Networks and Graphs lecture 8

routing with tree graphs

in a communication network each node(vertex) keeps a **routing table** of where to send messages. these messages should be **traversing along the spanning tree rooted at the destination**.

the **shortest path** for any message is not necessarily along the MST but it is of the **sink tree** of the **destination**. to get the **shortest path for all destinations** there would have to be computed a **sink tree for all vertices**.

Dijkstra's algorithm

the goal of the algorithm is to compute a sink tree rooted at a vertex *u*.

1. assume a map of the distance and set it to; of itself 0[$d(u)$] and all other nodes at ∞[$d(v)$].
2. also assume a map of the neighbors of the node where the message should be forwarded to reach *u*. initiate the map with all vertices having *n(v)* = ⊥.
3. start with the set S = {u} and H = {all other nodes}.
4. **extend S** with the reachable neighbors of the set S that has the **smallest *d*-value** and set their distances.
5. if at some point a **distance to any node** is found that is **less than the already known value** it is **updated**.

this algorithm has a complexity of O(n^2) .

> *Dijkstra's algorithm may not compute a correct sink tree if the graph has negative weights.*

Bellman-Ford algorithm

this algorithm also works if some of the weights have a negative value to compute a sink tree rooted at vertex *u*.

1. assume a map of the distance and set it to; of itself 0[$d(u)$] and all other nodes at ∞[$d(v)$].
2. also assume a map of the neighbors of the node where the message should be forwarded to reach *u*. initiate the map with all vertices having *n(v)* = ⊥.
3. the difference with Dijkstra's algorithm is that the next steps are computed in a **for loop** where the upper most case is **n-1**.
4. for every for loop **all arcs are considered** whether the distance of the **vertex at the out end of the arc** has a distance **more** than that of ***d(u,v) + w(arc)***. if this is the case update it.
5. repeat for **n-1** steps. or after no values has been updated.
6. after check for **each arc** if the **weight of the arc plus the distance to *v*** is less than the **distance of x** then there is a **negative weight cycle** and the program

exits with a **error**.

this algorithm has a complexity of O(nm)

Internet routing

the most used routing algorithms are based on the two algorithms above.

- **link-state routing** uses the Dijkstra, link state packets are exchanged by **all routers** periodically.
- **distance-based routing** uses Bellman-Ford, this algorithm notifies **only its neighbor nodes** instead of flooding the network.