HW 3

Name: Jiacheng Zhao

Resources. (All people, books, articles, web pages, etc. that have been consulted when producing your answers to this homework)

Textbook and instructor's slides

http://www.cs.cornell.edu/courses/cs614/1999sp/notes99/byzantine.html

On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment. Furthermore, I have disclosed all resources (people, books, web sites, etc.) that have been used to prepare this homework. This work is my own and is written in my own words.

Signature: JIACHENG ZHAO

Problem 1. Exercise 15.12

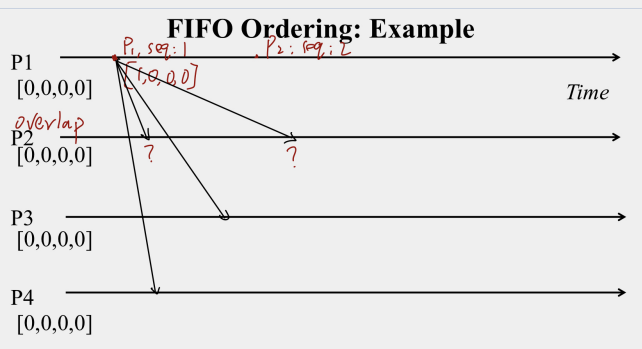
Solution.

Reversing the order will cause that a process can deliver a message and then crush before sending it to the members of other groups. This is contradicts the uniform agreement property.

The reliable multicast algorithm based on IP multicast does not satisfy uniform agreement. A recipient delivers the message as soon as it receives it. If the sender crushes during the transmission, then this message will not reach the other group members. In this case the uniform agreement will not be met.

Problem 2. Exercise 15.16

Solution.



Considering this scenario presented above. Suppose P2 is the intersection of two groups (P2, P3 for group g_1, P2, P4 for group g_2) . At some time P1 issues a multicast message m1 to one group, then m2 to the other with the same sequence number 1. Since these are two different groups, the sequence number could be same. However, there is no algorithm for P2 to decide which comes first, since both messages are coming with the sequence number 1. This case shows that FIFO order is not suitable for overlapping groups.

In order to modify the FIFO-order algorithm suitable for the overlapping groups, we can add the sequence number with the group name, e.g., (g_1, Sequence Number 1), (g_2, Sequence Number 1). This method will help overlapping process like P2 to decide the order of several coming messages. Take the example above, using the improved algorithm, P2 will determine that m1 comes first and m2 comes later, since they have the different sequence number.

Problem 3. Exercise 15.17

Solution.

Let's think Scenario 1. Suppose process P1 issues a multicast (g,m) to a group g and then issues (g,m') afterwards. The messages are FIFO ordered. In

this case,the sequencer should receive $m^{'}$ after receiving m since this is order they were sent, and every other process should receive the messages in the same order.

Let's think Scenario 2. Suppose process p1 issues a multicast (g,m) and then process p2 receives it. After receiving the message (g,m), P2 issues (g,$m^{'}$) afterwards. These messages are total-ordered. In this case, the order of receipts should remain the same at all processes. So in every other correct process, message $m^{'}$ should be received after message m being received.

If these two scenarios happens together, we show that causal ordering is achieved for the happened-before relation, since these two cases are also casually related.

I think it's true that any multicast that is both FIFO-ordered and totally ordered is thereby causally ordered, since the sequencer is being utilized and sequence number of any message sent is greater than that of any received by the sending process.

Problem 4. Exercise 15.22

Solution.

Integrity modification:

We can use the definition of majority to modify the integrity, which means that the value that occurs most often among its arguments could be the one which all the correct processes choose in the decision state. If there is no value occurring most among the arguments, we can use a special value V as the final decision and all the correct process will choose V in their final decision.

As for the algorithm, just choose the value that occurs most as the majority and let the all the correct processes to choose that value in their final decision. If the majority does not exist, let the pre-installed special value V as the majority and let all the correct processes to choose that value.

Problem 5. Exercise 15.23

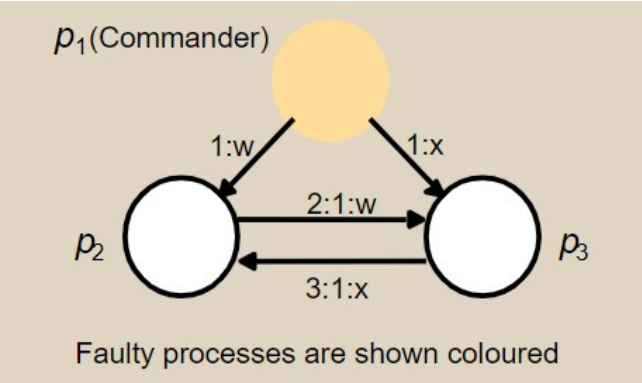
Solution.

Before using the signed messages, we first make the following assumptions:

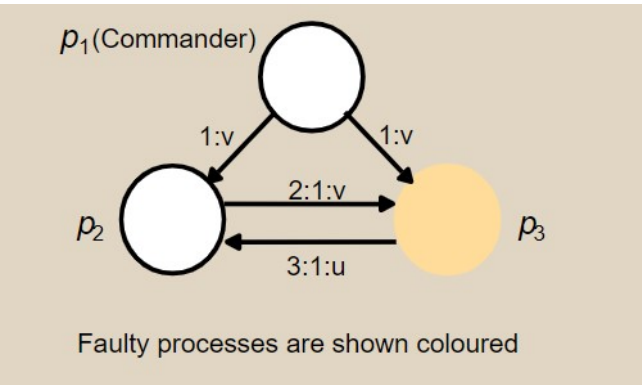
1. Any loyal generals signature can not be tampered, which means any changed can be detected. The signed message can be dropped, but can not be altered.

2. Anyone can verify the authentication of a signature. No one can fool a loyal general.

Let's consider the first scenario, which is that commander is treacherous.



Two lieutenants will find the signed messages they have received form the commander are different after exchanging them. Since both signed messages have authenticated signature from the commander and there is no way to tamper it, two lieutenants will recognize that there must be the commander who is faulty right now. Now the rest thing for two lieutenants to do is that they should come up with an agreement without obeying their commander, since they have found out that commander is faulty.



If one of the lieutenants is treacherous, its tampered message will be detected if it's received by a loyal lieutenant. Since the loyal lieutenant has found out that the other one is faulty, he should choose to obey the commander in order to achieve a consensus.

The proof above shows that the member who is treacherous can be detected or found out when signature is being put into effective. In that case the BGP can be solved when using signed messages.

10