DISTRIBUTED ALGORITHMS 2014

SEMINAR

Anastasija Efremovska and István Haller

Vrije Universiteit Amsterdam

November 13, 2014
Exercises for today:

- 9.1 - 9.4 (chapter Election - Special networks)
- 9.5 - 9.6, 9.8-9.11 (chapter Election - Minimum spanning tree)
Consider the Chang-Roberts algorithm on a directed ring of size $N$, under the assumption that every process is an initiator. For which distribution of id’s over the ring is the message complexity minimal, respectively maximal, and exactly how many messages are exchanged in these cases? Argue why no distribution of id’s gives rise to fewer or more messages.
Reminder: Chang-Roberts Algorithm

- Works with directed rings
- Initiators send message tagged with their id
- Active process receiving a smaller id message purges it
- Process receiving larger id message becomes passive
- Passive processes passes messages
- A process becomes leader it receives back original message (nobody purged it)
- $O(N^2)$
Give an initial configuration of a directed ring of size \( N \), with every process an initiator, for which the Dolev-Klawe-Rodeh algorithm requires only 2 rounds. Also give an initial configuration for which the algorithm requires \( \lfloor \log N \rfloor + 1 \) rounds.
REMINDER: DOLEV-KLAWE-RODEH ALGORITHM

- Works with directed rings and is round based
- Compare the id of process q with its neighbors p and r
- Due to direction only r can make the comparison for q
- If q has the largest id of the 3, r remains active and takes over the id of q for next round
- Leader found when only a single process is active (2 neighbors are equal)
- Original owner of id becomes leader
- $O(N \log N)$
Give a computation of the tree election algorithm on the network from Example 9.6 in which eventually the processes 1 and 4 have each other as parent.
Reminder: Tree Election Algorithm

- Broadcast based wake-up phase: process wakes up when received message from all neighbors, but performs broadcast when first hit by another one.
- Similar to Tree Algorithm in Waves, with maximum operator applied on children and forwarded to parent, but does not end at decision nodes.
- Decision nodes forward result towards children, everyone receives also message from parent.
- Leader will be process receiving its own id from its parent.
- A total of 2 messages on each edge.
Wakeup phase does not matter
- Requires a computation of the tree algorithm where 1 and 4 are the decision nodes
- Parent is last neighbor to send message, postpone messages on edge 1-4
(A) Show that for undirected networks with a diameter $D > 1$, the time complexity of the tree election algorithm (including the wake-up phase) is at most $2D$.

(B) For $D = 2$, give an example where the tree election algorithm takes four time units to terminate.

(C) Give an example to show that if $D = 1$, the tree election algorithm may take three time units to terminate.
Exercise 9.5

Perform the Gallager-Humblet-Spira algorithm on the following undirected network:
Reminder: Kruskal’s Algorithm

- Single-processor algorithm for minimum spanning tree
- Greedy algorithm, selects the smallest possible valid edge
- Start with forest consisting of individual nodes as trees
- An edge is valid if it connects two different trees in the current state (no cycle)
- Adding new edges makes other edges invalid, thus removed from the algorithm
- When there is a single tree (no more valid edges), the algorithm is finished
Processes on core edge broadcast `<initiate>` setting everyone to `find` and start evaluating the outgoing edges

`<test>` message sent across every unclassified outgoing edge, blocked until receiving process belongs to a `>=` fragment

Once every node receives first `<accept>` response or has no more edges, the information can be centralized and the processes change status to `found`

Smallest edge named branch and `<connect>` sent across it
REMINDER: GALLAGER-HUMBLET-SPIRA ALGORITHM

- `<connect>` to larger fragment always succeeds and receives `<initiate>` as response
- `<connect>` to same level fragment only successful if they select same branch edge (exchange `<connect>`s)
- In latter case they also exchange `<initiate>` with new fragment name and level, setting the branch edge as the core
- Terminates when no more outgoing edges to discover
Exercise 9.6

Argue that in the Gallager-Humblet-Spira algorithm, any fragment at a level $L$ always contains at least $2^L$ processes.
Induction on fragment size

If $L = 0$ it contains 1 node

Level increased by one when joining two equal level fragments

Number of processes in fragment of level $L+1 \geq 2 \times$ number of processes in fragment level $L$
Suppose a process reported a lowest-weight outgoing basic edge, and next receives a message \texttt{initiate<find>}. Explain by means of a scenario why it must test again whether this basic edge is outgoing.
initiate triggered only on fragment change.

Some edges become rejected because their destination became a member of the fragment.

Example built using two equal-sized fragments with multiple edges between them.
Suppose that, at some point in the Gallager-Humblet-Spira algorithm, a process $p$ receives a message $<\text{test}, FN, L>$ through channel $pq$, where $p$’s fragment has a different name than $FN$ and at least level $L$. Explain why $p$ can send an accept message to $q$, without fear that $p$ and $q$ are in the same fragment.
Fragment levels and names are updated at the same time.

*test* only sent using the latest fragment name/level.

Reductio ad absurdum

- Suppose that $p$ and $q$ are in the same fragment.

Since $q$ already has the latest update, only $p$ can change.

Changes to the fragment name of $p$ require level increase.

Thus the level of $p$ should be $<$ than the level of $q$.

Contradiction with the problem statement.
Consider the following scenario for the Gallager-Humblet-Spira algorithm. In a fragment \( F \) with name \( FN \) and level \( L \), the lowest-weight outgoing edge of the fragment has just been reported to the core edge. Concurrently a fragment \( F' \) with name \( FN' \) and level \( L' < L \) connects to this fragment. How can we be sure that \( F' \) can’t have a lower-weight outgoing edge than the one just reported to the core edge of \( F \)?
HINT

- $FN'$ sent the connect across $pq$, $pq$ is lowest weight edge of $FN'$
- $q$ finished testing, thus it never sent test to $q$
- Thus there is edge $qr$ with lower weight than $pq$
Give an example to show that the Gallager-Humblet-Spira algorithm could get into a deadlock if different channels were allowed to have the same weight.
If all fragments have the same size, they can issue concurrent `connect`.

By selecting different edges, neither can finish the `connect`.

Cycle of `connect` messages possible on set of equal-sized edges.