Ex. 9-1: Routing in General / Flooding

a) Classify the following routing methods due to the schematics of the lecture.

   i) Backward Learning

   ii) Flooding

   iii) Distance-Vector Routing

b) You are programming flooding in a router. Each router i knows about its ID i and links $L = \{l_1, \ldots, l_N\}$. Assume that your development environment offers the hook procedure
`packetReceived(NLPacket nlp, Link l)`,
i.e., this method is called on the reception of a packet p on link l. Also, you can use the procedures
`GiveToLower(NLPacket nlp, Link l)` and `GiveToUpper(ULPacket p)`.

   i) Define the minimum network layer packet data structure required for flooding. *HINT: You can use C/Java-style syntax to describe the data structure, e.g., the NLPacket payload would be NLPacket.payload.*

   ii) Implement the body of `packetReceived(..)`

c) What is the Big-Oh complexity of the number of LL packets of your algorithm, when the network has V nodes and E edges, the average path is L_P hops and there are U sending requests of the upper layers?

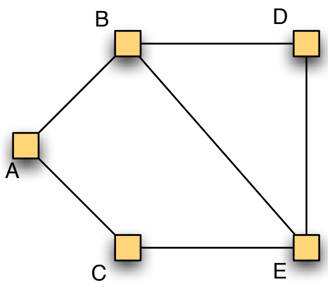
d) What additional information would be required to be able to additionally use backward learning and how could backward learning be integrated into your program?

Ex. 9-2: Routing Information Protocol (RIP)

a) Which routing principle is used by RIP?

b) Define the (minimalistic) packet format of a RIP-like protocol and describe the protocol using the data structure you have defined.

c) The following figure depicts an examplary network with 5 nodes with the Routing Table entries given below. Explain (step by step) the evolution of these routing tables after the failure of link $\overline{AB}$.



Node A		
Destination	**Link**	**Costs**
A	—	0
B	$\overline{AB}$	1
C	$\overline{AC}$	1
D	$\overline{AB}$	2
E	$\overline{AB}$	2

Node B		
Destination	**Link**	**Costs**
A	$\overline{AB}$	1
B	—	0
C	$\overline{AB}$	2
D	$\overline{BD}$	1
E	$\overline{BE}$	1

Node C		
Destination	**Link**	**Costs**
A	$\overline{AC}$	1
B	$\overline{AC}$	2
C	—	0
D	$\overline{CE}$	2
E	$\overline{CE}$	1

Node D		
Destination	**Link**	**Costs**
A	$\overline{BD}$	2
B	$\overline{BD}$	1
C	$\overline{DE}$	2
D	—	0
E	$\overline{DE}$	1

Node E		
Destination	**Link**	**Costs**
A	$\overline{BE}$	2
B	$\overline{BE}$	1
C	$\overline{CE}$	1
D	$\overline{DE}$	1
E	—	0

d) What is the Big-Oh complexity of the number of LL packets of the algorithm, when the network has V nodes and E edges, the average path is L_P hops and there are U sending requests of the upper layers?