Distributed Algorithms 2014 Seminar

Anastasija Efremovska and István Haller

Vrije Universiteit Amsterdam

November 6, 2014
Exercises for today:

- 8.4 - 8.5 (chapter Routing - Merlin-Segall (All shortest paths to initiator))
- 8.6 - 8.10 (chapter Routing - Toueg (All pair shortest path))
- 8.16 (chapter Routing - Dijkstra (All shortest paths to initiator))
Centralized algorithms for shortest path towards initiator

**Chandy-Misra**
- Each node broadcasts its distance towards initiator whenever it is updated
- When a message is received, it checks to update the current distance
- Potential spam of messages (incremental updates)

**Merlin-Segall**
- Organized into rounds to accumulate updates
- Single broadcast per round → At message from parent
- Round finishes when all messages received and message sent to parent (similar to echo)
- Distances updated continuously
- Parent only updated at end of round
Suppose that in the Merlin-Segall algorithm a node $v$ would update $next_v$ each time it updates $dist_v$. Explain why the worst-case message complexity would become exponential.
Exercise 8.4

Run the **Merlin-Segall** algorithm on the following undirected weighted network, to compute all shortest paths toward node $y$. Give an execution that takes four rounds before the correct sink tree has been computed.
Reminder: Merlin-Segall Algorithm

- Centralized algorithm to compute shortest paths to $u_0$.
- Initial sink tree with root $u_0$ and $\text{dist}_v(u_0) = \infty$ for all $v \neq u_0$.
- Each round $u_0$ sends $<0>$ to its neighbor.
- Let node $v$ receive $<d>$ from node $w$.
  - If $d + \omega_{wv} < \text{dist}_v(u_0)$, then $\text{dist}_v(u_0) = d + \omega_{wv}$, also save $w$ for $\text{next}_v(u_0)$ in next round.
  - If $w = \text{next}_v(u_0)$ (from $u_0$ along sink tree), then broadcast distance to all neighbors except $\text{next}_v(u_0)$.
- Wait for message from all neighbors before sending distance to $\text{next}_v(u_0)$ and ending round (echo-like to ensure separation between rounds).
- Algorithm terminates in $N - 1$ rounds.
Run Toueg's algorithm on the following undirected weighted network. Take as pivot order: $u \ v \ w \ x \ y$. 
**Reminder: Toueg’s Algorithm**

- Compute for each pair $u, v$ a shortest path from $u$ to $v$
- Each node maintains $dist_v$ and $next_v$ for all other nodes
- No initial sink-tree (built up progressively)
- Incrementally add intermediate nodes as pivots
  - Pivots selected uniformly in the same order
  - Nodes request distance vector of current pivot
  - Pivot broadcasts its routing table along sink tree
  - For each pivot $w$ udpate distance vector:
    $$d^{S \cup \{w\}} = \min \{d^S(u, w) + d^S(w, v), d^S(u, v)\}$$
  - Pivot round terminates after distance update and propagation of pivot routing table
- Pointers in sink tree consist of processed pivots
Argue that Toueg’s algorithm is an all-pairs shortest path algorithm.
Exercice 8.8

Analyze the space complexity of Toueg’s algorithm.
In Toueg’s algorithm, when a node $u \neq w$ in the sink tree of the pivot $w$ receives the distance values of $w$, let $u$ first perform for each node $v$ the check whether $\text{dist}_u(w) + \text{dist}_w(v) < \text{dist}_u(v)$. Explain why $u$ only needs to forward those values $\text{dist}_w(v)$ for which this check yields a positive result.
Suppose that edges can carry negative weights. Explain how the output of Toueg’s algorithm can be used to detect the presence of a negative-weight cycle of at least two edges.
Develop a distributed version of Dijkstra’s celebrated single-source shortest path algorithm. Discuss the worst-case message and time complexity of your algorithm.
**Reminder: Dijkstra’s Algorithm**

- Single source shortest path algorithm
- Extension of Breadth First Search with weighted edges
- Basic scheme: expand closest unvisited node first
- Logic: any component of optimal path is also optimal