Distributed Algorithms

Exercise Class 5
Routing

Archana Nottamkandath & Istvan Haller
Routing Algorithms

- Find the shortest path from the initiator to every other node in the network.
8.11. Apply Fredrickson's algorithm with $l=1$ and $l=2$, to find BFS tree rooted in $U$. 

$L = 1$
$L = 2$
Exercise 43, “The simple algo”

- Round1
  - u sends <explore,1> to v and x
  - v and x
    - Make u their parent
    - send back <reverse,true> to u
Exercise 43, “The simple algo”

- Round2
  - U sends <Forward,1> to v and x
Exercise 43, “The Simple Algo”

- W gets the <explore,2> from v and makes v its parent
- W gets <explore,2> from x
- W sends back <reverse, false> to x

- V receives
  - <Reverse,true> from w
  - <explore,2> from x
  - Sends <reverse,true> to u
Exercise 43, “The Simple Algo”

- X
  - Receives <explore,2> from v and <reverse,false> from w
  - Sends <reverse,false> to u
- V
  - Receives <explore,2> from x and <reverse,true> from w
  - Sends <reverse,true> to u

(new node added)
Exercise 43, “The Simple Algo”

- Round 3
  - U sends \(\text{<forward,2>}\) to \(v\)
  - V sends \(\text{<forward,2>}\) to \(w\)
Exercise 43, “The Simple Algo”

- U terminates after receiving <reverse,false> from v
Exercise 43, with $l=2$

1. <explore, 1>

2. <explore, 2>

1. <explore, 1>

2. <explore, 2>

2. <explore, 2>
Exercise 43, with $l=2$

- $W$ makes $v$ its parent after getting $<\text{explore},2,1>$ from $v$ first
Exercise 43, with \( l=2 \)

- V receives <explore,2,1> from x and <reverse,2,true> from w.
- V sends <reverse, 1, true > to u
- X receives <explore,2,1> from v and <reverse,2,false> from w.
- X sends <reverse,1,false> to u
- End of round 1
Exercise 43, with l=2

- Round2
- u sends <forward,2> to v
- v sends <forward,2> to w
- w sends back <reverse,false> to v
- v sends <reverse,false> to u
Ex 8.12

• In Fredrickson’s algorithm consider a process at k hops from the initiator with \( k! = \ln n \) for any \( n \). Argue that this process is guaranteed to receive a message \(<\text{reverse}, k+1, _>\) or \(<\text{explore}, j>\) with \( j \leq \{k, k+1, k+2\} \) from all its neighbors.
Ex 8.12

- In Fredrickson’s algorithm consider a process at $k$ hops from the initiator with $k \neq \ln n$ for any $n$. Argue that this process is guaranteed to receive a message $<\text{reverse}, k+1, \_>$ or $<\text{explore}, j>$ with $j \leq \{k, k+1, k+2\}$ from all its neighbors.
Ex 8.12

• Argue that this process is guaranteed to receive a message <reverse,k+1,>_ or <explore,j> with j ∈ {k,k+1,k+2} from all its neighbors.

• 1. Each neighbour at k-1 is guaranteed to send <explore,k> to q.
Ex 8.12

- Argue that this process is guaranteed to receive a message \(<\text{reverse}, k+1, _>\) or \(<\text{explore}, j>\) with \(j \in \{k, k+1, k+2\}\) from all its neighbors.

- 2. Since \(k! \neq \text{ln}\) for any \(n\) (round is not over), each neighbor at \(k\) is guaranteed to send \(<\text{explore}, k+1>\) to \(q\).
Ex 8.12

• Argue that this process is guaranteed to receive a message 
  <reverse,k+1,_> or <explore,j> with j £ {k,k+1,k+2} from all its 
  neighbors.

• Suppose neighbor r at depth k+1 does not send <reverse,k+1,true> 
  to q.

• r must have received <explore,K+1> from p!=q

• If k+1 != ln(more rounds), then r will send <explore,k+2> to q
Ex 8.12

• Argue that this process is guaranteed to receive a message 
  \( <\text{reverse},k+1,\_> \) or \( <\text{explore},j> \) with \( j \leq \{k,k+1,k+2\} \) from all its neighbors.

• Suppose neighbor \( r \) at depth \( k+1 \) does not send \( <\text{reverse},k+1,\text{true}> \) to \( q \).

• \( r \) must have received \( <\text{explore},K+1> \) from \( p!\neq q \)

• If \( k+1 = \ln \) (then end of round), then \( r \) will send \( <\text{reverse},k+1,\text{false}> \) to \( q \)
Exercise 8.13

- Give computation of Fredrick’s algorithm on undirected ring of size 3 with l=2, to show that forward can be sent to a node that is not a child of the sender.
Exercise 8.13

Root sends <explore,1,2> to p and q
Exercise 8.13

q sends <explore,2> to p

Dist=1
(Root=parent)
Exercise 8.13

q becomes p’s parent
Exercise 8.13

q sends <reverse,1,true> to root
At same time p received <explore,1,2>
Exercise 8.13

At same time \( p \) received \(<\text{explore},1>\), made root its parent and send \(<\text{reverse},1,\text{true}>\)

\( p \) sends \(<\text{explore},2>\) to \( q \)
Exercise 8.13

Root receives <reverse,1,true> from p and q, starts new round, sends <forward,2> to p and q
Exercise 8.13

p sends back \(<\text{reverse},\text{false}>\) (**p had already received <explore, 2> from q in previous round**) but q thinks p is still child of q and sends \(<\text{forward},\text{2}>\) to p

p purges this forward message

![Diagram](image)
Exercise 8.13

q finally receives $<\text{explore,2}>$ from u and realizes p is not child

\[
\begin{aligned}
\text{Dist}=1 \\
\text{Root}=\text{parent} \\
\end{aligned}
\]

\[
\begin{aligned}
\text{Dist}=1 \\
\text{Root}=\text{parent} \\
\end{aligned}
\]
Controllers

- Generation of packets when buffers are empty
- Forward the packet when buffer space is available at the next node
- Consumption of the packet at the destination
Exercise 8.17

• Show that there does not exist a deadlock free controller that uses only one buffer place per node and allows each node to send packets to at least one other node.
Exercise 8.17

• Hint: Any controller must allow the generation of packets at a node if the buffers at this node are all empty.
• Consider initial configuration where buffers are all empty.
Destination Controller

- Each process has $N$ buffer slots from 0 to $N-1$. $i$th buffer slot at process $q$ used to mimic sink tree $T_i$
Exercise 8.18

• Show that destination controller is not deadlock-free if packet routing is as follows.
• Packets from P1 -> q path:p1,r1, p2....q
• Packets from p2 to q routed via path: p2, r2, p1....q
Exercise 8.18

P1 generates K1 for q
K1: forwarded from P1 -> r1
P1 generates K2 for q
P2 generates K3 for q
K3: forwarded from P2 -> r2
P2 generates k4 for q
Deadlock occurs as buffers for destination q are full
Buffers for q are full!!
Acyclic Orientation Cover

- An acyclic orientation of an undirected network $G$ is a directed, acyclic network obtained by directing all the edges of $G$
- $G_0, G_1, \ldots, G_{n-1}$ form the acyclic orientation covers of set of paths $P$ in $G$
- Each node has $n$ buffers from 0 to $n-1$
- Let $vw$ be an edge in $G_i$. Each $i$th buffer place of $v$ linked to $i$th buffer place of $w$
- Else if $i < n-1$, $i$th buffer place of $w$ linked to $(i+1)$th buffer place of $v$
Exercise 8.19

- Argue that the acyclic orientation cover of a ring of size 6 covers all shortest paths in this ring.
Exercise 8.19

• Consider each pair of distinct nodes $^6P_2 = 30$
• For some of the pairs, all 3 acyclic orientations in cover are needed to construct shortest path between these nodes
Exercise 8.21

• Given the undirected cube, prove that there is an acyclic orientation cover G0, G1 such that between every two nodes in the cube, there is a minimum-hop path that is concatenation of paths in G0 and G1.
Exercise 8.21

• Verify for each pair of nodes \((u,v)\) some minimum-hop path from \(u\) to \(v\) in cube can be obtained as concatenation of a path in \(G_0\) and a path in \(G_1\).