Assignment 1 – Kristoffer Dalby

Problem 1

a.
In an unstructured network, to find a node you will in a worst case scenario need to seek through all the nodes. In a best case scenario, the first node that is checked is the node that you are looking for. The nodes does not need to have information about each other.

In a structured network, lookup is based on keys (GUIDs) which in turn makes the amount of jumps to a node significantly smaller on average than the unstructured network. Many of the structured network can find the next node in log(n) hops. The downside is that each node must save information about the other nodes.

b.
Routing table for 599:

0	1	2 219	3 329	4	5	6	7 709	8	9	A	B B23	C C61	D	E EC9	F F19
50	51	52	53	54 540	55	56	57	58	59	5A	5B	5C	5D	5E 5EC	5F
590	591	592	593	594	595	596	597	598	599	59A	59B	59C	59D	59E	59F

Routing table for C61:

0	1	2 219	3 329	4	5 540	6	7 709	8	9	A	B B23	C	D	E EC9	F F19
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
C60	C61	C62	C63	C64	C65	C66	C67	C68	C69	C6A	C6B	C6C	C6D	C6E	C6F

c.

The four possible scenarios is based on the four possible values under F in C61s routing table. The possibilities are F19, F84, FF2 and FC7.

The algorithm starts by checking the first digit from the goal node (F) in C61s routing table, this yields a node, if this node is the goal node or the goal node is in the leaf nodes, return and terminate.
If none of the checked nodes match, go back to the routing table, but use an additional digit to narrow the search for the goal node (FC). In this case, this will yield the goal node.

In the first alternative, the node under F in C61s routing table if F19, the algorithm checks the leaf nodes and finds EE0 and F84.
C61 -> F19 -> FC7

In the second alternative, the node under F in C61s routing table is F84, the algorithm checks the leaf nodes of F84 and finds FC7.
C61 -> F84 -> FC7

In the third alternative, the node under F in C61s routing table is FF2, the algorithm checks the leaf nodes of FF2 and finds FC7.
C61 -> FF2 -> FC7

In the fourth alternative, the node under F in C61s routing tables is FC7. Algorithm terminates.
C61 -> FC7

Problem 2

a.

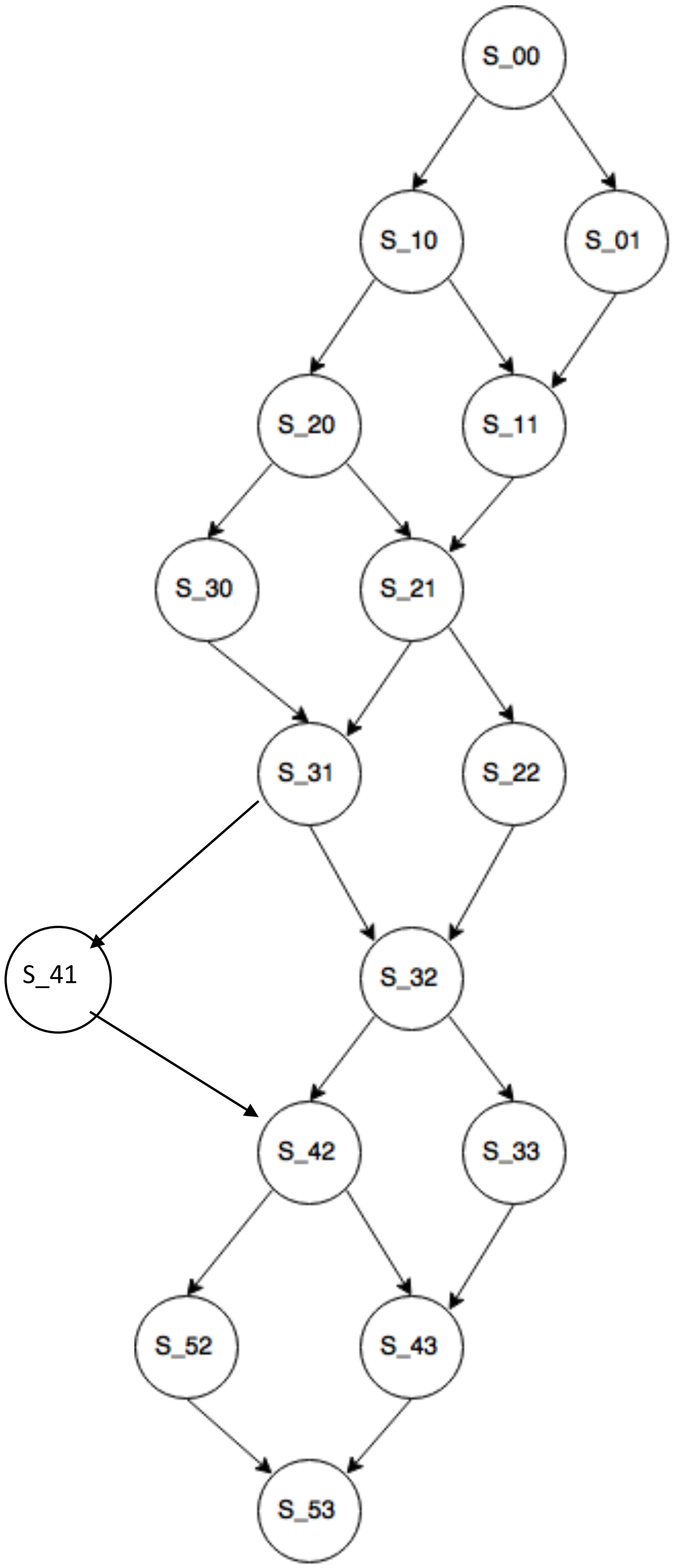
If a set of Lamport values/timestamps are such that L (e) < L (f) we can deduce that e did not "happened before" f, this is because the e and the f execution not necessarily has sent a message to each other. And if they have not done this, the timestamp has not been exchanged and incremented by f.

Vector clocks are more suited for management of dependencies and casualties. This is because the clock information is stored for each process in a vector with N elements, where N is the number off processes. A great example of how vector clock can be used to check dependencies is the solution in Problem 3, where one can imagine that the X and Y can be the digits in the vector clock.

b.

b = (4, 0, 0)
k = (4, 2, 0)
m = (4, 3, 0)
c = (4, 3, 2)
u = (4, 4, 0)
n = (5, 4, 0)

Problem 3


S_00
S_10
S_01
S_20
S_11
S_30
S_21
S_31
S_22
S_41
S_32
S_42
S_33
S_52
S_43
S_53