Sample Solution for Exercise Communication Networks I

TECHNISCHE UNIVERSITÄT DARMSTADT

Prof. Dr.-Ing. Ralf Steinmetz
Multimedia Communications Lab
Institut für Datentechnik
Fachbereich Elektrotechnik und Informationstechnik
Fachbereich Informatik (Zweitmitglied)

Published at:	12.06.2015
Tutorial date:	18.06.2015

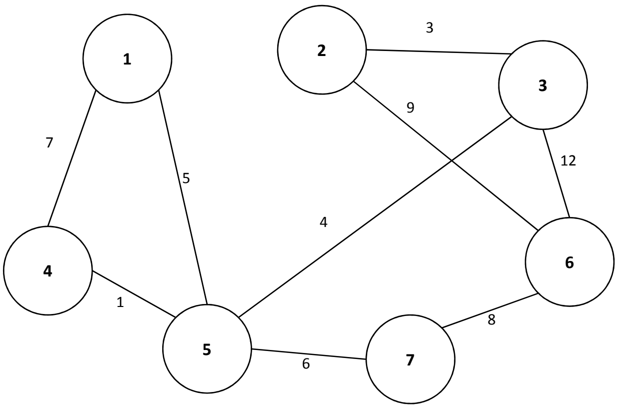
General Remarks

Welcome to the exercise for Communication Networks I. Please adhere to the following general remarks regarding the organization of the exercise during this summer term.

- One week before the tutorial, a new exercise will be published at the Exercise area of the KN1 Moodle (`https://moodle.tu-darmstadt.de/course/view.php?id=5268`)
- The exercise serves as your hands-on experience in addition to the lecture and as a preparation for the exam
- The questions in the exercise can be discussed at the tutorial date
- The sample solution for the exercise is available at the Exercise area of KN1 Moodle in addition to the corresponding tutorial. Nevertheless, we encourage students to try to solve the exercise themselves before the tutorial date without looking into the solution as a good practice to understand the subject of the lecture

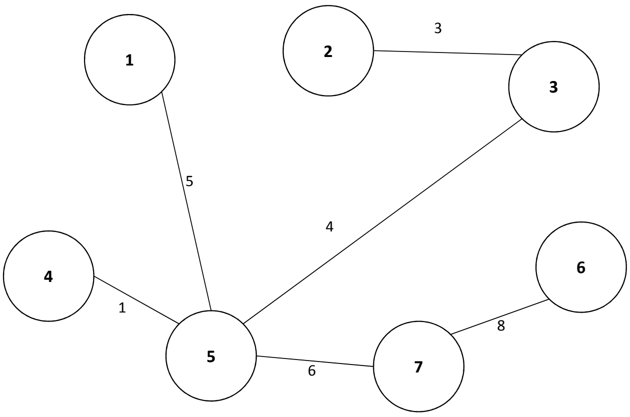
Problem 1 - Kruskal's Algorithm

a) Calculate the length of the shortest spanning tree using Kruskal's Algorithm. Draw the final graph and show your results in the table.



Edge	Length	Choice
(4,5)	1	1
(2,3)		

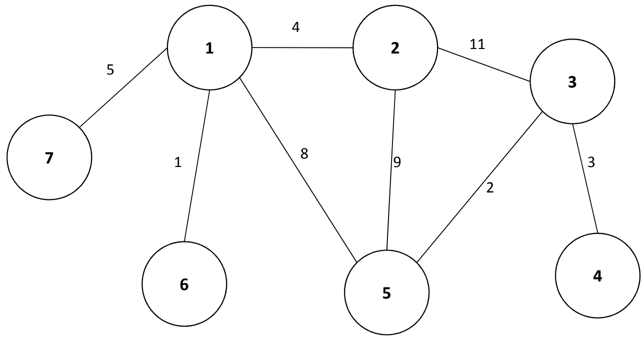
Solution: The length of the shortest spanning tree is 1+3+4+5+6+8=27



Edge	Length	Choice
(4,5)	1	1
(2,3)	3	2
(3,5)	4	3
(1,5)	5	4
(5,7)	6	5
(1,4)	7	Reject
(6,7)	8	6
(2,6)	9	-
(3,6)	12	-

Problem 2 - Prim's Algorithm

a) Calculate the length of the shortest spanning tree using Prim's Algorithm. Draw the final graph and show your results in the table. Complete *U* and *S*.

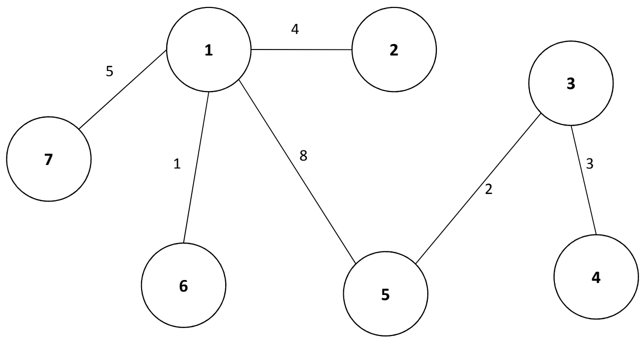


k	i(k)	λ_k
1	1	-
2		
3		
4		
5		
6	1	1
7		

U={1,6}
S={(1,6)}

Notation:
k - node
i(k) - preceding node
λ_k - distance between k and i(k)
U={} - ordered set of nodes
S={} - ordered set of edges

Solution:



k	i(k)	λ_k
1	1	-
2	1	4
3	5	2
4	3	3
5	1	8
6	1	1
7	1	5

Length of spanning tree is 23

U={1,6,2,7,5,3,4}
S={(1,6),(1,2),(1,7),(1,5),(5,3),(3,4)}

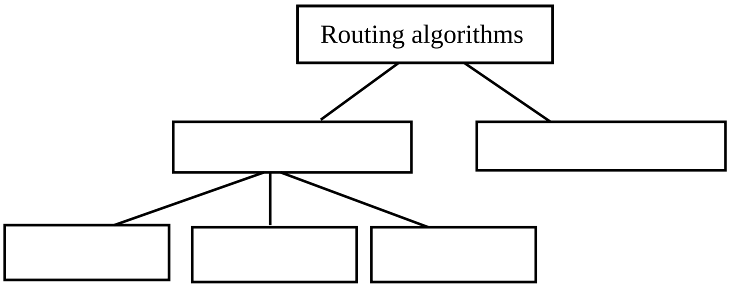
b) Compare Prim's Algorithm with Kruskal's Algorithm: What is the main difference between these two algorithms? How are cycles avoided in Prim's Algorithm?

Solution: The main difference is the way cycles are avoided

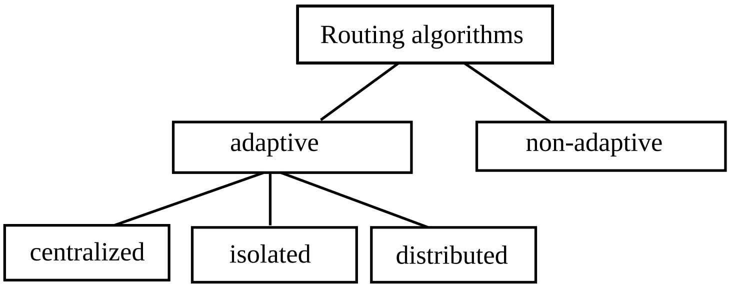
- Kruskal has to search actively for cycles before adding any edge to the result set
- Prim is a greedy algorithm
  - uses a set of visited vertices
  - in every step a (local) edge from a visited to a non-visited vertex is selected and added to the result set
  - cycles are automatically avoided
  - gives a tree in every intermediate step

Problem 3 - Routing in General

a) We know a lot of routing algorithms. Please give a classification of these algorithms, using the following scheme:



Solution:



b) Describe shortly some advantages and disadvantages of the different classes of routing algorithms.

Solution:

- non-adaptive routing:
  - \+ simple (pre-defined routing tables)
  - \+ good with consistent topology and traffic
  - – can not react on network changes / network state
  - – poor performance in non-consistent networks
- adaptive routing:
  - \+ can adapt to actual network state
  - – routing tables must continously be re-calculated
- centralized:
  - \+ RCC has complete information → perfect decisions
  - \+ intermediate systems (IS) free of routing calculations
  - – low robustness
  - – IS receive routing tables at different times
  - – traffic concentration around RCC
- isolated:
  - \+ no routing overhead
  - – only limited adaption to network state possible

- distributed:
  + better adaption to network state
  − routing overhead

c) Name three routing algorithms and the class they belong to!

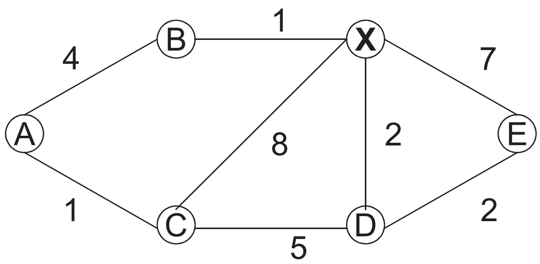
Solution:

Non-adaptive routing Shortest Path; Flow Based; Flooding

Adaptive isolated routing Backward Learning

Adaptive distributed routing Distance-Vector; Link State

d) Determine the routing table for node X according to Dijkstra's Shortest- Path-Algorithm for the following network topology:



Solution:

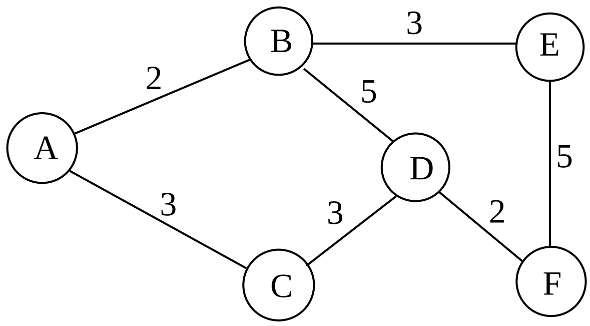
Dest.	Distance	Outgoing Link
A	5	B
B	1	B
C	6	B
D	2	D
E	4	D

Problem 4 - Backward Learning

a) Which class of routing algorithms does the Backward Learning algorithm belong to?

Solution: adaptive, isolated

b) Look at the following network (the numbers indicate a metric for the distance of 2 nodes):



In this network, the Backward Learning routing algorithm is used. You can find the routing tables of the nodes in the following table (next page). The intital state of the routing tables was specified by the network administrator. The entries in the routing tables have the following form:

Station	Target	Egress/Costs

A question mark means, that the distance is not yet known.
Now six packets are transmitted over the network (in this order):

Sender		Receiver
F	→	A
D	→	A
B	→	F
C	→	B
E	→	D
A	→	F

Determine after each transmission, how the routing tables of all the stations look like (fill out the table on the next page). You need not to copy the entries, when nothing changes.

			after transmission of packet…					
Node	Target	Initial state	F → A	D → A	B → F	C → B	E → D	A → F
A	A	-						
	B	B/2						
	C	C/3						
	D	B/?		B/12				
	E	B/?						
	F	B/?	B/10					
B	A	A/2						
	B	-						
	C	D/?				D/8		
	D	D/5						
	E	E/3						
	F	D/?	E/8					
C	A	A/3						
	B	D/?						
	C	-						
	D	D/3						
	E	D/?						
	F	D/?						
D	A	F/?						
	B	B/5						
	C	C/3						
	D	-						
	E	F/?					F/7	
	F	F/2						
E	A	B/?						B/5
	B	B/3						
	C	B/?						
	D	B/?		F/7				
	E	-						
	F	F/5						
F	A	E/?						E/10
	B	E/?			E/8			
	C	E/?						
	D	D/2						
	E	E/5						
	F	-						

c) Now, the entries of the routing table of station B are not yet optimal. For which targets the optimal path was not found? Can you give some packets, whose transmission improve B's routing table?

Solution: Not optimal are the entries for C and F.
No, C and F both do not send their packets over the shortest path to B, so B can not learn these paths.

d) Assume the network is in the initial state again (like in b). Find a sequence of packets, that will cause the best possible entries in B's routing table (considering the initial state). Fill out the following table with the needed packets and B's routing table after each transmission (you need not write down the routing tables of the other stations):

Solution:

			after transmission of packet...					
Node	Target	Initial state	B→F	F→B	C→B			
B	A	A/2						
	B	-						
	C	D/?			D/8			
	D	D/5						
	E	E/3						
	F	D/?		D/7				

Solution: With the first packet (B → F), F learns the shortest way to B, than (second packet) B can learn the shortest way to F. The third packet is like in b). B can not learn the way to C over A, because C sends its packets to B over D (and not over A).

e) How can the Backward Learning algorithm be modified, so that the quality of the routing tables improves?

Solution: The routing tables become better, when a packet uses a shorter path then the packets before. To achieve this in cases like in b) and c), e.g. the following could be done:

- a sender could sometimes randomly choose an egress
- a sender could, from time to time, flood its packets over all outgoing lines

If the network topology changes, periodic deletion and relearning of the routing tables will help.