

Erlang's B & C-Formulas: Determining Optimal Number of Organs

James K. Tamgno, Mamadou Alpha Barry, Simplicie 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 summarizes the work performed to study Poisson distribution model, Erlang distribution model and constraints of dimensioning a network related to these models. These studies paper also describe the planning tool of a network based on the models considered. These studies have let us to the development of a tool for planning network which also highlighted the limits of the use of the rule of Rigault.

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. The control network is the most difficult problem of the telecommunications engineer by the realities and concepts in question, the greater its economic and social consequences, the most traditional and modern at the same time. Indeed looking for a high quality service, more and more sought after by users becomes a major orientations of the structures of public and private services that are to be effective and competitive, while at the same time, networks of more numerous, the internalization of growing, questioning the quality of service, up to that of a single product - the phone - which uses technologies and concepts relatively simple but proven.

Faced with these new concepts, techniques, technologies, diversified within a single structure - the telecommunications network - the mind is naturally led to seek synthesis. Service needs are essentially random and obedient to various processes, the basic design of systems (nodes and arcs networks) can be obtained only through statistical laws between the limits of cost and quality of service made. The design stress must not only take into account the basic systems themselves but also their mode of association and management procedures, among other things how to carry traffic.

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 RIGAUULT. Section five concludes the paper and outlines our future work.

II. 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 organs.

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

III. Objectives and methodology

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

To achieve these goals 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 organ 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 RIGAUULT.

IV. 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.

To find out the blocking probability B, we need to know the offered traffic and the number of resources. We call "erlangB" the function that returns the value of the blocking probability. Its parameters are the offered traffic and the number of resources.

A.1 Description of the function erlangB

```
erlangB(traffic A, resources N){
    Variables: A, invB in real;
    N, i in Integer; InvB=1;
    for i equals 1 to N
```

Return ;}

A.2 Illustration of the function erlangB

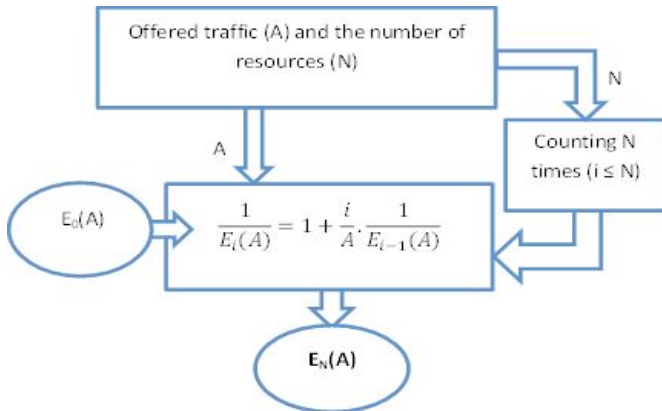


Figure 1. Illustration of the function erlangB.

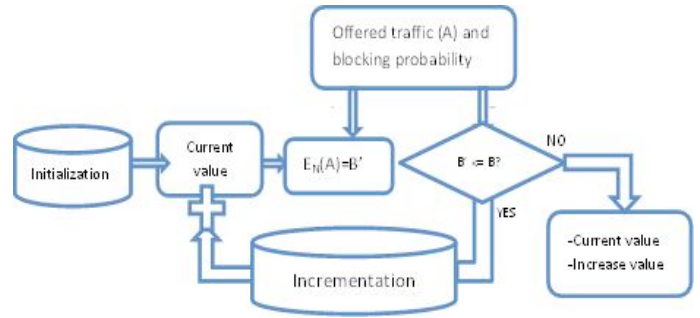
B. Algorithm for calculating the number of resources

To determine the number of resources to consider for an offered traffic and for a certain blocking probability B, we increment by one (1) from one (1), the value of N and for each of them we calculate the blocking probability B', we compare it to B. The first value of N for which B' is less than or equal to B will be selected as the number of required resources. We call "number_Of_resourcesB" the function that returns the number of resources. Its parameters are the offered traffic (A) and the blocking probability (B).

B.1 Description of number_Of_resourcesB function

```
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);}
```

B.2 Illustration of the function number_Of_resourcesB



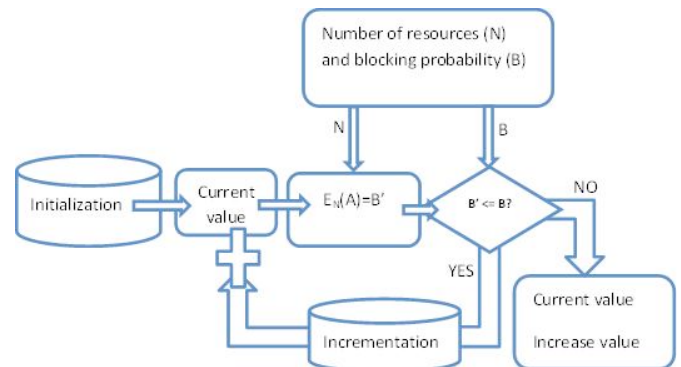
C. Algorithm for traffic calculation

To assess the offered traffic to a group of resources operating at a certain blocking probability B, we increment by 10^{-3} , from zero, the value of traffic and for each of them we calculate the blocking probability B' that we compare to B. The first value of the traffic for which B' is greater than or equal to B will be selected as the offered traffic. We call "trafficB" the function that returns the offered traffic. Its parameters are the number of resources involved (N) and the blocking probability (B).

C.1 Description of the function trafficB

```
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);}
```

C.2 Illustration of the function trafficB



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.

We use this formula in its form expressed according to the blocking probability B.

To find out the waiting probability C, we need to know the offered traffic and the number of resources. We call "erlangC" the function that returns the value of the blocking probability. Its parameters are the offered traffic and the number of resources.

D.1 Description of the function *erlangC*

```
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); }
```

D.2 Illustration of the function *erlangC*

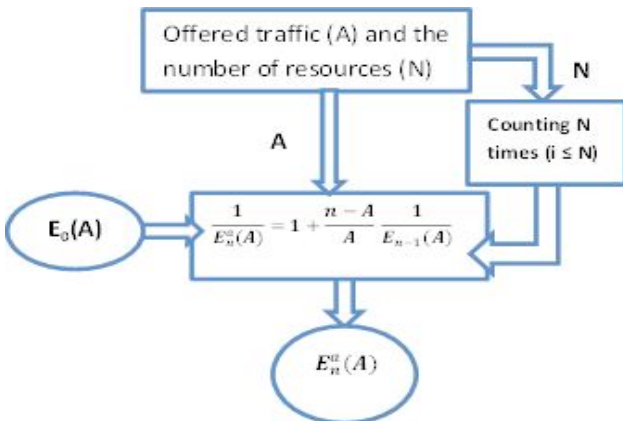


Figure 4. Illustration of the function erlangC

E. Algorithm for calculating the number of resources

To determine the number of resources to consider for an offered traffic and for a certain waiting probability C, we increment by one (1) from one (1), the value of N and for each of them we calculate the waiting probability C', we compare it to C. The first value of N for which C' is less than or equal to C will be selected as the number of required resources. We call "number_Of_resourcesC" the function that returns the number of resources. Its parameters are the offered traffic (A) and the waiting probability (C).

E.1 Description of the function *number_Of_resourcesC*

```
Number_Of_resourcesC (double A, double C){
```

```
    variables: int N = 1;
```

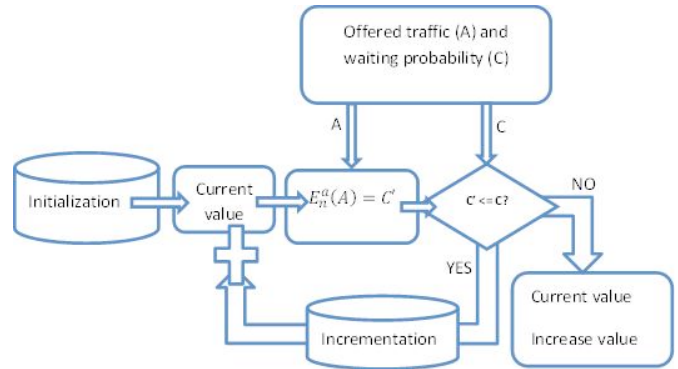
```
    double C';
```

```
    do { C'=erlangC(A,N);N++; }
```

```
    while(C'>C);
```

```
    return (N-1); }
```

E.2 Illustration of the function *number_Of_resourcesC*



F. Algorithm for traffic calculation

To assess the offered traffic to a group of resources operating at a certain waiting probability C, we increment by 10^{-3} , from zero, the value of traffic and for each of them we calculate the waiting probability C' that we compare to C. The first value of the traffic for which B' is greater than or equal to B will be selected as the offered traffic. We call "trafficC" the function that returns the offered traffic. Its parameters are the number of resources involved (N) and the waiting probability (C).

F.1 Description of the function *TrafficC*

```
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); }
```

F.2 Illustration of the function trafficC

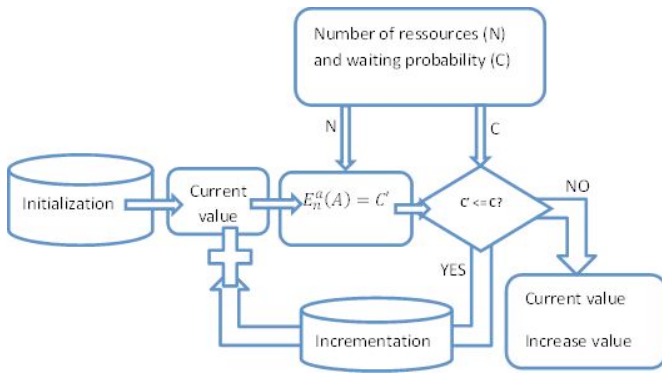


Figure 6. Illustration of the function trafficC

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%

Number of resources	12 Servers	17 Servers
Traffic in Erlang	9.5	9.5
Blocking probability	10.1%	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.

Number of resources	12 Servers	17 Servers
Traffic in Erlang	9.5	9.5
Blocking probability	10.1%	1%

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 offred traffic becomes 5.875 Erlangs.

Number of resources	12 Servers	12 Servers
Traffic in Erlang	9.474	5.875
Blocking probability	10.1%	1%

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%.

Number of resources	15 Servers	18 Servers
Traffic in Erlang	9.5	9.5
Blocking probability	7.1%	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.

Number of resources	15 Servers	18 Servers
Traffic in Erlang	9.5	9.5
Blocking probability	10%	1%

Offered traffic calculation

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

Number of resources	15 Servers	15 Servers
Traffic in Erlang	9.97	7.393
Blocking probability	10.1%	1%

G.3 Erlang's curves

G.3.1 Erlang B curves

Erlang curves are a representation of the number of resources based on the offered traffic following a certain QoS.

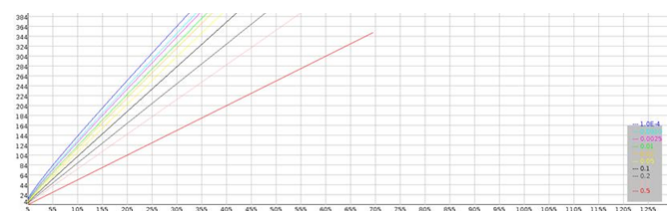


Figure 7. Erlang B curves

We noticed that we must provide more resources if we want a low probability of loss.

We also noticed that the curves tend towards nearly straight lines for high value of traffic and blocking probability.

G.3.2 Erlang C curves

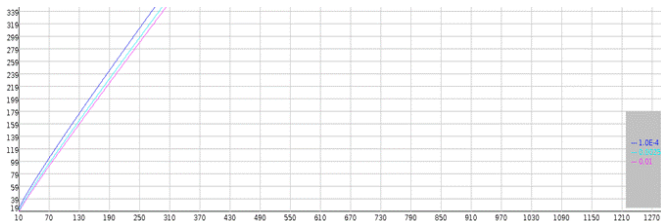


Figure 8. Erlang C curves

The results used to simulate the running of these two models are conformed to Erlang's curves 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.

V. STUDY LIMITATIONS OF FORMULA

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.

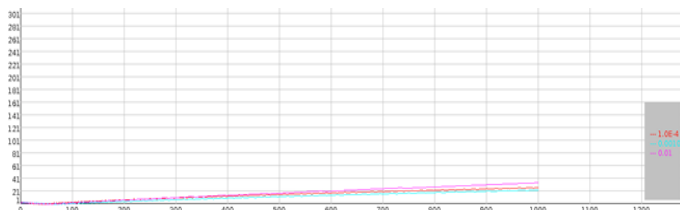


Figure 9. Graph for $B=0.01 ; 0.001 ; 0.0001$ et $A=[0 ; 1000]$

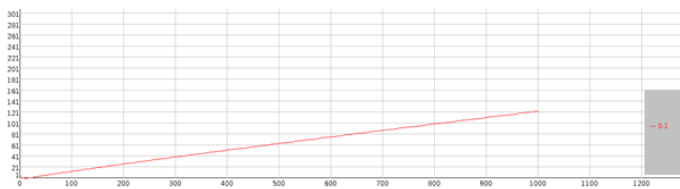


Figure 10. Graph for $B=0.1$ et $A=[0-1000]$

➤ 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.

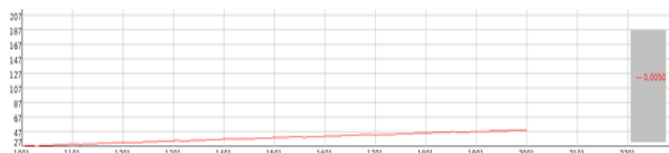


Figure 11. Graph for $B=0.005$ et $A=[1001-200]$

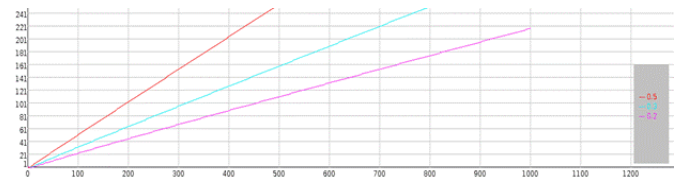


Figure 12. Graph for $B=0.2 ; 0.3 ; 0.5$ et $A=[1001-2000]$

VI. INTERPRETATION

According to the results obtained, we can draw two observations:

Note 1 :

In this case, the difference between the Erlang B formula and Rigault's formula in terms of number of resources is low for low QoS, but this difference increases considerably when the offered traffic increases or when moving away of 0.1%.

Note 2 :

Here as before the difference is small for very low values of QoS and increases very rapidly, on the order of hundreds, when QoS increases.

VII. RECOMMENDATION

After observations, analysis and interpretation of data output from the application, we are able to give the limits of use of the Rigault's formula.

Recommendation 1 :

For QoS, the use of Rigault formula is not recommended for traffic value exceeding 200 erl and QoS greater than 1%.

Recommendation 2 :

For QoS, the use of Rigault formula is not recommended for QoS greater than 1%.

VIII. 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 organs, 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.

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] J. Ma and L. Deng, "Target-directed mixture linear dynamic models for spontaneous speech recognition. IEEE transactions on speech and audio processing", vol. 12, pp. 31-38, January 2004.
- [5] Guoping Zeng (June 2003), Two common properties of the erlang-B function, erlang-C function, and Engset blocking function, Elsevier Science, retrieved 2011-02-03.



Dr. James K. Tamgno, As a senior lecturer at ESMT, i obtained my Ph.D in may 2014 at University Cheikh Anta Diop of Dakar. My researches focused on Promotion of African's languages and dialects and Contribution to a Wider Accessibility of ICT, by Circumventing Language Barriers and Lessening the Impact of Illiteracy or Disability. Previously I have gotten a M.Sc. in Engineering Science from University Cheikh Anta DIOP of Dakar, studied numerical

analysis and mathematics at the University of Yaoundé, and worked as a research scientist at ESMT (Multinational High School of Telecommunication) of Dakar. <http://www.esmt.sn/pages/personnel.php?c=J>



Mr. Mamadou Alpha Barry is a senior lecturer at the Ecole Supérieure Multinationale des Télécommunications (ESMT) of Dakar. He obtained a Master 2 professional at University of Rouen, France in the field of Consulting and Engineering Training. And a Master of research of Engineering Science on Computer modeling simulations of complex systems. I had made many expertise on and training on

telecommunications systems (switching, ISDN, ATM, architecture and NGN and IMS services). Elsewhere, he studied the implementation of numbering plans (Management of scarce resources) and audit multiple network system such as ISDN, ATM, NSS.



Mr. Simplic E. Gnang was a student at Ecole Supérieure Multinationale des Télécommunications of Dakar. Telecoms and IT systems consultant, I worked on many research projects in Telecommunications, QoS and network optimisation with ESMT Dakar (Ecole Superieure Multinationale des Telecommunications) where I did my Master's Degree in telecommunications and network engineering. I currently work at Huawei Technologies as Service Engineer in applications and softwares