ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Communication Systems

Shortest Path, Distance Vector

University of Freiburg
Computer Science
Computer Networks and Telematics
Prof. Christian Schindelhauer

CoNe
Freiburg

IIF
INSTITUT FÜR
INFORMATIK
FREIBURG

Copyright Warning

- This lecture is already stolen
- If you copy it please ask the author
    - Prof. Dr. Gerhard Schneider
- like I did

Internet Protocol
the Universal Service

- IP routing
  - Routing decision is renewed for every packet (introduction to static IP routing last lecture)
  - No state of previous routing decisions is kept (!)
- By now: Introduced IPv4 static / manual routing setup
  - Rather laborious and error prone on a larger scale level
  - Repeated updates of routing tables on many routers if a new network is attached or the network layout is modified
- These mechanisms not suitable for routing on larger scale
  - Campus-wide inter LAN routing
  - DFN-wide, inter-provider-routing, ...

Routing Protocols in Packet Networks

- Internet doesn't have very predictable traffic flow, may have unreliable links
- Routers are assumed to know
  - address of each neighbor
  - cost of reaching each neighbor
- Choices for Internet routing
  - centralized vs. distributed routing
  - source based vs. hop-by-hop
  - single or multipath
  - dynamic vs. static

Routing Strategies – (non) adaptive Routing

- Routing algorithms are grouped into two major classes
- Nonadaptive RA do not base their routing decisions on (continuous) measurements or estimates of current bandwidth usage and topology
  - no need for specific measurement service run continuously or scheduled
- The routes to use are computed in advance, off-line and downloaded to routers when network is coming up
- That is the typical scenario for networked end systems – normally the system administrator provides the routes during machine setup
- Or the routing information is transferred via DHCP (centralized setup of networking resources)

Adaptive Routing

- Routing done that way often named static (type of routing discussed yet falls into that category)
- Adaptive algorithms change their routing decisions to reflect changes in traffic/bandwidth usage and topology
- Algorithms differ in where they get their information ...
  - Locally from own measurements or from adjacent routers
  - Or (globally) from all routers
- ... and when changes are executed
  - Every ΔT seconds when network load changes
  - Or changes in topology occur
  - Or event driven ...

Routing Algorithms – Routing Mechanisms

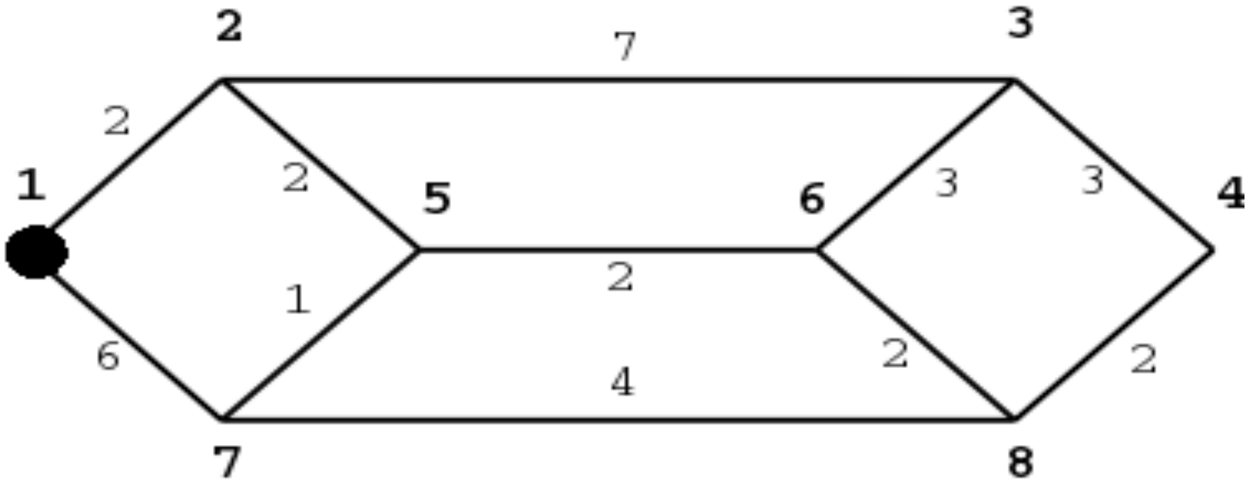
- Distance vector & link state routing
  - distributed algorithms
- Distance vector
  - Tell neighbors about distances to each destination
  - Each nodes computation depends on its neighbors
- Link state (next lecture)
  - Tell all routers distance to each neighbor
  - Each router computes its best paths
- Distance vector uses shortest path
  - Single adaptive cost of a link

Shortest Path Routing

- Routing technique widely used, because it is simple and easy to understand
- Idea: Build a graph of the subnet with each node representing a router and each arc representing a link
- To choose a route between a given pair of nodes the algorithm just finds the shortest path on the graph
- You have to explain the metric used for shortest path measuring:
  - hop count, physical distance, bandwidth, latency, communication costs, mean queue length, ...

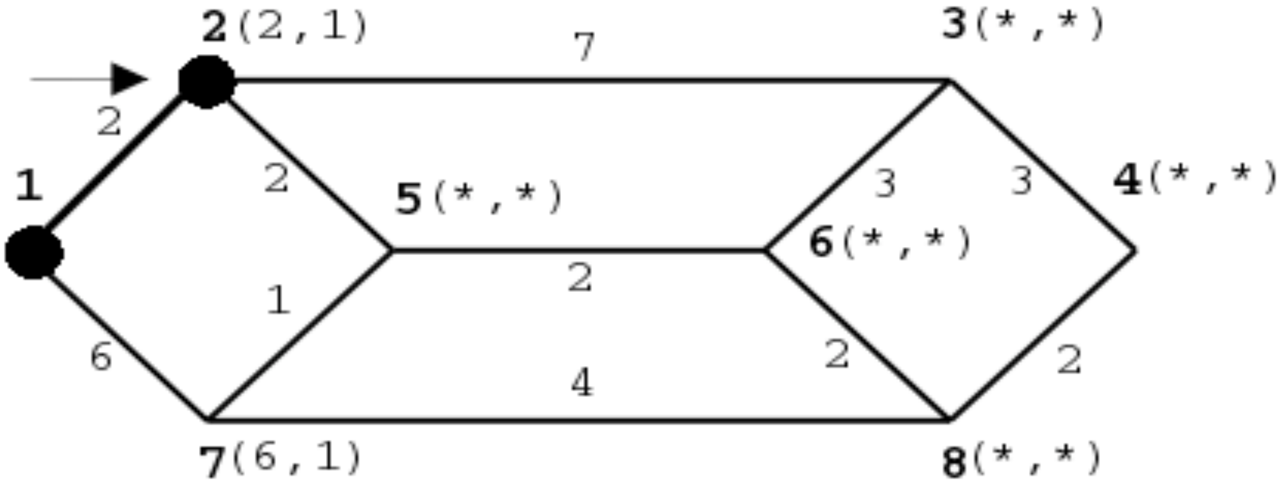
Shortest Path Routing

- Hop count metric has same length for path 1-2-5 and 1-2-3 (nodes are fat numbers, costs smaller numbers between nodes)
- Geographic distance is for 1-2-3 much longer then for 1-2-5 (assuming the graph is drawn in scale)
- For other metrics the weighting function may be computed through hourly test packets sent and computed
- Criteria for computation of metric may combined



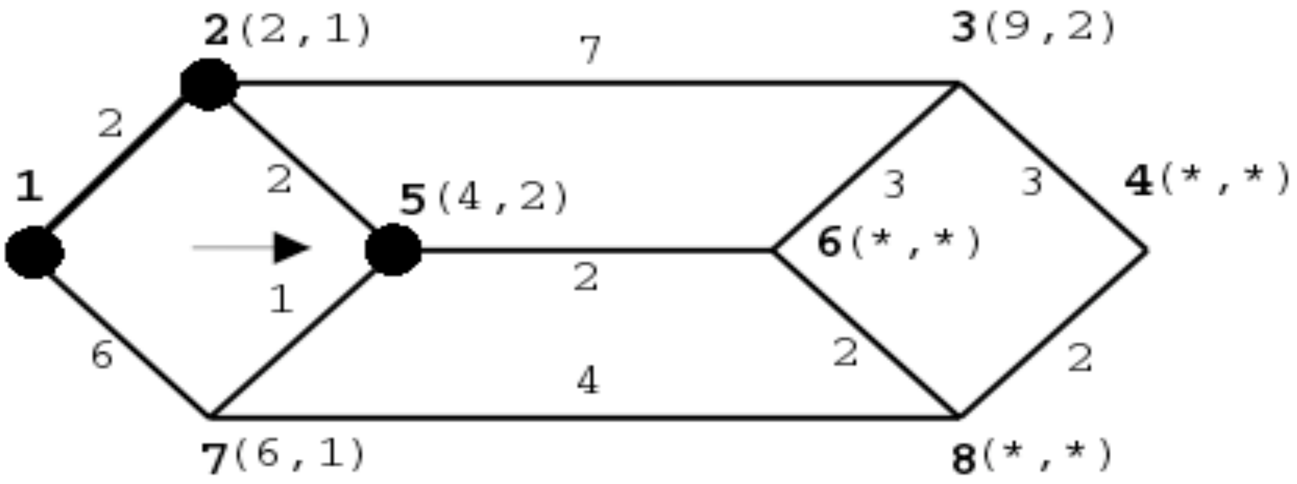
Shortest Path Routing

- One algorithm for computing the shortest path is Dijkstra's
- Each node is labeled with its distance from source node along the best known path
- Initially no paths are known (and labeled accordingly)
- As algorithm proceeds and paths are discovered labels may change reflecting better paths



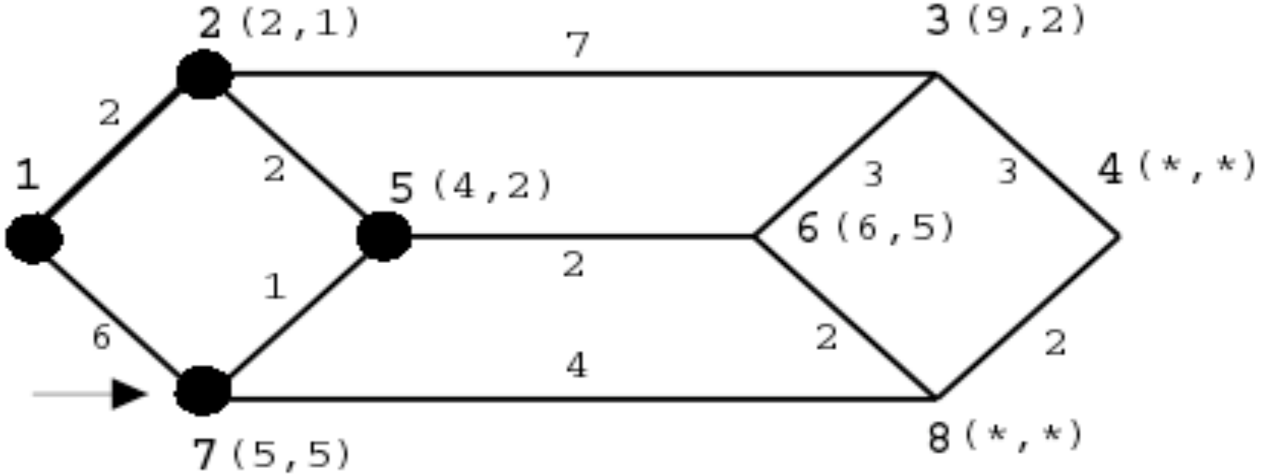
Shortest Path Routing

- The shortest path (sp) from 1 to 4 is searched for
- We are using the geographic distance for computing the sp
- Started from 1 marking it permanent (big dot), then examined the adjacent nodes to 1 (node 2 and 7)
- Whenever a node is (re)labeled the source node we come from is filled in (distance, source node)



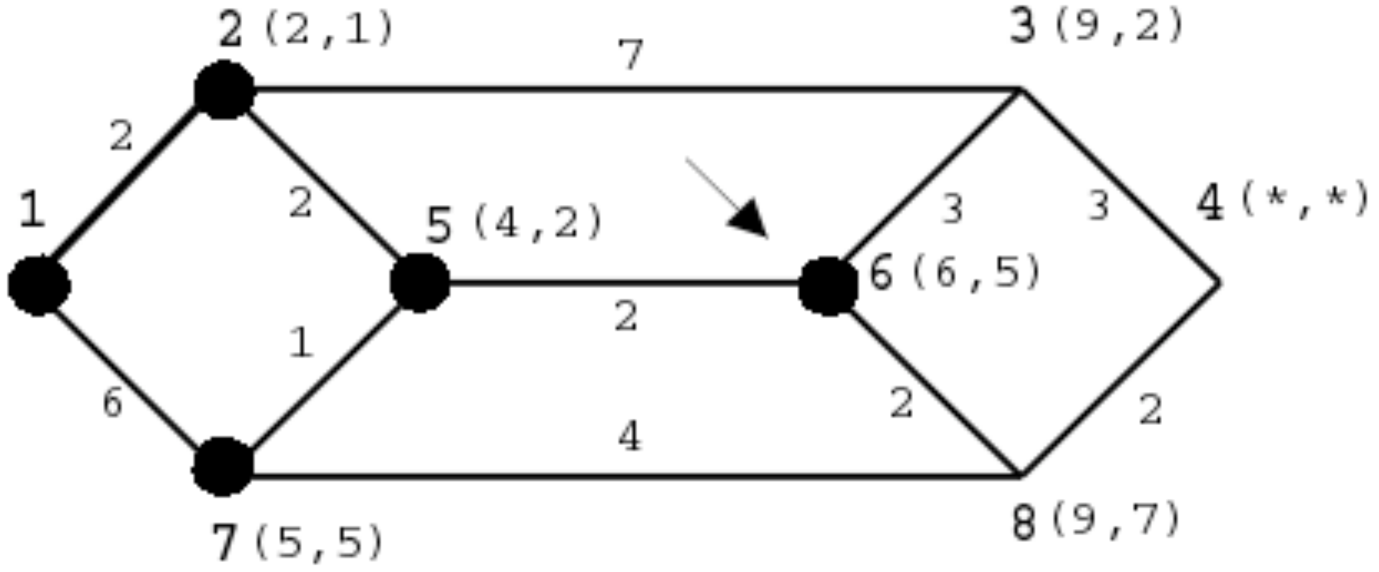
Shortest Path Routing

- When adjacent routes are examined the smallest distance is labeled permanent (node 2)
- In the next step the process is restarted from node 2
- We have to take the sums of the route up to here in to account and we get node 5 (last picture)
- Note: In next step node 7 has to be relabeled { (6,1) -> (5,5) }



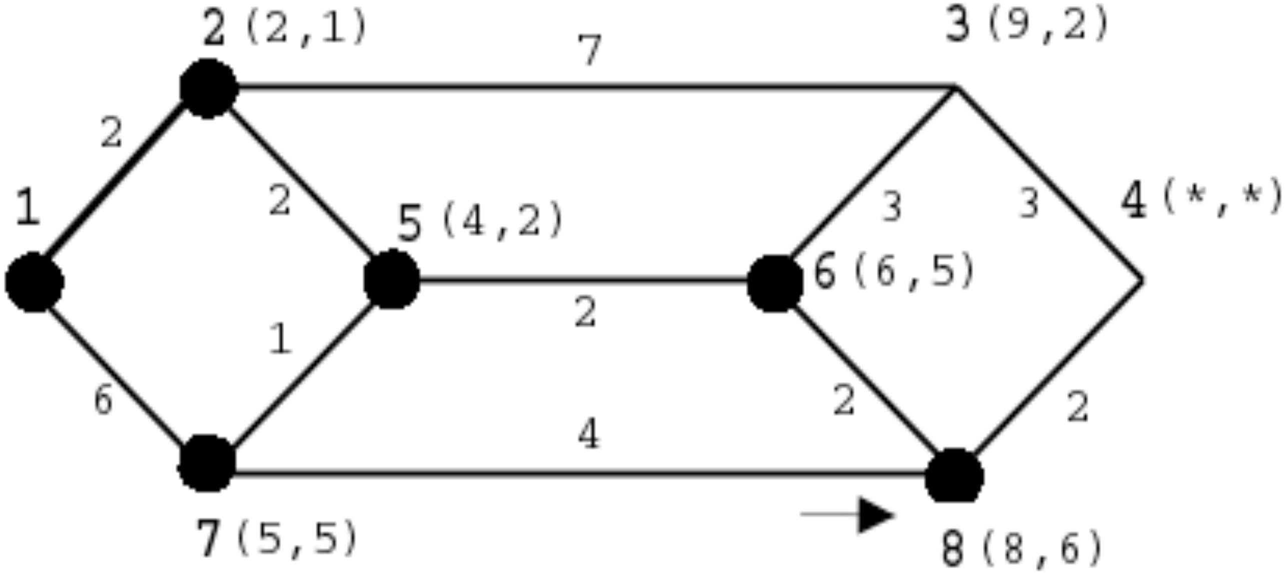
Shortest Path Routing

- Path with 3 hops to 7 has lesser costs than direct with 1 hop
- Next step shows that from node 7 the metric to 8 is higher than from node 5 over 6 to 8
- Node 8 has to be relabeled then (see next slide)



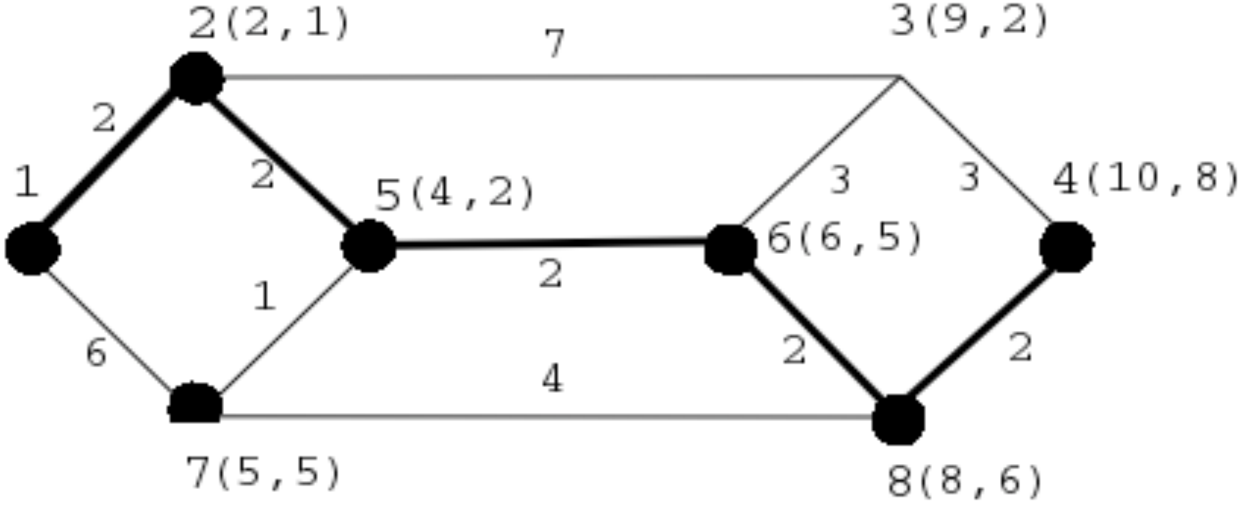
Shortest Path Routing

- Step shows that from node 7 the metric to 8 is higher than from node 5 over 6 to 8 (node 8 has to be relabeled)
- There is a more optimal route to node 8
  - 1 – 2 – 5 – 7 – 8 (4 hops with cost of 9)
  - 1 – 2 – 5 – 6 – 8 (4 hops with cost of 8)



Shortest Path Routing

- For getting the path we start from the destination and get the predecessor from the labels
- In the end we get a route 1 – 2 – 5 – 6 – 8 – 4
- Remember: route optimal in “costs” not in hops
  - Simple hop count routing would prefer 1 – 2 – 3 – 4 (cost of 12) or 1 – 7 – 8 – 4 (same cost) – 2 points higher than route named above



Distance Vector Routing

- For distance vector routing each router maintains a table (called a vector for a given destination – computed with shortest path) delivering the best known distance to each destination
- These tables are updated by regularly exchanging information with neighbors
- Other name of this algorithm is distributed Bellmann-Ford or Ford-Fulkerson
- In distance vector routing each router maintains a routing table containing the pair of each router of the (sub)net and the destination to it
- Entry contains information on outgoing line and the estimate of time, distance to destination

Distance Vector Routing

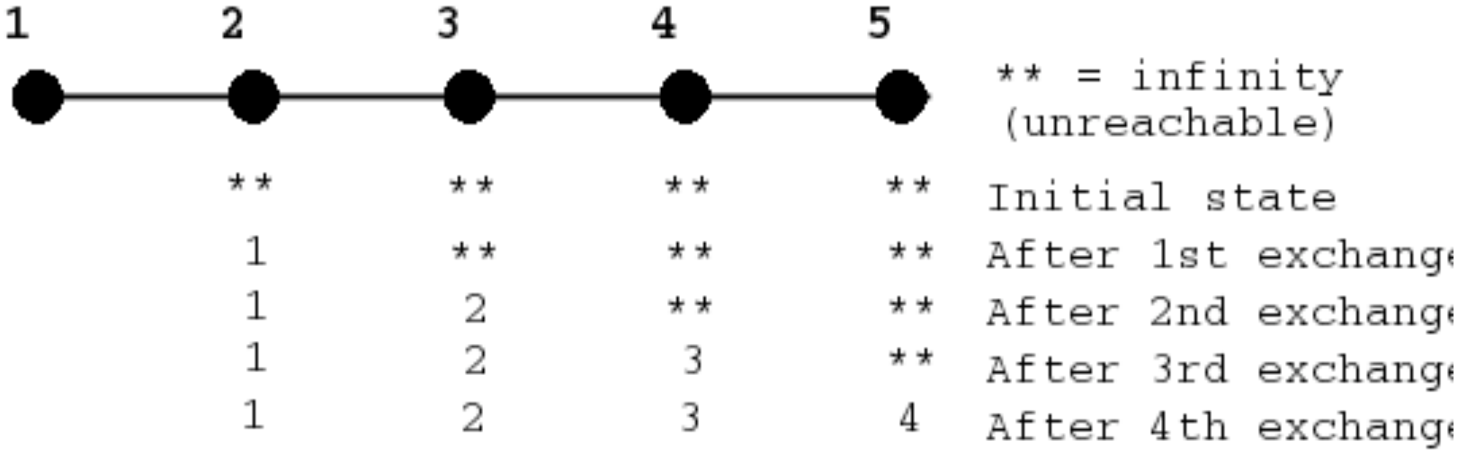
- The metric might be one of the types we named earlier
- Every router should know the distance to its neighbors, with hop count (typically used with RIP – explained next lecture in greater detail) it is just one hop
- Queue length as metric might be computed through simply checking each outgoing line
- For delay the router might ping each neighbor with special ECHO packets and compute the round trip time
- For setting up of the tables each router sends a packet with the list of distances to each router in the (sub)net
- Every router receives such packets and uses them for computing an updated table

Distance Vector Routing

- Each destination might be reachable on different paths, but the router takes the shortest distance from all packets and removes the other information to the same destination
- Such the router computes information on which line which router is reachable
- This mechanism works quite well in theory, but has some drawbacks too ...

Count-to-Infinity Problem

- Although distant vector converges to the correct answer it may do so very slowly
- It reacts fast to positive news but leisurely to bad
- To see how fast good news propagates see the next simple example (five routers on a linear subnet, hops to 1 shown)



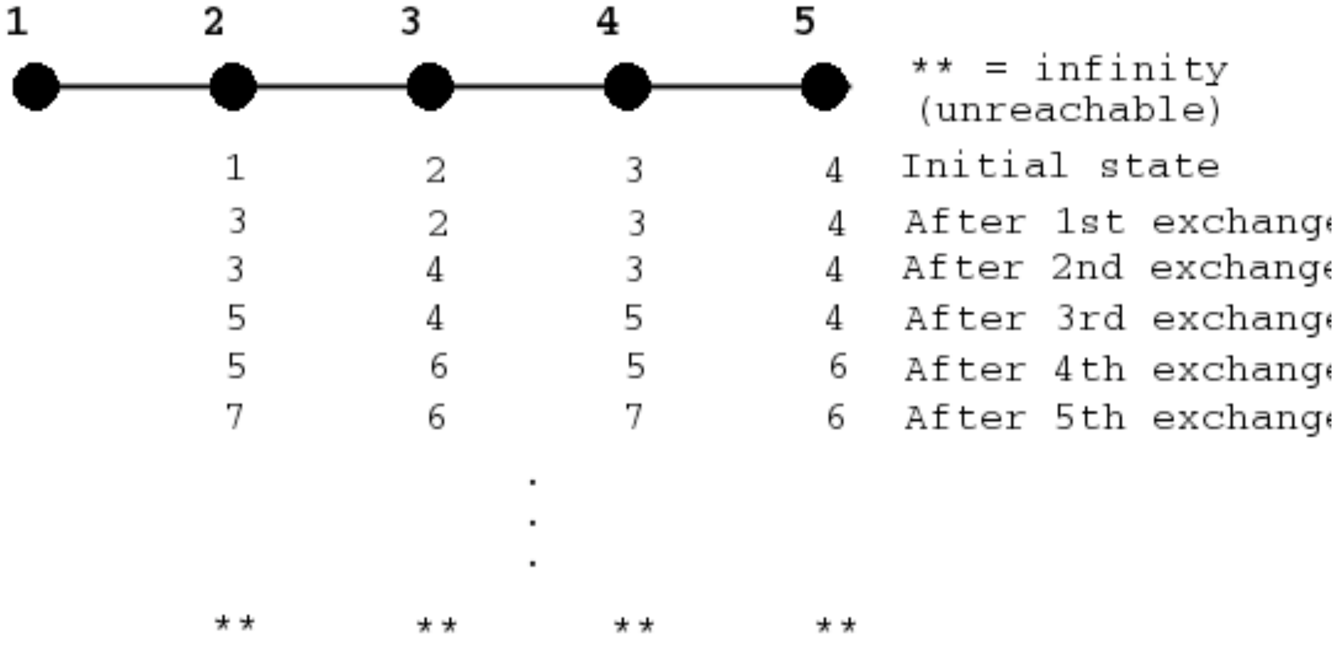
1	2	3	4	5	
	**	**	**	**	Initial state
	1	**	**	**	After 1st exchang
	1	2	**	**	After 2nd exchang
	1	2	3	**	After 3rd exchang
	1	2	3	4	After 4th exchang

Count-to-Infinity Problem

- With 1 down initially no router knows a path to it (in routing terms: infinite route)
- With 1 coming up - at the first exchange (for simplicity: all routing information is exchanged at exact the same moment) 2 gets the good news that 1 has a route of “0” to 1
- 2 adds this information to its table
- In the next round of exchange 2 tells 3 that it knows a route with hop count “1” to 1, 3 learns that and adds the metric “2” for the route to 1
- Nexts steps are accordingly – process ends after the 4th round

Count-to-Infinity Problem

- Now compute the routes for the opposite scenario (router 1 is going down for some reason)



1	2	3	4	Initial state
3	2	3	4	After 1st exchang
3	4	3	4	After 2nd exchang
5	4	5	4	After 3rd exchang
5	6	5	6	After 4th exchang
7	6	7	6	After 5th exchang
		. . .		
**	**	**	**	

Count-to-Infinity Problem

- All lines were initially up, but 1 failed
- The first round of packet exchange 2 hears nothing from 1, but 3 says “no worry, I know a route to 1”
- Hence 2 updates its table to the hop count of “3” for path to 1
- 2 does not know that 3s route runs through itself
- With second exchange router 3 has two entries for 1 with the same metric of “3”, therefore picks one at random and updates its table to a hop count of “4”
- Problem: No one router has a hop count greater than the minimum of all neighbors, gradually they walk up to infinity

Count-to-Infinity Problem

- When router 1 is set “unreachable” depends on the value for infinity
- Therefor the value for infinity should set to the maximum diameter of the network plus “1”
- But even in moderate sized networks the exchanges needed for setting a router “unreachable” may be regarded as to much
- If the metric is a time delay then the upper bound of the infinity value should be set reasonably high, else simple problems within the network (congestions of a short moment, delays in queues, ...) could bring routers out of range
- One of many suggested solutions is the “Split-Horizon-Hack”

ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Communication Systems

Distance-Vector

University of Freiburg
Computer Science
Computer Networks and Telematics
Prof. Christian Schindelhauer

CoNe Freiburg

IIF
INSTITUT FÜR INFORMATIK FREIBURG