Erlang's B and C-Formulae: Another Method to Estimate the Number of Routes

James K. Tamgno, Mamadou Alpha Barry, Simplice E. Gnang

ESMT-DTE-LTI, ESMT, Dakar Senegal
ESMT-DTE, Dakar Senegal
james.tamgno@esmt.sn, ralpha.barry@esmt.sn, avecgnang@gmail.com

***Abstract* — This paper recapitulates the work performed to study Poisson distribution model, Erlang distribution model and constraints of dimensioning a network related to these models. This paper also describes the planning tool of a network based on the models considered. We introduce the application behavior, used for packaging Erlang resources. Resources can vary from libraries to process clusters, and can be configured to run on a single processor or be distributed on a set of nodes. These studies have led us to the development of a tool for planning network, which also underlined the limits of the use of Rigault's rule.**

Keywords – Erlang; Rigault; Traffic; Network; Dimensioning.

I. INTRODUCTION

In this new era of information and communication well underway and growing, the telecommunications engineer plays an important role, either as a creator of new technologies that provides players the economic and social or as interpreter of these latter with designers.

To meet these two requirements, among these clusters whose development requires both knowledge in electronics, computing and telecommunications that, the question arises essential point or the overall design and dimensioning of telecommunication networks.

In our approach, we devote the first section to the study of fish distribution model and formulas of Erlang B and Erlang C and in the second, we propose algorithms for determining resources according to formulas given, the third is devoted to the simulation results and comparison with the curves of Erlang is available, and in the fourth section we will study the limits of use of the formula Rigault. Section five concludes the paper and outlines our future work.

II. CONTEXT AND MOTIVATIONS

One of the major problems of the switching is to determine the number of components (Circuits, multi recorders, subscribers, servers) must be installed to cope with the offered traffic following a given quality of service attached.

To solve this problem, we observe the number of calls or number of packets in a data network at every moment they arise, and use formulas Erlang B or Erlang C to determine the number of routes.

We plan to develop an application that will determine the number of routes knowing the offered traffic while providing an opportunity for generation of curves Erlang.

III. RESEARCH APPROACH

The main goal of this research is to implement a program in the Java environment that determines the number of routes available for traffic following a given quality of service (blocking probability or likelihood of waiting).

To achieve these goal we recorded the work methodology includes studying the distribution pattern of fish and formulas of Erlang B and Erlang C, propose an algorithm for determining route following the formulas given, automatically generate curves with Erlang service qualities set, simulation results and comparisons with Erlang curves available, studying the limits of use of the formula RIGAULT.

IV. MODELING OF THE ERLANG FORMULA

A. *Concerning the Erlang formula B*

A1. Standard Model of Erlang B

This is the original model which we will list the assumptions on which it is built:

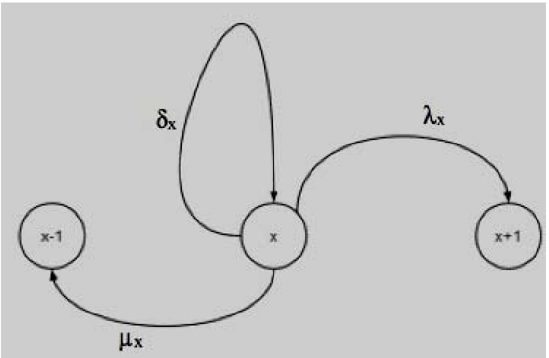
- Calls arrive individually and collectively at random according to a law of fish. This implies that the call appearance rate is independent of the number of calls already committed and that the number of elementary source is infinite. For elementary sources we mean a telephone that can only be in two states, or at rest or during a call (the fault condition is not retained).
- A steady state exists, and, in terms of probability is equivalent to the "steady-state".
- Calls the operation cannot be used due to lack of resources are rejected, that is to say, that for these calls, the length of service is zero. The system does not know what will become of these attempts result in failure and in particular does not distinguish the correlations between successive attempts of the same application.

Attempts are declared indistinguishable will be treated as independent calls.

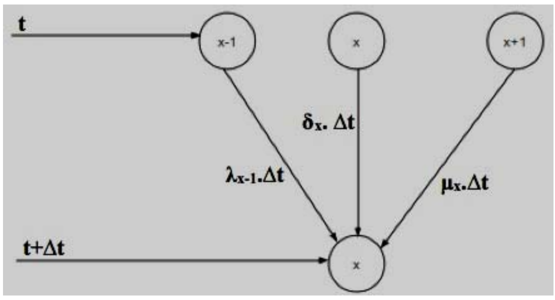
- Service accessibility is total, which is to say that any route may serve any appeal provided that it is available.
- Service time is distributed according to a negative exponential law.

So, giving the state x, and the following two cases:

- transition from state x



- transitions to state x



A2. Calculation of State Probabilities

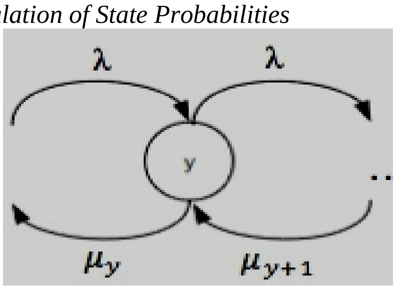


Figure 1. Linear Markov chain model application with loss.

The steady-state probability it is obtained by:

$$P_y = \frac{\prod_0^y \lambda_i}{\prod_{i=1}^y \mu_i} P_0 = \frac{\lambda^y}{\mu_1\mu_2\mu_3 \dots . \mu_y} P_0 \ with \ \mu_x = \frac{x}{\tau}$$

Finally, the probability of congestion is therefore:

$$\mathrm{P_n} = \frac{\frac{\mathrm{A^n}}{\mathrm{n!}}}{\sum_{\mathrm{i=0}}^{\mathrm{n}} \frac{\mathrm{A^i}}{\mathrm{i!}}} = E_n(A) = B$$

This formula is known as the Erlang B formula.

The computer calculation B formula can make use of the following recurrence relation:

$$\frac{1}{E_n(A)} = 1 + \frac{n}{A} . \frac{1}{E_{n-1}(A)} \quad with \ E_0(A) = 1$$

B. Concerning the Erlang Formula C

B1. Standard Model of Erlang C

Although the field of use of standby modes of operation with less extensive than that of the farm to dropped calls, it is nonetheless an area long used in telephone networks, and more recently in data transport networks and computer networks.

Indeed models with timeout are systematic application in the management of central units switches, CPUs known to receive instructions from subscribers, subscribers they provide part of the management.

1) –B2. Calculation of State Probabilities

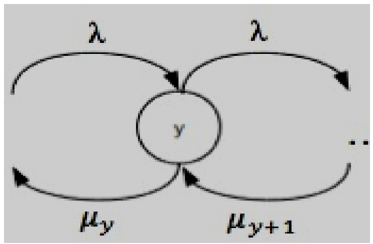


Figure 2. linear Markov chain model application without loss.

The stationary probabilities are:

$$\begin{cases} P_y = \dfrac{\frac{A^y}{y!}}{\sum_{i=0}^n \frac{A^i}{i!} + \frac{A^n}{n!} \cdot \frac{A}{n-A}}, y \le n \\ P_y = \dfrac{\frac{A^n}{n!} \left(\frac{A}{n}\right)^{y-n}}{\sum_{i=0}^n \frac{A^i}{i!} + \frac{A^n}{n!} \cdot \frac{A}{n-A}}, y \ge n \end{cases}$$

Where boundary y = n is to wait for the next request, thus this case is included in the probability of waiting.

$$P_a = \sum_{y=n}^{\infty} P_y = \frac{\frac{A^n}{n!} \sum_{y=n}^{\infty} \left(\frac{A}{n}\right)^{y-n}}{\sum_{i=0}^n \frac{A^i}{i!} + \frac{A^n}{n!} \cdot \frac{A}{n-A}}$$

Similarly, $\sum_{y=n}^{\infty} \left(\frac{A}{n}\right)^{y-n}$ the sum of a geometric progression of reason v

$$\sum_{y=n}^{\infty}\left(\frac{A}{n}\right)^{y-n}=\sum_{k=0}^{\infty}\left(\frac{A}{n}\right)^{k}=\frac{1}{1-\frac{A}{n}}$$

From where:

$$\frac{1}{E_n^a(A)}=1+\frac{n-A}{A}\frac{1}{E_{n-1}(A)}$$

This formula is known as the Erlang C formula.

V. DIMENSIONING ALGORITHMS

In this section we use the formulas discussed in the previous chapter to develop algorithms that will be used later in the program.

A. Algorithm for Calculating the Blocking Probability

Erlang B formula gives the blocking probability depending on traffic (A) and the number of resources (B).
We use the recurrent form of this formula.

```
erlanB(trafic A,  resources N){
        Variables: A, invB in real;
        N, i in Integer;InvB=1;
        for i equals 1 to N
Return   ;}
```

B. Algorithm for Calculating the Number of Resources

```
Number_Of_resourcesB (double A , double B){
        Variables: N in Integer;
                    B' in real ; N=1 ;
              Do {  B'=erlangB (A,N);N++; }
                While (B'>B);
 return (N-1);}
```

C. Algorithm for Traffic Calculation

```
trafficB (int N , double B){

              Variables:  incrementation,  A,  B'  in
        real ;
              incrementation =0.001;  A = 0.000;
               Do {  A = A+ incrementation;
                  B' = erlangB(A,N); }
               While (B'<B);
        return (A- incrementation);}
```

D. Algorithm for Calculating the Waiting Probability: Erlang C Formula

The Erlang formula C gives the probability of waiting according to the traffic and the number of A resource N.

```
erlangC(double A,int N){
                double inv C=0;
                if (A<N){
                 double B = erlangB(A,N-1);
                inv C=1+ ((N-A)/ (A*B)); }
return(1/invC); }
```

E. Algorithm for Calculating the Number of Resources

```
    Number_Of_resourcesC (double A, double C) {
                variables: int N = 1;
                double C';
                  do { C'=erlangC(A,N);N++; }
                while (C'>C);
return (N-1); }
```

F. Algorithm for Traffic Calculation

```
TrafficC(int N,double C){
                double incrementation =0.001;
                double A = 0.000;
                double C';
                do {  A = A+incrementation;
                      C' = erlangC(A,N); }
                While (C'<C);
return (A-incrementation); }
```

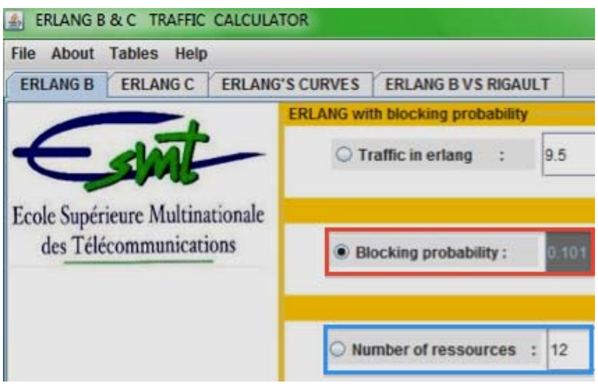
G. Simulations

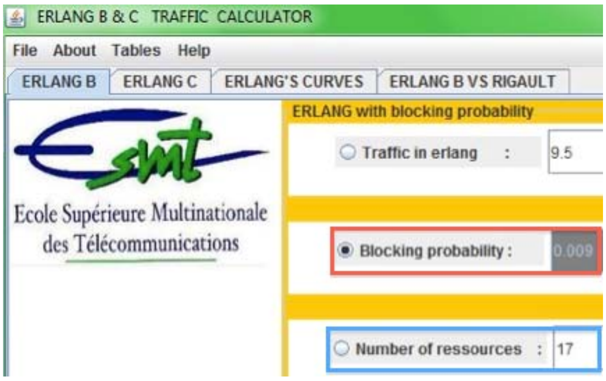
G.1 Erlang B Model

Erlang B formula can express easily the QoS entrained by the addition of a device or degradation when the traffic increases.

Calculation of the blocking probability

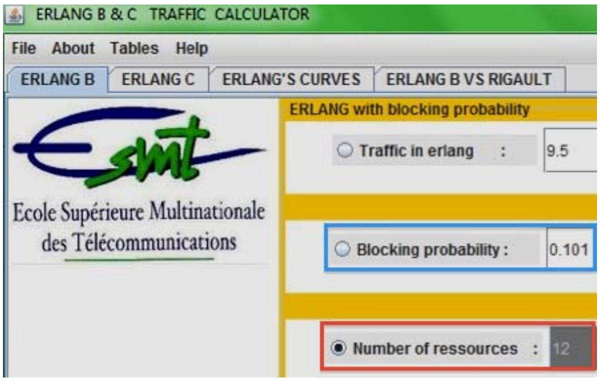
We can see that when offering **9.5** Erlangs traffic to a group of **12** servers, they work with a blocking probability of **10.1%.** And if we increase the number of 5 servers, loss probability drops to 0.9%

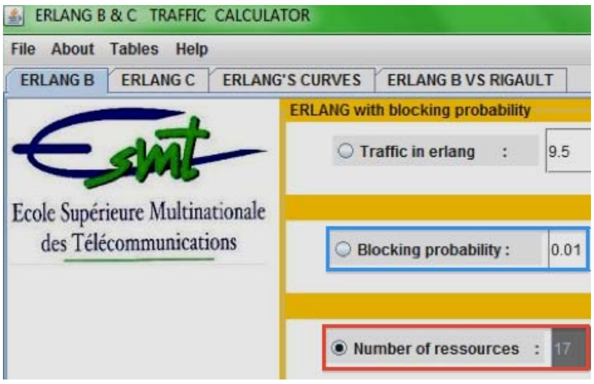




Calculation of the Number of Resources

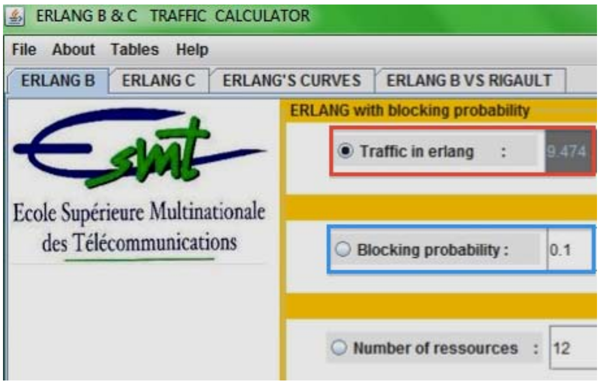
For example to cope with 9.5 Erlangs, 12 servers operating with a risk of rejection of 10% is required. And if we want to improve the quality of service by putting the blocking probability to 1%, we should increase 5 servers.

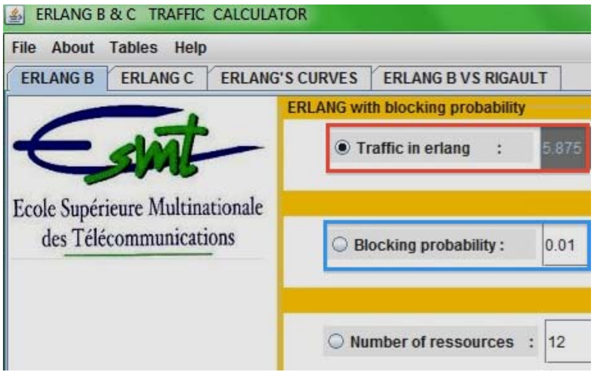




Offered Traffic Calculation

For a group of 12 servers operating with a blocking probability of 10% the offered traffic is 9.474 Erlangs. And if the blocking probability is 1% the offered traffic becomes 5.875 Erlangs.





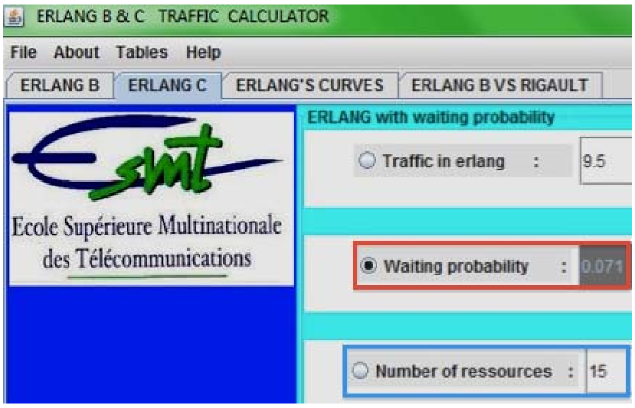
Referring to the results obtained for the simulation on Erlang B formula, following the representation of the number of resources based on the offered traffic following a certain QoS, we can concluded that we must provide more resources if we want a low probability of loss. We also noticed that the QoS decrease for high value of traffic and blocking probability.

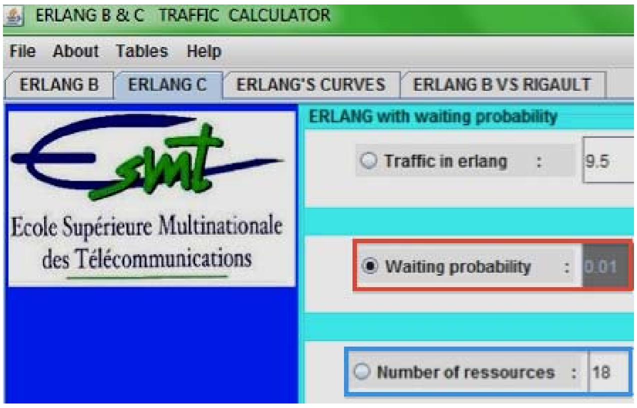
G.2 Erlang C Model

Erlang C formula can express easily the QoS entrained by the addition of a device or degradation when the traffic increases.

Calculation of the Blocking Probability

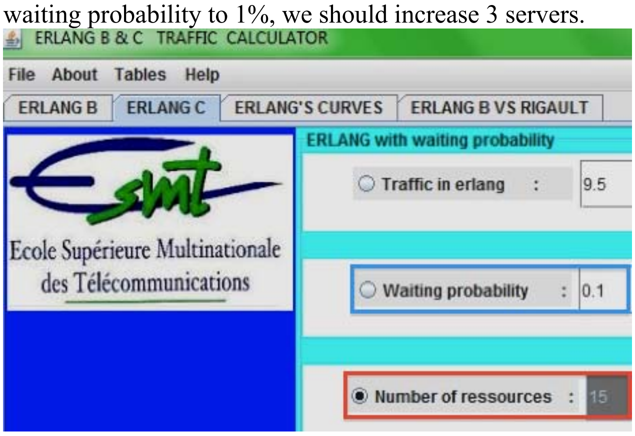
We find that when offering 9.5 Erlangs traffic to a group of 15 servers, they work with a waiting probability of 7.1%. And if we increase the number of 3 servers, the waiting probability fall to 1%.

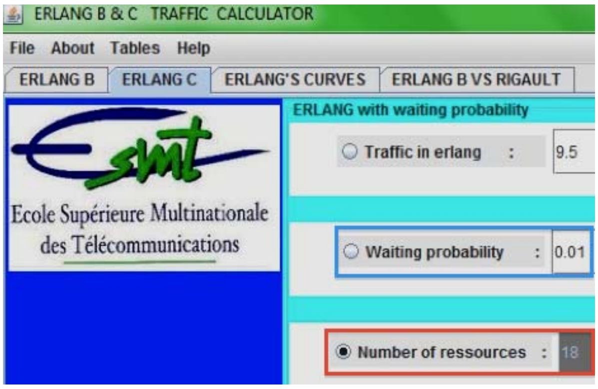




Calculation of the Number of Resources

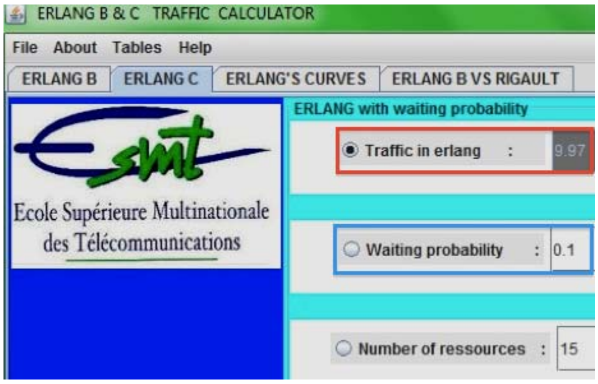
For example to cope with 9.5 Erlangs, 15 servers operating with a risk of waiting of 10% are necessary. And if we want to improve the quality by increasing the waiting probability to 1%, we should increase 3 servers.

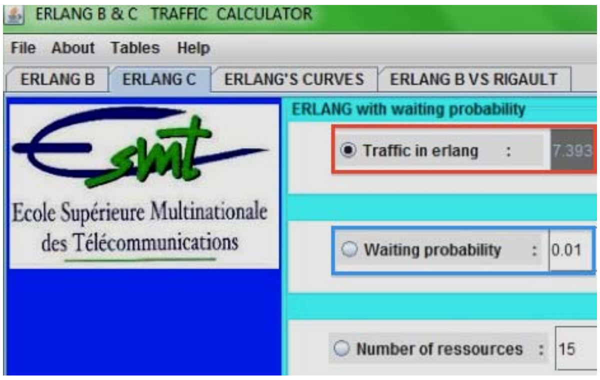




Offered Traffic Calculation

For a group of 15 servers running with a waiting probability of 10%, the offered traffic is 9.97 Erlangs. And if the waiting probability is 1% the offered traffic becomes 7.393 Erlangs.





Referring to the results obtained for the simulation on Erlang C formula, following the representation of the number of resources based on the offered traffic following a certain QoS, the results used to simulate the running of these two models are conformed to Erlang's statistics and Erlang's tables published by the ITU.

To verify that the values returned by the software are correct, several tests were compared to values of the

Erlang's curves and tables published by the ITU. After these tests the results are satisfactory.

VI. LIMITATIONS OF THE FORMULAE

We noticed that the Rigault's formula is simple and easily manageable than Erlang B and therefore is widely used in the dimensioning calculations.

Observations

- **When B is a power of 10:**

B from 0.0001 to 0.1 and a range of traffic from 1 to 1000 Erlangs and from 1001 to 2000 Erlangs.

Required parameters are missing or incorrect.

Figure 9. Graph for B=0.01 ; 0.001 ; 0.0001et A=[0 ;1000]

Required parameters are missing or incorrect.

Figure 10. Graph for B=0.1 et A=[0-1000]

- **When B is not a pure power of 10:**

B from 0.0005 to 0.5 1 and a range of traffic from 1 to 1000 Erlangs and from 1001 to 2000 Erlangs.

Required parameters are missing or incorrect.

Figure 11. Graph for B=0.005 et A=[1001-200]

Required parameters are missing or incorrect.

Figure 12. Graph for B=0.2 ; 0.3 ; 0.5 et A=[1001-2000]

VII. CONCLUSION

From the beginning of the first switch and telecommunication networks has raised the question of their quantitative sizing. Empirical origin, the answer very quickly became the subject of a formal mathematical approach to reflect the growing economic stakes associated with the rapid expansion of telecommunications networks and services. It is very common in a telecommunications network to make sharing of resources, whether transmission channels, highly specialized routes, especially for control and signaling, and switching devices.

This implies a trivialization speculation statistics on the number of unmarked resources to plan and a risk of congestion when all available resources are occupied simultaneously causing temporarily unable to meet the demand for services.

Based on voice traffic model whose characteristics are well known, we have proposed a design tool that allows assessing the risk and providing the means to control a correct dimensioning of the network in a constant compromise between cost and quality of service. Through this tool, we made a comparative study between the Erlang B formula and rule Rigault after which we made recommendations regarding the use of the rule Rigault. Following the obtained, you should start thinking about Erlang when you need applications that:

- Handle larger number of concurrent activities
- Are easily distributable over a network of computers
- Scale with number of machines in the network
- Are fault-tolerant to both software and hardware error

The exploration models based on traffic data would be a possibility to extend the functionality of the tool to design networks said "all IP".

REFERENCES

[1] "Mnesia - A distributed robust DBMS for telecommunications applications". First International Workshop on Practical Aspects of Declarative Languages (PADL '99): 152–163. 1999.

[2] G. Doyon, Systèmes et réseaux de télécommunications en régime stochastique, Dunod Mars 1989, 677 pages.

[3] Armstrong, Joe; Virding, Robert; Williams, Mike; Wikstrom, Claes (16 January 1996). Concurrent Programming in Erlang (2nd ed.). Prentice Hall. p. 358. ISBN 978-0-13-508301-7.

[4] http://www.itu.int/ITU-D/study_groups/SGP_1998-2002/SG2/StudyQuestions/Question_16/RapporteursGroupDocs/teletraffic.pdf, last accessed on the 5th of Oct 2014.

[5] http://www.itu.int/ITU-D/study_groups/SGP_1998-2002/SG2/StudyQuestions/Question_16/Q16Index-fr.html, last accessed on the 5th of Oct 2014.

[6] http://www.itu.int/ITU-D-StGrps/SGP_1998-

[7] 2002/SG2/StudyQuestions/Question_16/Q16Index.htm, last accessed on the 5th of Nov 2014.

[8] http://www.tarrani.net/linda/ErlangBandC.pdf, last accessed on the 5th of Dec 2014.

[9] http://gestion.coursgratuits.net/technique-de-gestion/theories-des-files-d-attentes.php, last accessed on the 5th of Fev 2015.