Net Centric Systems

Summer Term 2011

http://www.p2p.tu-darmstadt.de/teaching/summer-term-2011/ncs-net-centric-systems/

TECHNISCHE UNIVERSITÄT DARMSTADT

Prof. Dr. Thorsten Strufe
Dipl.-Inform. Andreas Höfer
Peer-to-Peer Networks Group (P2P)

Prof. Dr. Max Mühlhäuser
Dipl.-Inform. Sebastian Döweling
Telecooperation Group (TK)

Exercise 3

Submit until 20/5/2011, 18:00 in moodle (as pdf file).
This exercise is relevant for the exam bonus.

Please note, that by submitting your solution to this exercise, you confirm that you are the exclusive author(s) of the respective material. For additional information, we would like to refer you to: http://www.informatik.tu-darmstadt.de/de/studierende/studium-alt/plagiarismus/

Task 3.1: Network Flows (10 P.)

Ford Fulkerson Algorithm:

```
initialize flow f to 0
while there exists an augmenting path p in the residual network Gf:
      augment flow f along p
return f
```

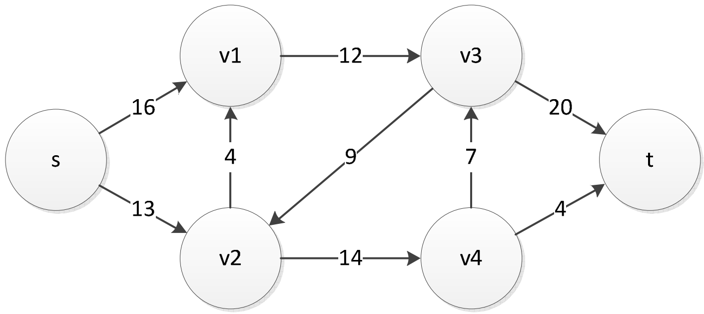

Figure 1: Network flow graph for Task 3.1

a) Using the network flow graph from Figure 1, please calculate the maximum flow through the network with the Ford-Fulkerson algorithm.

- To determine the next step, please use a **depth-first** (search-)strategy to identify the augmenting path.
- For each steps of your calculation, please list the augmented path you chose, the capacity added to the network flow and the respective flow in the network graph with the current flow and capacity of the individual edges (first step as an example below)

- Drawing graphs for each step is **not** required for your solution (however, it might help to understand the algorithm).

Step 1/Example:

Augmenting Path:	S → V1 → V3 → V2 → V4 → T	
Maximum Capacity:	4	
Updated Flow:	S → V1:	4/16
	V1 → V3:	4/12
	V3 → V2:	4/9
	V2 → V4:	4/14
	V4 → T:	4/4

Residual graph and new flow graph:

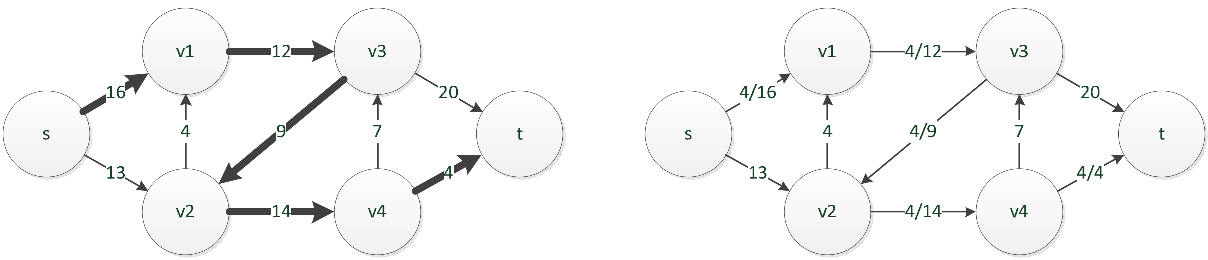

Figure 2: [Step 1] Residual graph on the left (augmenting path with bold connections), new flow graph on the right.

Task 3.2: Distance Vector Routing (17 P.)

a) Assume a network topology as shown in Figure 4.1 and calculate the distance tables after initialization phase (before any update messages have been received or processed). How do the first update messages look like? (2 P.)

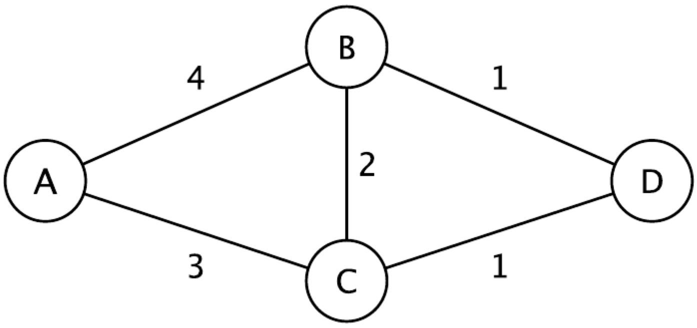

Figure 4.1 Topology of the observed network

b) In the topology (as shown in Figure 4.1), assume the existence of perfectly synchronized timers which force all nodes to send their updates periodically and simultaneously. For each round, until the stable state, give the complete distance tables and denote the updates that each node receives. Note: Updates are sent for new minima, i.e., in the first round for every known distance. (4 P.)

c) What would change when the link cost between A and B decreases to 3? Which *updates* would be sent from and to which neighbor? State new distance tables for all nodes highlighting the changed values. (3 P.)

d) What happens when the link between B and D goes down? (3 P.)

e) Assume that we use distance vector routing with the split horizon technique. What happens when node D goes down? (3 P.)

f) In path vector routing each node sends to its neighbors the (shortest) paths to all nodes (instead of the distances as in distance vector routing). Is count-to-infinity also a problem in path vector routing? Explain your answer briefly. (2 P.)