D(v): current value of cost of path from source to destination v
p(v): predecessor node along path from source to v
N': set of nodes whose least cost path definitively known
Link state routing: example

- Using link state routing to setup a forwarding table for node u
Let’s work it out

<table>
<thead>
<tr>
<th>$N'$</th>
<th>$D(v), p(v)$</th>
<th>$D(w), p(w)$</th>
<th>$D(x), p(x)$</th>
<th>$D(y), p(y)$</th>
<th>$D(z), p(z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u$</td>
<td>2, $u$</td>
<td>5, $u$</td>
<td>1, $u$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$ux$</td>
<td>2, $u$</td>
<td>4, $x$</td>
<td></td>
<td>2, $x$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$uxy$</td>
<td>2, $u$</td>
<td>3, $y$</td>
<td></td>
<td></td>
<td>4, $y$</td>
</tr>
<tr>
<td>$uxyv$</td>
<td></td>
<td>3, $y$</td>
<td></td>
<td></td>
<td>4, $y$</td>
</tr>
<tr>
<td>$uxyvw$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4, $y$</td>
</tr>
<tr>
<td>$uxyvwz$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Let’s work it out

<table>
<thead>
<tr>
<th>N’</th>
<th>D(v), p(v)</th>
<th>D(w), p(w)</th>
<th>D(x), p(x)</th>
<th>D(y), p(y)</th>
<th>D(z), p(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>2, u</td>
<td>5, u</td>
<td>1, u</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>ux</td>
<td>2, u</td>
<td>4, x</td>
<td></td>
<td>2, x</td>
<td>∞</td>
</tr>
<tr>
<td>uxy</td>
<td></td>
<td></td>
<td>3, y</td>
<td></td>
<td>4, y</td>
</tr>
<tr>
<td>uxyv</td>
<td></td>
<td></td>
<td>3, y</td>
<td></td>
<td>4, y</td>
</tr>
<tr>
<td>uxyvw</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4, y</td>
</tr>
<tr>
<td>uxyvwz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Distance vector routing

- Bellman-Ford equation (dynamic programming)

- let

- $dx(y) := \text{cost of least-cost path from } x \text{ to } y$

- then

- $dx(y) = ?$
Distance vector routing

• Bellman-Ford equation (dynamic programming)

• let

• \( dx(y) := \text{cost of least-cost path from } x \text{ to } y \)

• then

• \( dx(y) = \min_v \{ c(x,v) + dv(y) \} \)
Distance vector routing: example

- What’s the cost of least-cost path for $u \rightarrow z$?
Let’s work it out

• clearly:

  • $dv(z) = \, ?, \, dx(z) = \, ?, \, dw(z) = \, ?$
Let’s work it out

- clearly:
  - \( dv(z) = 5, \ dx(z) = 3, \ dw(z) = 3 \)

- According to B-F equation:
  - \( du(z) = \min \{ \ ? \} \)
Let’s work it out

• clearly:

  • $dv(z) = 5$, $dx(z) = 3$, $dw(z) = 3$

• According to B-F equation:

  • $du(z) = \min \{ c(u,v) + dv(z), c(u,x) + dx(z), c(u,w) + dw(z) \}$
Let’s work it out

- clearly:
  - $dv(z) = 5$, $dx(z) = 3$, $dw(z) = 3$

- According to B-F equation:
  - $du(z) = \min \{c(u,v) + dv(z), c(u,x) + dx(z), c(u,w) + dw(z)\}$
    - $= \min \{2 + 5, 1 + 3, 5 + 3\} = 4$
Distance vector routing: key idea

- from time-to-time, each node sends its own distance vector estimate to neighbors
- when x receives new DV estimate from neighbor, it updates its own DV using B-F equation.
Distance vector routing: caveat

• Count-to-infinity problem.

• Can you work out an example?
Distance vector routing: caveat

- Count-to-infinity problem.
- Can you work out an example?
Distance vector routing: caveat

- Count-to-infinity problem.
- Can you work out an example?

- Can you propose a solution?
  - basic idea?
Distance vector routing: split horizon

• Previous solution idea:

  • split horizon

  • if B reaches D through C, B should not tell C that B can reach D

  • Then C will not attempt to go through B to reach D

  • Are we good?
Distance vector routing: split horizon

• Previous solution idea:
  • split horizon
    • if B reaches D through C, B should not tell C that B can reach D
    • Then C will not attempt to go through B to reach D
  • Are we good?
Distance vector routing: poison reverse

- Split horizon + poison reverse

- if A reaches D through C:
  - A tells C that A’s distance to D is infinite
  - Then C will not attempt to go through A to reach D
  - In practice, infinite == 16 hops
## Link State v.s. Distance Vector

<table>
<thead>
<tr>
<th></th>
<th>Link state</th>
<th>Distance vector</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>message complexity</strong></td>
<td>with n nodes, E links, $O(nE)$ msgs sent</td>
<td>exchange between neighbors only (convergence time varies)</td>
</tr>
<tr>
<td><strong>convergence speed</strong></td>
<td>$O(n^2)$ algorithm requires $O(nE)$ msgs</td>
<td>convergence time varies (may be routing loops)</td>
</tr>
<tr>
<td><strong>robustness</strong></td>
<td>node can advertise incorrect link cost; each node computes only its own table</td>
<td>DV node can advertise incorrect path cost; error propagate thru network</td>
</tr>
<tr>
<td><strong>implementation</strong></td>
<td>OSPF</td>
<td>RIP</td>
</tr>
</tbody>
</table>
Hierarchical routing

- aggregate routers into regions
- AS: autonomous systems
- routers in same AS run same routing protocol
- “intra-AS” routing protocol
- routers in different AS can run different intra-AS routing protocol
BGP (Border Gateway Protocol)

• an inter-domain routing protocol

• BGP provides each AS a means to:
  
  • eBGP: obtain subnet reachability information from neighboring ASs.
  
  • iBGP: propagate reachability information to all AS-internal routers.

• allows subnet to advertise its existence to rest of Internet: “I am here”
BGP: routing policy

- A, B, C are provider networks
- X, W, Y are customer (of provider networks)
- X is attached to two networks.
  - It does not want to route from B via X to C
  - … so X will not advertise to B a route to C
BGP: routing policy

- A advertises path AW to B
- B advertises path BAW to X
- Should B advertise path BAW to C?

Legend:
- Provider network
- Customer network
BGP: routing policy

- A advertises path AW to B
- B advertises path BAW to X
- Should B advertise path BAW to C?
  - No! B gets no “revenue” for routing CBAW since neither W nor C are B’s customers
  - B wants to force C to route to w via A
  - B wants to route only to/from its customers!
BGP: practice problems

• Explain how loops in paths can be detected in BGP.

• BGP advertisements contain complete paths showing the AS’s the path passes through, and so a router can easily identify a loop because an AS will appear two or more times.
BGP: practice problems

• Suppose that there is another stub network V that is a customer of ISP A. Suppose that B and C have a peering relationship, and A is a customer of both B and C. Suppose that A would like to have the traffic destined to W to come from B only, and the traffic destined to V from either B or C. How should A advertise its routes to B and C? What AS routes does C receive?

  • A should advertise to B two routes: A-W and A-V
  • A should advertise to C only one route: A-V
Multicast: RPF

- Reverse Path Forwarding

- For each node, how to forward A’s multicast packets?
Routing: summary

- Intra-domain routing V.S. inter-domain routing
  - Performance V.S. policy
  - Scalability: hierarchical routing
- Distance-vector routing V.S. link-state routing
  - Fully-distributed algorithm V.S. decentralized algorithm
- Unicast V.S. multicast
Link layer: introduction

• understand principles behind link layer services:

• error detection, correction — CRC (cyclic redundancy check)

• sharing a broadcast channel: multiple access

• link layer addressing

• local area networks: Ethernet, VLANs
Medium Access Links and Protocols

- Condition: broadcast channel shared by multiple hosts
  - What if we only have unicast channel?
  - What’s the pros and cons for a broadcast channel?

- Three classes of MAC protocols
  - Channel partitioning: FDMA, TDMA, CDMA
  - Random access: Aloha, CSMA/CD, Ethernet
  - Taking turns: Token ring/passing

- Pros and cons for each class of protocol?
Random access: slotted ALOHA

• Assumptions:
  • all frames same size
  • time divided into equal size slots (time to transmit 1 frame)
  • nodes start to transmit only slot beginning
  • nodes are synchronized
  • if 2 or more nodes transmit in slot, all nodes detect collision
Random access: slotted ALOHA

• suppose: N nodes with many frames to send, each transmits in slot with probability p

• \( \Pr(\text{given node has success in a slot}) = p(1-p)^{(N-1)} \)

• \( \Pr(\text{any node has a success}) = Np(1-p)^{(N-1)} \)

• max efficiency: find \( p^* \) that maximizes \( Np(1-p)^{(N-1)} \)

• for many nodes, take limit of \( Np^*(1-p^*)^{(N-1)} \) as \( N \) goes to infinity, gives:
  • max efficiency = \( 1/e \approx 0.37 \)
Random access: ALOHA efficiency

- Slotted ALOHA max efficiency = $1/e = 0.37$
- Unslotted ALOHA max efficiency = $1/2e = 0.18$
CSMA (carrier sense multiple access)

- Listen before transmit:
  - if channel sensed idle: transmit entire frame
  - if channel sensed busy, defer transmission
  - “don’t interrupt others!”

- Collision?
  - hidden terminal problem
CSMA/CD (collision detection)

- CSMA/CD: carrier sensing, deferral as in CSMA
  - collisions detected within short time
  - colliding transmissions aborted, reducing channel wastage
- collision detection:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: received signal strength overwhelmed by local transmission strength
Three questions

- What’s the difference between IP address and MAC address?
- If we are using ARP, what should destination address, source address and frame type look like?
- If we are using IP, what should destination address, source address and frame type look like?
Ethernet

• Connectionless and unreliable protocol
  • Why doesn’t Ethernet provide reliable data transfer?
• MAC protocol: CSMA/CD + exponential backoff
  • Can we use CSMA/CD in wireless network?
• Switch-based Ethernet
  • No real broadcast channel anymore
  • Self-learning algorithm: support plug-and-play
    • Differences between routing table, switch table and ARP table?