**INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR**

**Department: Computer Science and Engineering**

**Spring Semester: 2013**

Date Time Full Marks: 50

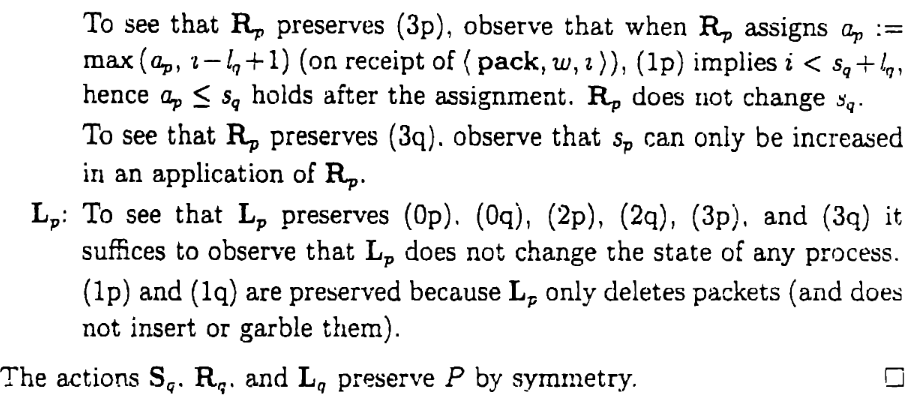
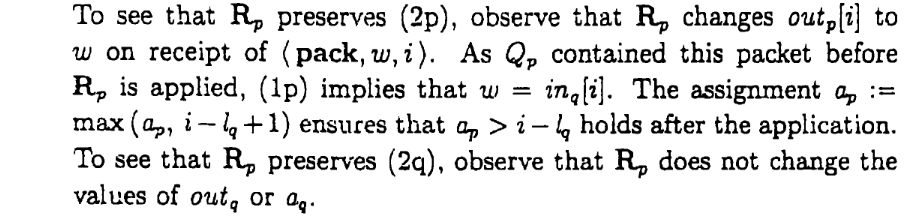
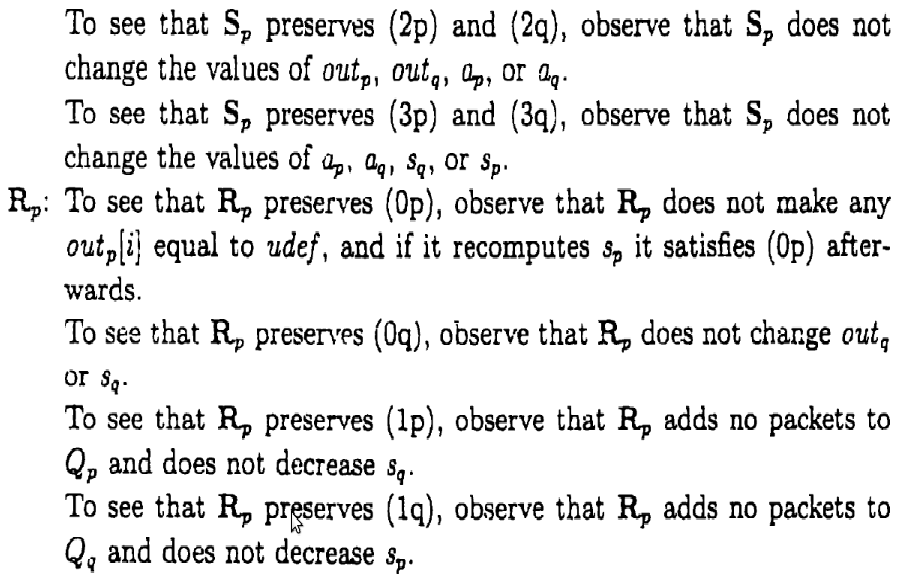
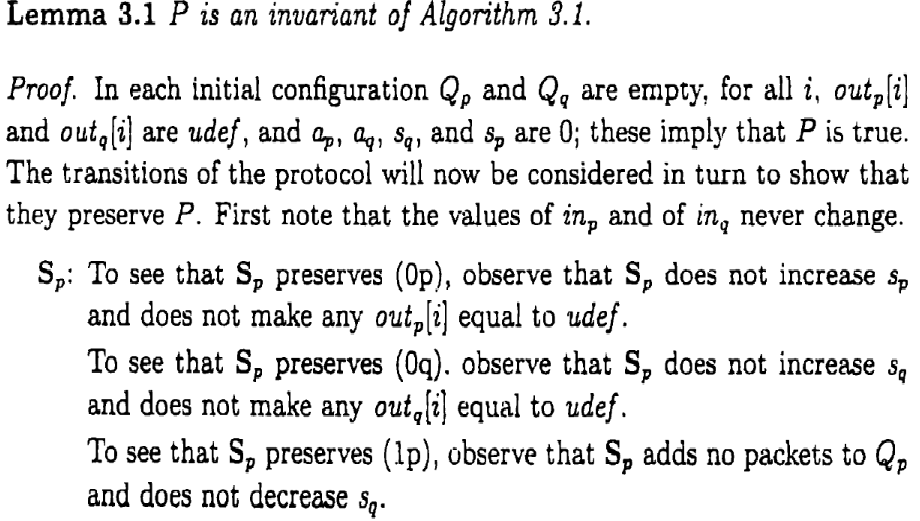
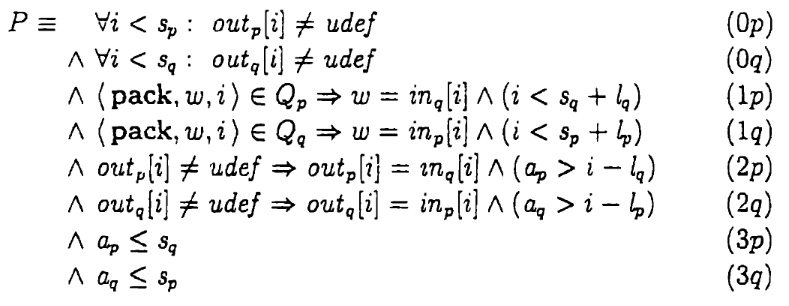
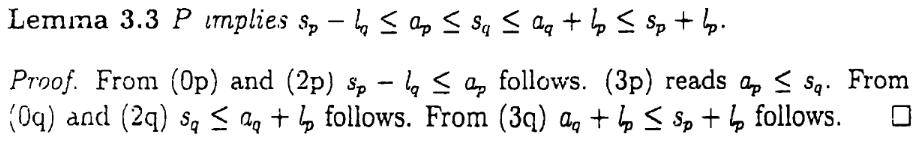
Sub. No: CS60002 Sub. Name: Distributed Systems

Answer as much as you can

1. Two processes P and Q send and receive packets using sliding window protocol. Prove the following

Sp – lq ≤ ap ≤ sq ≤ aq + lp ≤ sp + lp

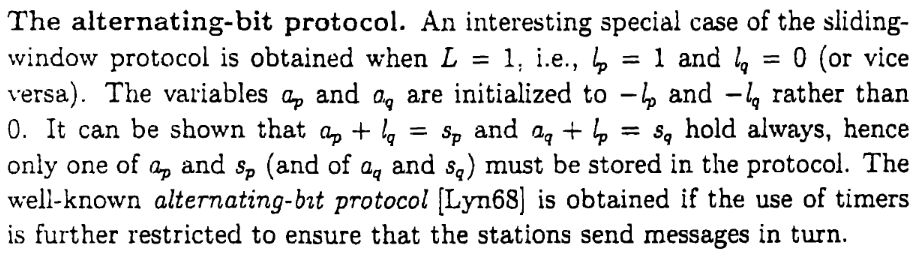
7

Ans: 

1. Modify the parameters of balanced sliding window protocol in such a manner that it acts as a stop and wait protocol. Show example with packet exchanges among two processes P and Q.

4

Ans:

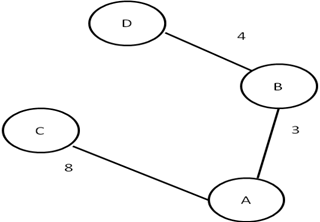
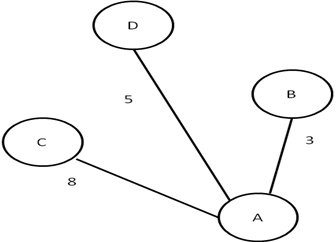


1. Consider that Toueg’s algorithm is being used to build routing table of the following network. Let’s assume that pivots are selected in the following order A, B, C, D, E, F and G.

Draw rooted trees (to pass routing table) for A, B and C when they are selected as pivot.

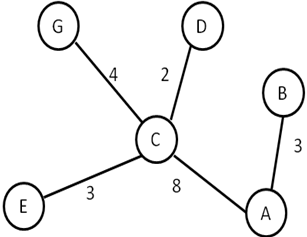
3x2=6

Ans:



Rooted tree for B

Rooted tree for A

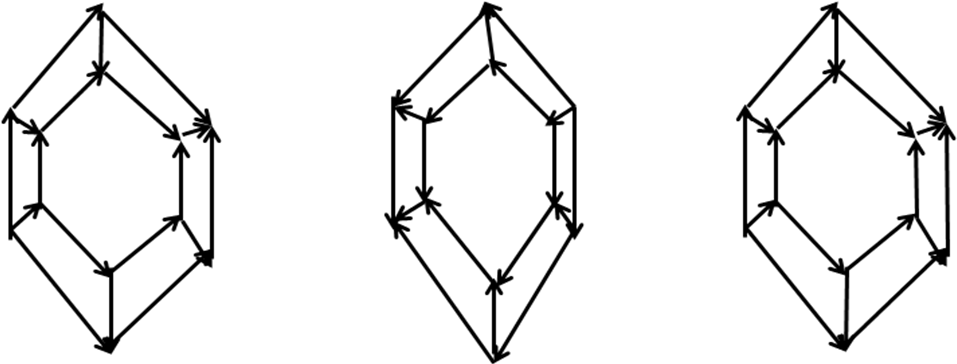


Rooted tree for C

* 1. Find acyclic covers for the following network
  2. If more buffers are used (than acyclic orientation cover), what extra advantage will we get?

4+1=5

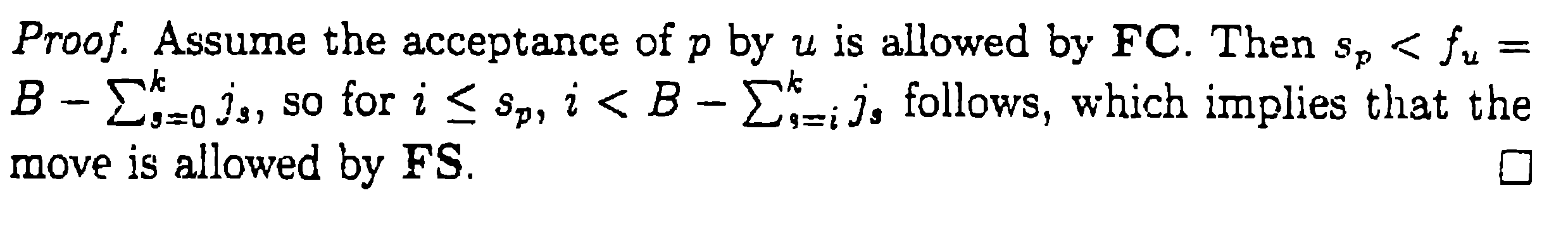
Ans: a)



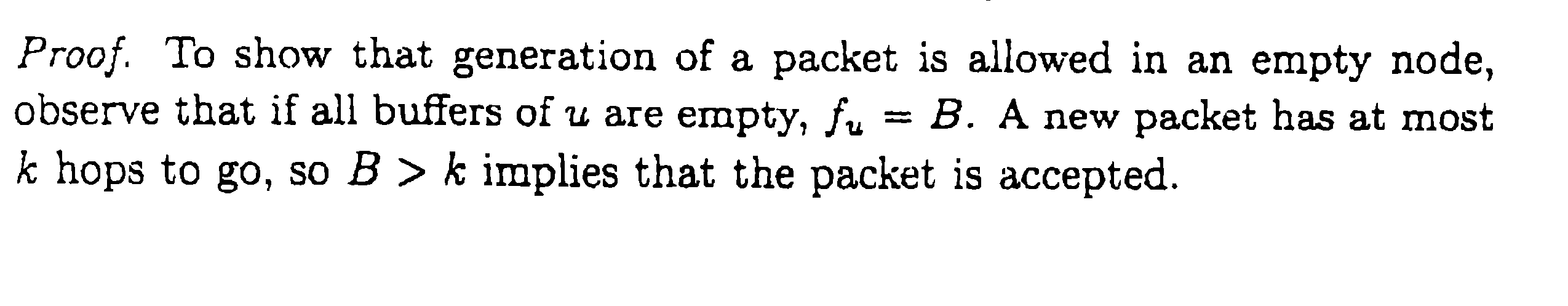
b)

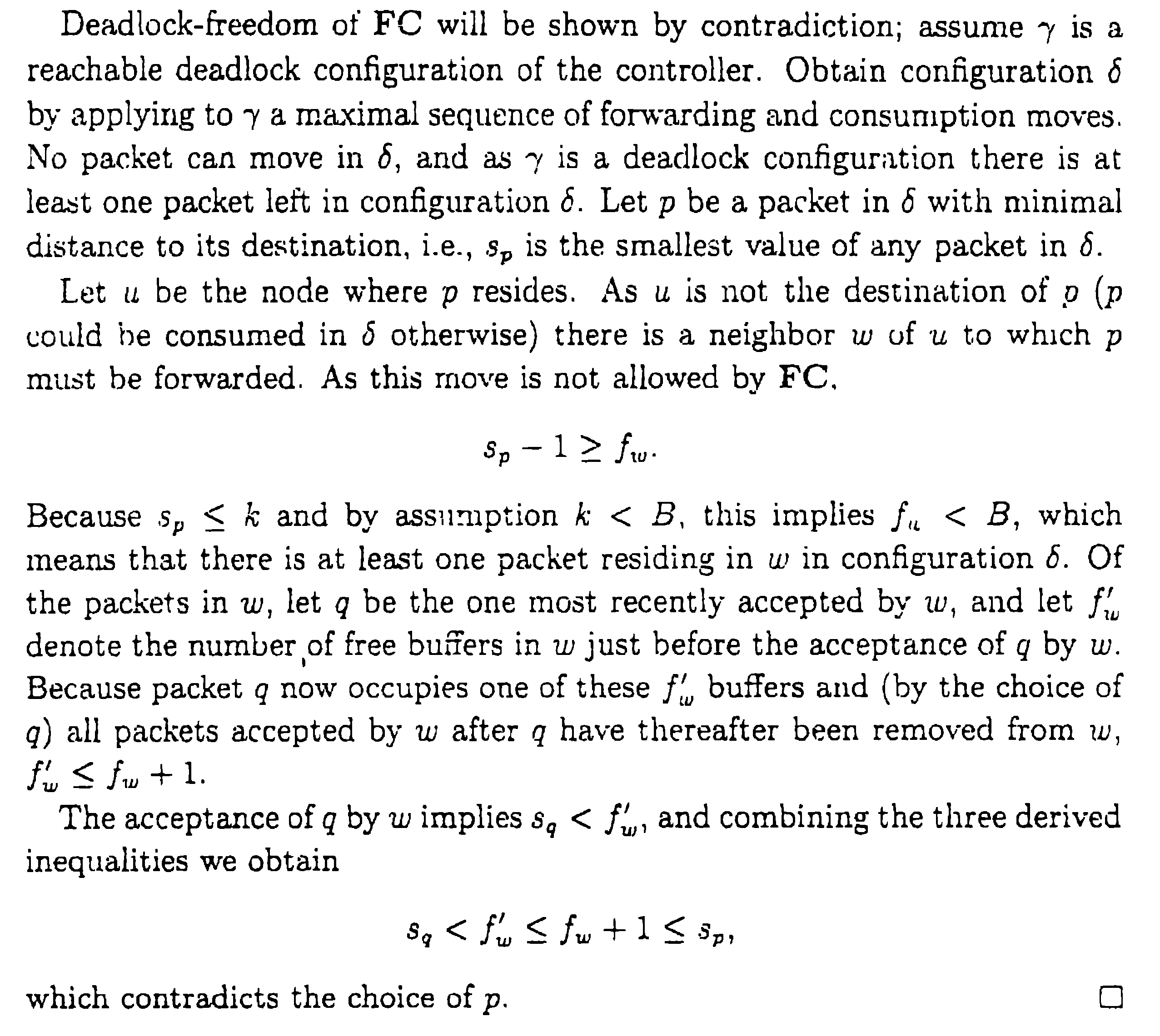
It will increase the bandwidth. Assume 3 buffers are needed according to acyclic orientation cover. Instead of 3 buffers, if we use 6 buffers we may have one redundant plane for parallel routing.

* 1. Prove that, each move allowed by Forward Count Controller, is also allowed by Forward State Controller.

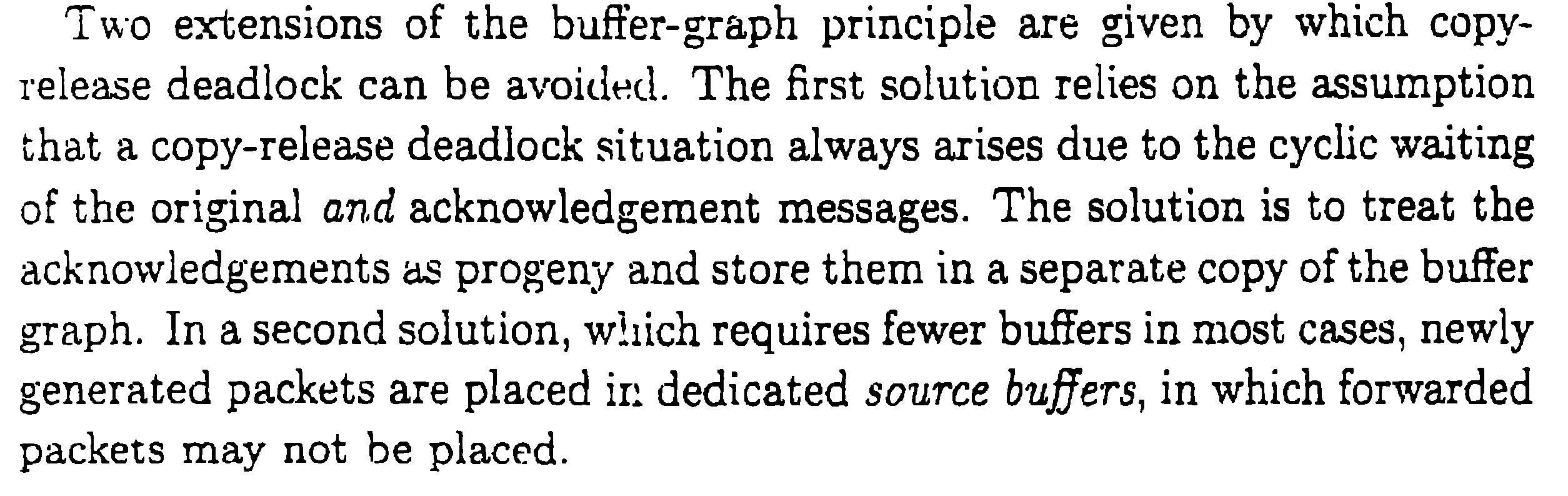


* 1. A controller that is a “dual” to Forward count Controller is called Backward Count Controller, i.e., it accepts a packet p iff tp > k – fu . Prove that, a Backward Count Controller is a deadlock-free controller.

FC is dual of BC. BC is deadlock free iff FC is deadlock free subjected to this condition. 



* 1. Copy-release deadlock may arise when the source node holds the copy of the packet until an end-to-end acknowledgement from the destination is received. But, this violates the assumption of deadlock free packet switching, that the buffer becomes empty when the packet is forwarded. Please explain your extension of buffer graph principle to combat against this type of deadlock.



4+6+2=12

* 1. Show that getting any message from all other processes is necessary but not sufficient condition to enter critical section in Lamport’s mutual exclusion algorithm.
  2. Show that being at top of the own request queue is necessary but not sufficient condition to enter critical section for Lamport’s mutual exclusion scheme.
  3. Comment on feasibility of the below message sequence in Lamport’s mutual exclusion algorithm.

2+2+1=5

Answer:

1. In Lamport’s mutual exclusion algorithm, whenever a process receives a request for entering critical section from another process, it sends reply back. So, it is feasible that at one time instant more than one process receives reply from all other process. However, two processes can not enter critical section at same time.
2. Let assume that there are 3 processes say x, y and z are competing for critical section. Also assume that process x receives reply of critical section message from only y and currently its own request is on top of the queue. However, it may happen that 3rd process that is z from which x still have not received the reply would be at the top. In other words, z may receive all replies from both x and y and its own request is at the top of queue. So only being at top of queue is not enough to enter critical section.
3. Scenario is infeasible. P2 will never send message corresponding to timestamp 13 as already there is a pending critical section request.
4. Give an example to show that vector clocks are more powerful than Lamport’s logical clock.

3

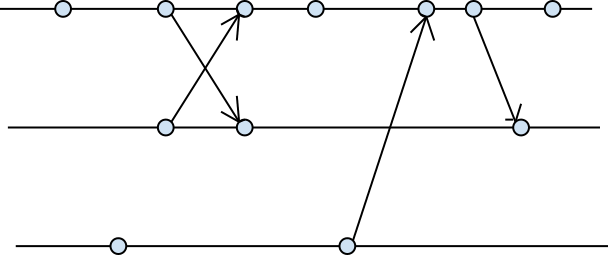
Lamport timestamps can tell us that

* x \nrightarrow y (if C(x) \geq C(y))

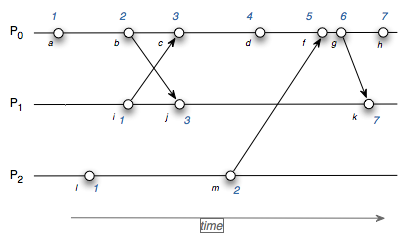
Vector clocks can tell us that

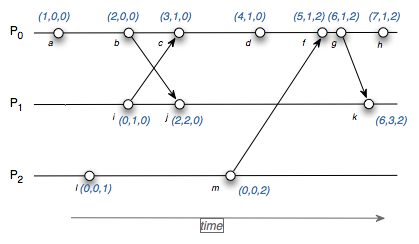
* x \nrightarrow y (if VC(x) > VC(y))
* x \rightarrow y (if VC(x) < VC(y))
* neither x \rightarrow y or y \rightarrow x otherwise

1. Find the Lamport’s logical clock and vector clock time stamp of all the events shown as circles below



3



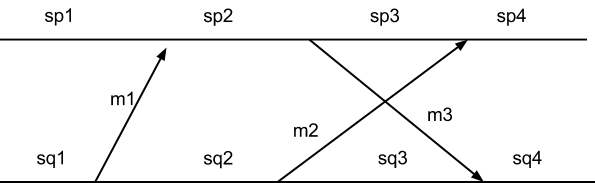


1. In Huang’s algorithm prove that W=1 signifies termination.

2

Initially the controlling agent starts out with a weight W=1. Each time a request is sent to another process, the requesting process loses some weight. It is returned to it when the request has been completed. The total sum of all the weights of all processes is constant i.e. 1 and it is just distributed among many processes until termination occurs (i.e. all requests were fulfilled) that is when all the weight comes back to the controlling agent and the total sum once again becomes 1.

1. Give 3 examples of consistent, in-consistent and transit states from process diagram given below (Note: signifies that local state was recorded at that instant).



Inconsistent = sp2,sq1

sq4,sp2

sp4,sq2

Consistent = sp1,sq1

sp2, sq2

sp4,sq4

Transit = sq2,sp1

sq3, sp3

sp3,sq2