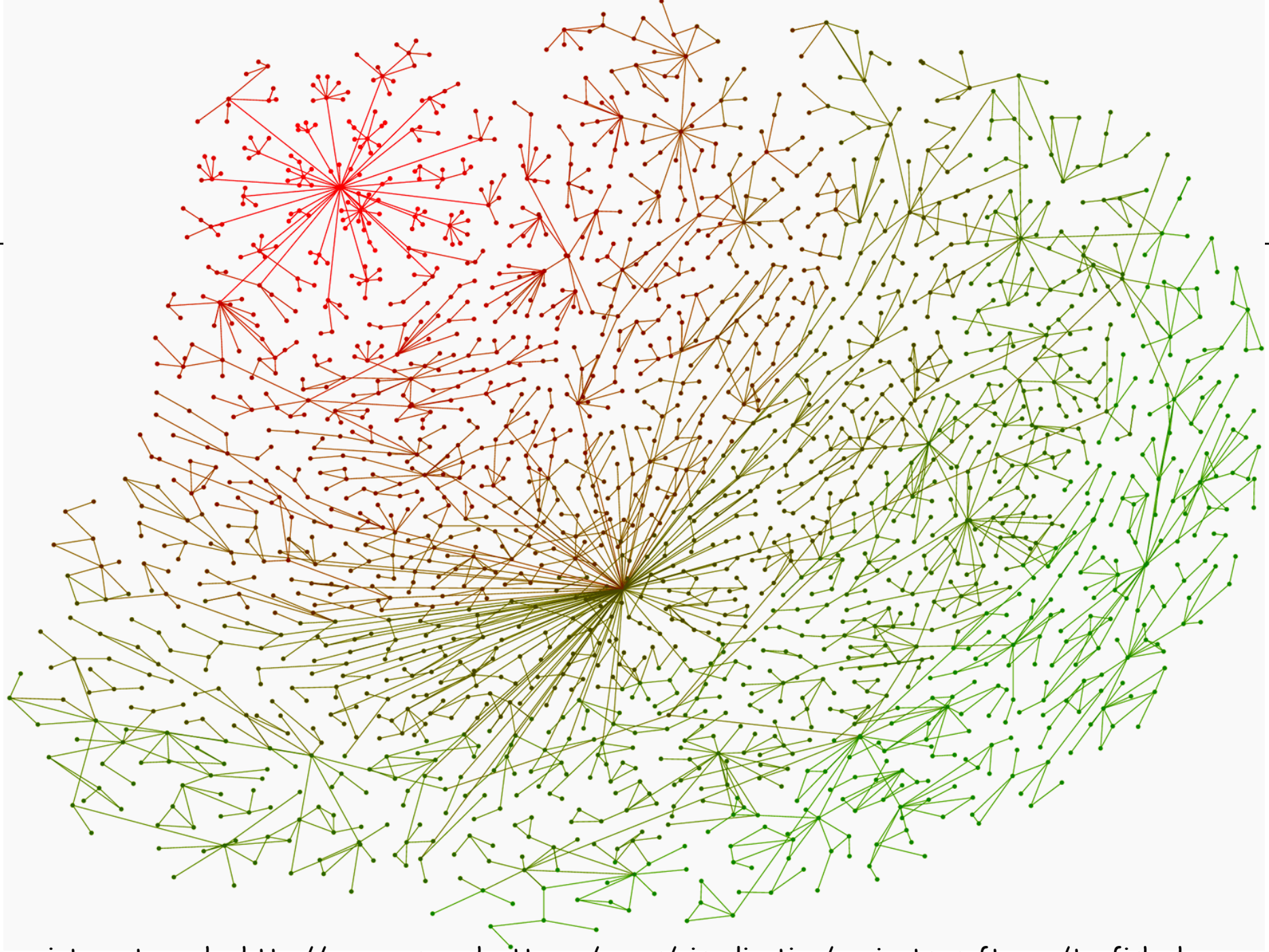
6.9 Distance Vector Protocol



internet graph , http://www.research.att.com/areas/visualization/projects_software/topfish.php

Distance Vector Protocol

Communication network.

- Nodes ≈ routers.
- Edges ≈ direct communication link.
- Cost of edge ≈ delay on link. ← nonnegative, but Bellman-Ford used anyway!

Problem. Packets need to be transmitted. What is the shortest path to their destination?

Q. Which of the previous shortest-path algorithms can we easily adapt to such a distributed setting (and how)? (1 min)

TU Delft

Distance Vector Protocol

Communication network.

- Nodes ≈ routers.
- Edges ≈ direct communication link.
- Cost of edge ≈ delay on link. ← nonnegative, but Bellman-Ford used anyway!

Problem. Packets need to be transmitted. What is the shortest path to their destination?

Dijkstra's algorithm. Requires global information of network.

Bellman-Ford. Uses only local knowledge of neighboring nodes.

Synchronization. We don't expect routers to run in lockstep. The order in which each `foreach` loop executes in not important. Moreover, algorithm still converges even if updates are asynchronous.

TU Delft

Asynchronous Shortest Path Implementation

```
Asynchronous-Shortest-Path(G, s, t) {
    foreach node v ∈ V {
        M[v] ← ∞
        predecessor[v] ← ϕ
    }

    M[s] ← 0
    make s active
    while there is an active node {
        choose an active node w
        foreach node v such that (w, v) ∈ E {
            if (M[v] > M[w] + c_wv){
                M[v] ← M[w] + c_wv
                predecessor[v] ← w
                make v active
            }
        }
        w becomes inactive
    }
    return M[t]
}
```

TU Delft

Distance Vector Protocol

Distance vector protocol.

- Each router maintains vector of shortest path lengths to every other node (distances) and the first hop on each path (directions).
- Algorithm: each router performs n separate computations, one for each potential destination node.
- "Routing by rumor/gossip." (Determine value based on neighbors.)

Ex. RIP, Xerox XNS RIP, Novell's IPX RIP, Cisco's IGRP, DEC's DNA Phase IV, AppleTalk's RTMP.

TU Delft

Distance Vector Protocol Implementation

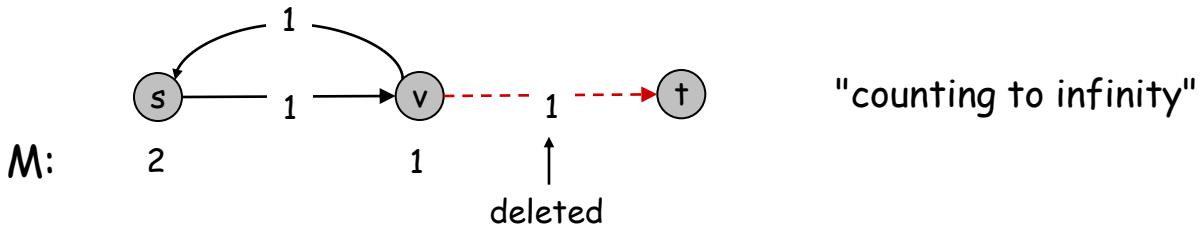
```
DVP-for-node(G,w) {
    foreach node s ∈ V {
        M[s] ← ∞
        predecessor[s] ← ϕ
    }
    M[w] ← 0
    foreach node v such that (w,v) ∈ E {
         send( update(w, w, M[w]) )
    }
    while wait-for( update(v, s, d) ) {
        if (M[s] > d + c_wv){
            M[s] ← d + c_wv
            predecessor[s] ← v
            foreach node v such that (w, v) ∈ E {
                send( update(w, s, M[s]) )
            }
        }
    }
}
```

send update(v,s,d) messages:
"you can now reach s via me (v) with distance d"

Distance Vector Protocol

Caveat. Edge costs may change during algorithm (or fail completely).

$\Rightarrow$ Inform neighbors.



node v needs to update its distance vector since directly to t is impossible:

- go via s, cost M[s]+1 = 3
- tell neighbor(s) that its cost is updated (increased)
- costs for s become M[v]+1 = 4
- etc.

This is solved by keeping track of the whole path for each destination:

path vector protocols

Path Vector Protocols

Link state routing.

not just the distance and first hop

- Each router also stores the entire path.
- Avoids "counting-to-infinity" problem and related difficulties.
- Stores complete subnetwork structure.
- Based on Dijkstra's algorithm.

Ex. Border Gateway Protocol (BGP), between subnetworks
Open Shortest Path First (OSPF), within a subnetwork.

TU Delft