**Latency analysis for perioding communication requests in a ring network**

******

Stagiaire: Falade Joannes Structure d’accueil : Laboratoire DAVID Université de Versailles Saint Quentin en Yvelines (UVSQY)

Encadré par : Mr Dominique BARTH

Table des matières

[***Introduction*** 2](#_Toc457566820)

[***I.*** ***Description*** 3](#_Toc457566821)

[***II.*** ***Implementation*** 5](#_Toc457566822)

[***III.*** ***Simulation*** 7](#_Toc457566823)

[***Conclusion*** 13](#_Toc457566824)

# 

# ***Introduction***

In optical network, the goal is to ensure reliable and efficient communication between differents nodes of the system. However, It’s very important to analyse the duration of exchange. This is called latent period in communication. By definition, latent period is considered as the difference between the time when the transmitter have to give out his message and when the receiver really read it.

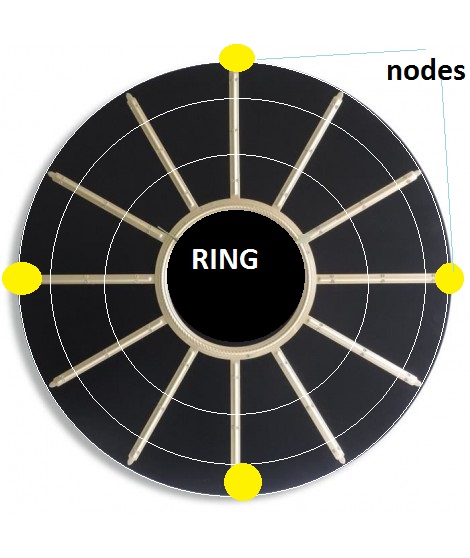
In this report, the objective is to acquaint the operating principle of a ring network structure and analyze the different cases of latency in communication between nodes. We have to implement the model stucture in C langage and make more experience about it to find out latency evolution in the time and propose suggestion to reduce the latency per message.

Firstly, this report contains an shortly description of problem and model we try to analyze; then we explain our implementation and finally discuss about our simulation test.

1. ***Description***

There are several distributed algorithms which are used to ensure coordination in the system. Our study is based on ring system where each node is link up with one predecessor and one sucessor in order to make a circle.

The ring has a clock wich move on and permitted dragging of message containeur arround all nodes. And for each node, it can write on one wavelength forwards and also read on backwards. For writing, the node can send his message if the containeur is empty and read if the message on the reading position is for its. To evaluate the latent period, it is important to notice that the message is spontaneously create when the node is capable to send it but is put on the list message of node. So, the latency is calculate with time of message creating and time when the receivor get it. The model that we are going to study guaranteed message transmission but it is the latent period which is acttracted our analyze and how to change it better.

This picture explain exactly what we are trying to do because it represents what about we are talking in this document. Notice that the node of the ring are yellow fixed points. The space which separate two stick is what we are calling slot and the wavelength is the number of small box in the slot. In addition, for each node the wavelength is the maximum number of message that the node can send. At the click of the clock, all the ring turn around the node which are fixed as we told you upper ; this move represents one slot shifting around the node. After this move it is important to mention that each can check in the slot box before it’s to find if this contain message addressed to it or not and after can put in the slot box after him the message that it have to transmit if the box is empty , else the node have to waiting for the next click of clock.

According to our picture, it’s important to hold back we have four nodes ; the wave length is three and there are three slot between two nodes. After one clock’s click, the four nodes are fixed and check in the slot box or send message if possible.

By analysing the system running, we note that all the message that one node have to send may not be sent automatically, because the wave length can be small than the number of message this node has to send. So, each node has one list where he can put all the message in waiting to send. Then start the problem of latency, which is difference between the real period message reception and the period that this message has to be receive if it was sent on time.

1. ***Implementation***

For implementation, we do it by using C programming language.

The system that we use is a simple ring, which has a circle form. In implementation, we have to model the ring and nodes structure to know at each step the evolution of transmission system. As a distributed algorithm, we set up our program to permit each node to update his state about sending or message receiving.

About node, it possess number which is unique in the ring, one line which conserve all the message that need to be send by this node and waiting the container ; the request table that can be seen as a table where position identify specific request that the node can send for the rest of node in the ring.

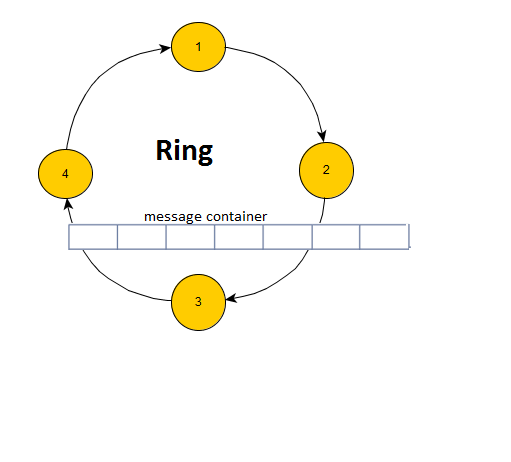
In regards to ring, it a list of different node of system which are chain as circle. The information’s that ring manager know is the number of node that form ring, the message container which turn around the nodes to get or send message for each of them. Notice that the ring also define the the capacity maximal of message sending or receiving per node ; the ring also contain the number of slot which separe two consecutive node. We also describe the message form in the code like one stucture which contain source and destination number and the time that the message is effectively create.  
To enable the protocol simulation we create several method and ring management function. Among, the latter must:

**preparerMessage( ):** this permitted to each of node to prepare the message which he has to send and put it in his waiting list ;

**envoyerMessage( ):** that’s assure the message sending in the containeur if the place is empty.

**retirermessage( ):** the current node in the ring execute this method to check if the message in reception position is for him and try to get it ; else the let it in the container for real receivor.

**unclicAnneau( ):** this execute what will be done on one ring turn and each node try to catch and send message.

**simulerEnvoi( ):** permeitted us to simulate the ring turning during a period what we want to analyse the communication.

This picture resume what we really implement. So you can understand that we realise the ring as a file of node. About the message container that is a table of all box number arround the ring. This length of table is :

wave\_length\*number\_slot\_between\_two\_nodes\*number\_of\_node.

Then, for each clock’s clic we have to make one course of the ring to send or get message from node. Particular emphasis is placed on the position of insertion for each moves. And for each course we calculate the average of node message reception.

In the code source, these parameter are important to know to change your simulation running to find the modification.

**nombreSommet:** represent the number of nodes you want to create in your ring simulation

**longueurOnde :**represent the wave length

**nbSlotEntreDeuxSommet:** this, permetted you to fixe the number of slot betweent two nodes

**DUREE:** allow you to fix the duration of your simulation. The value represent the number of your ring’s turn as a clock clic eventually

**NMAXPAQUET:** this define the maximum number of message for each request you have to create in the ring

**TIMEMAXDEBUT:** is the max of start date for request on the ring

**NBMAXSLOT:** is the maximum slot frequency that you want to product your request.

This parameter are used to change our simulation value to see more different observation. By exemple, if you want to increase the node charge int the system, you have to enlarge the value of **NMAXPAQUET** and reduce the frequence value **NBMAXSLOT**.

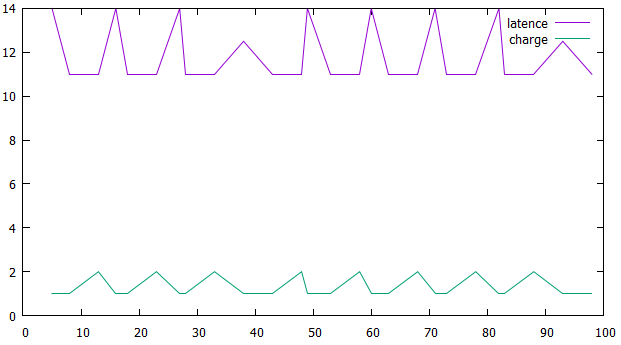
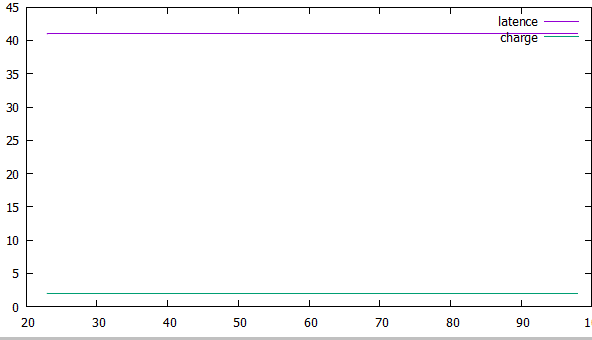
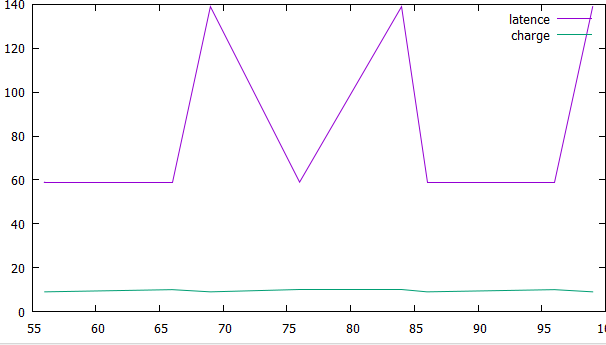
The goal of this implementation is calculate for each message received, the difference between the time when it should be received in normal case and the time where it was really received ; If there is no more time, we considere that no latent period existe else, the difference is be considered as latent periode. We have to analyze the system evolution by making more simulation and point out the curve of this. For more, we will try to add ressource or limitted request form to avoid or make latent period better.

1. ***Simulation***

In this section, we have to execute many times our program implementation to observe what has happened in the time by changing some parameter.

1. **Ring with Small charge**

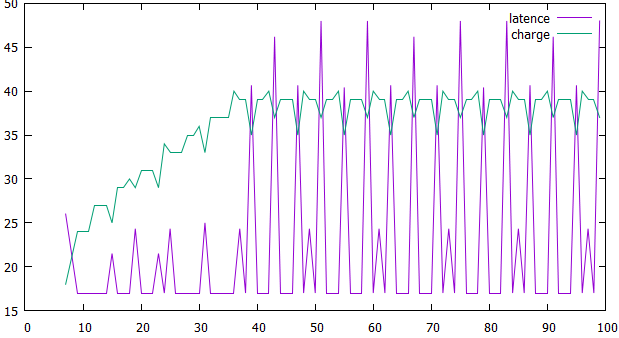
To avoid having too full system, it is important to have few request, not a lot message by request and having a high value of the number of slot between two requests. This condition reduce the frequency of message coming in the ring and ensure the ring having a small charge.

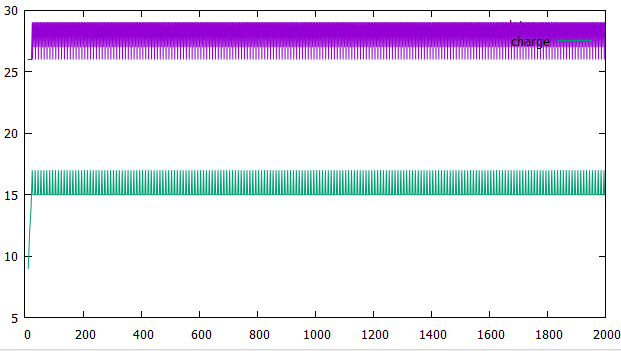
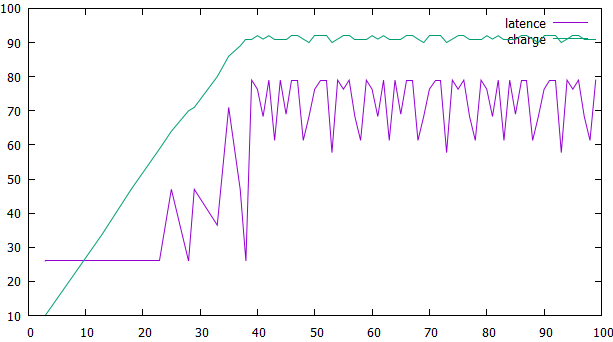


On these pictures, we can observe the evolution of stable system. By simulating, we notice that the charge system is low during the time. The consequence of this is the fact that there are not lot of latency period and if latency appear, it is by increasing at each step. When, the latency change, we have sometimes a pic but this is controlled. Adding that the sequence of latency in the system is periodic and you can predict the system future; so we can find the solution of priority to improve the latency problem in stable system.

1. **Link between Charge and latency period**

To observe this relationship between charge and latency period, we have to change the parameter of request to put more request in the system, then we will reduce the slot of two sending request, and we have to simulate the system for just a short time.

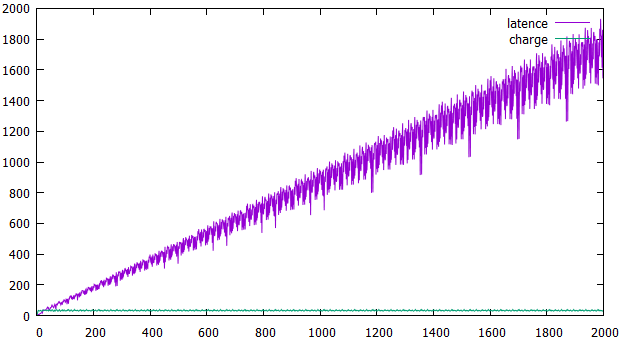
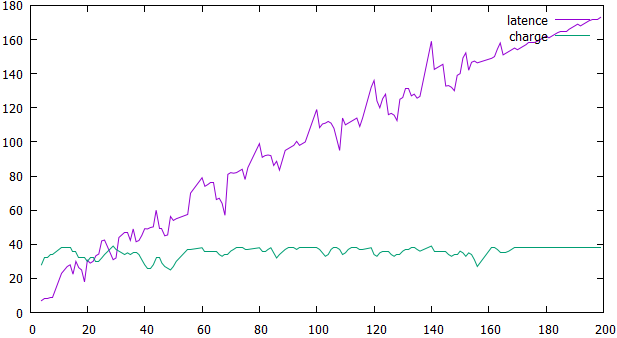
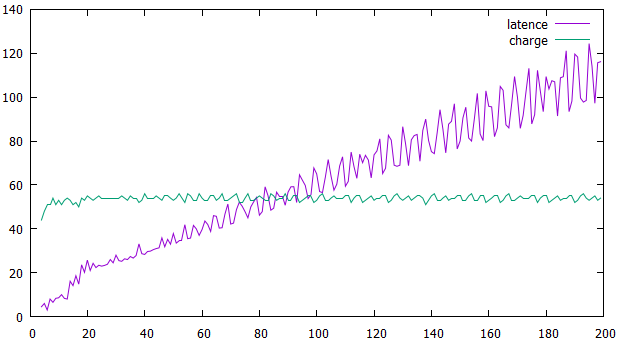




On this picture, the first thing you have to notice is that the system is stable and we can find the period for that. Added , when the charge value change and increase, the value of latency change and get exponential high increased, at the same when the charge get down, the latency period get it too. Therefore, it have the high relationship between the charge and the long-time of latency in the ring. More the ring is charged the longer the waiting time is.

1. **Evolution of highly dense system**

On This state, what we have to do is to put a lot of message in the system, more request and high frequency of their coming. Let us show what can the system be like and what can be done for this.



On this picture, you have to see the evolution of a full system. As you can observe, the curve of charge look like stable because, the system is very full at any time, so the curve does not change. Nevertheless, the curve of latency increases exponentially. That is explain the system contain always message and it is not possible to predict what will be done at this period time.

# ***Conclusion***

In conclusion, what we have to notice about latency period is the fact that it is directly link with the charge of the ring. This is results of more experiences, which show us there are a similarity in latency period curve and the ring charge curve evolution. Then, notice that this relationship has its effects and sense when the system is stable and not full. To control the charge of system, it is important to reduce number of request and the frequency of their coming in the ring. As we explain to you, the latency period cannot be controlled in the system which is too charged.

To find the solution of dense system, we have two options which are:

Firstly trying to kill and destroy the old message to reduce general latency

Secondly, put priority to each request in the system. By example, the request which have the maximum of message has the best priority to get place in the ring.