****

**Department of Electrical Engineering,**

**Syed Babar Ali School of Science and Engineering,**

**Lahore University of Management Sciences, Lahore, Pakistan**

**Senior Year Design Project (Sproj) Proposal**

**“Call Center Simulation Using NS-3”**

**Submitted by**

**Ammarah Azmat Bajwa Roll No: 18100243**

**Hira Tariq Roll No: 18100215**

**Zarmeen Khan Roll No: 18100273**

**Muhammad Shamaas Roll No: 18100217**

**Under the supervision of**

**Dr. Tariq Mahmood Jadoon**

Designation: Associate Professor

Email:  jadoon@lums.edu.pk

**Signatures of Approval**

**Project Advisor:**

**Project Co – Advisor(s) (if any):**

Contents

[1 Problem Statement 4](#_Toc493866480)

[2 Literature Review 5](#_Toc493866481)

[3 Design and Implementation 7](#_Toc493866482)

[3.1 Block Diagrams 10](#_Toc493866483)

[4 Relevance towards Solution to Societal/Country Problems 11](#_Toc493866484)

[5 Project Goals and Objectives 12](#_Toc493866485)

[6 Proposed Timeline 13](#_Toc493866486)

[7 References 15](#_Toc493866487)

# Problem Statement

Call center modeling is a growing research field. They are an example of queuing systems. Queuing models help estimate ways to optimize staffing and maximize performance in a call center. We used NS-3, which is a discrete event network simulator, for our modeling. Calls arrive, wait in a virtual queue and then they are serviced via a routing mechanism. The most common model used at large both by practitioners and academics is the Erlang C model. The model however works under many assumptions mainly: calls arrival follows a Poisson distribution with a known average rate, callers wait infinitely without abandoning the call. Erlang X: an extension of Erlang C is more complex and realistic as it incorporates call abandonment, heavy call traffic, call drops and skill based routing. We will attempt to improve Erlang X with additive variables, which makes it close to a realistic call center. We will verify it by comparing simulation results against the performance records of a real call center.

# Literature Review

In the paper "Experience-Based Routing in Call Center Environments" the author Thomas R. Robbins has highlighted the issues in the oversimplified queuing model and has suggested another routing scheme for the call center environments. Queuing system, is implemented in all the call centers. If the agent is busy then the call is placed in the queue and waits there until the agent gets free. Mostly call centers are observed as having a "standard queuing model" which is basically an oversimplification of the practical call center (Perros 2009, Iversen 2001). The "standard queuing model" is based on two assumptions. First, all the agents are assumed to be "homogenous" in their services i.e. having same set of skills and speed to answer the call. Secondly, calls routing is done in such a way that the call is forwarded to the agent who is idle for the longest interval of time. This oversimplification affects the performance of the call center by increasing the average waiting time of the queue. However, the performance can be enhanced by introducing some changes in the routing scheme. Agent requires particular skills in handling a certain call efficiently which depend on their experience as well. More the experience of the agent in handling calls lesser will be the time he will take to service the call, thus the average waiting time in the queue will decrease (Robbins 2015). Robbins has suggested "experience-based routing" during the peak hours that is when the call arrivals are high then the calls should be delivered to the experienced agents so that they can efficiently manage the load by servicing the calls in lesser time. However, when the arrival rate of calls is small then, according to Robbins, calls should be routed to the non-experienced agents so that they can learn how to service the calls and could gain some experience. This routing scheme not only improves the "system performance" and the "customer service" but also enhance the learning of the newly hired agents (Gans et al. 2003).

Thomas R. Robbins, D.J. Medeiros and Terry P. Harrison in their paper “Evaluating the Erlang C and Erlang A Models for Call Center Modelling” have compared the queuing models that are widely used in the call center analysis. First is the Erlang C model, many assumptions in the model simplifies it. This model assumes that the agents are “statistically identical” all have the same speed and ability to answer the call. Also, the calls which arrives according to the Poisson distribution are having a known average rate of arrival (Keshav 2007). Moreover, it assumes that the size of the queue is infinite and calls can wait for any large amount of time inside the queue without abandoning it. It means that the average waiting time could be large enough which is quite unrealistic and can have a drastic impact on the customer service and call center operations. However, in Erlang A model, the extension of the Erlang C model, abandonments are considered. The callers have some finite patience to wait in the queue and if it turns out the caller abandons the queue (Koole 2007). “Quality-driven and the Quality and efficiency-driven (QED) regimes”, where the call centers are capable enough to handle all the calls and there is no abandonment, are considered in this paper and the performance metrics for both Erlang A and Erlang C in QED regimes are being compared. Erlang A model though give better results but it does not depict the accurate staffing requirement in the cases when the arrival rates have significant uncertainties (Gans et al. 2003). In such cases Erlang C is used in making the staffing decisions. However, in case of the “Efficiency-driven regime” when the arrival rate is large and all agents are occupied in servicing the calls here the Erlang C model simply gets collapsed because there are significant abandonments. Erlang A model, which allows the abandonments, is used to predict the performance metrics in this kind of regime (Iversen 2001).

# Design and Implementation

We are implementing a Simulation of an Optimized Call Centre Model on NS3 Software which is a discrete event simulator (Jerry). We have implemented packet switching in our simulation. This has been done in the software, by allocating a caller node, distributor and Agent Node. As the following is a representation of a more realistic model, so we have implemented Erlang X. It in cooperates queue abandonment of callers, unpredictable call traffic, call drops and also skill based routing. Fastest Server First, Longest Idle Server First and Experience Based routings each in different circumstances have been used to service calls in our model (Robbins et al.).

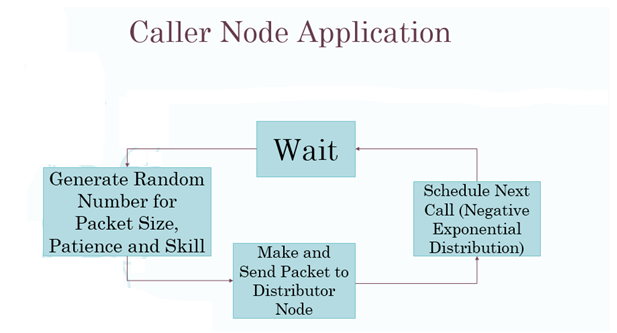
The Call Centre structure comprises of three fundamental entities; the Caller Node, the Agent Node and the Distributor Node. Having built a structure for the Agents, we have furtherly differentiated them on the basis of their skills, no of calls serviced and operator number in our model. The callers have been differentiated on the basis of call times, patience tag, waiting time in the queue and the skill type they require. While a common distributor node (buffer) has also been made which Enqueues arriving packets and Dequeues packets to the agent node.

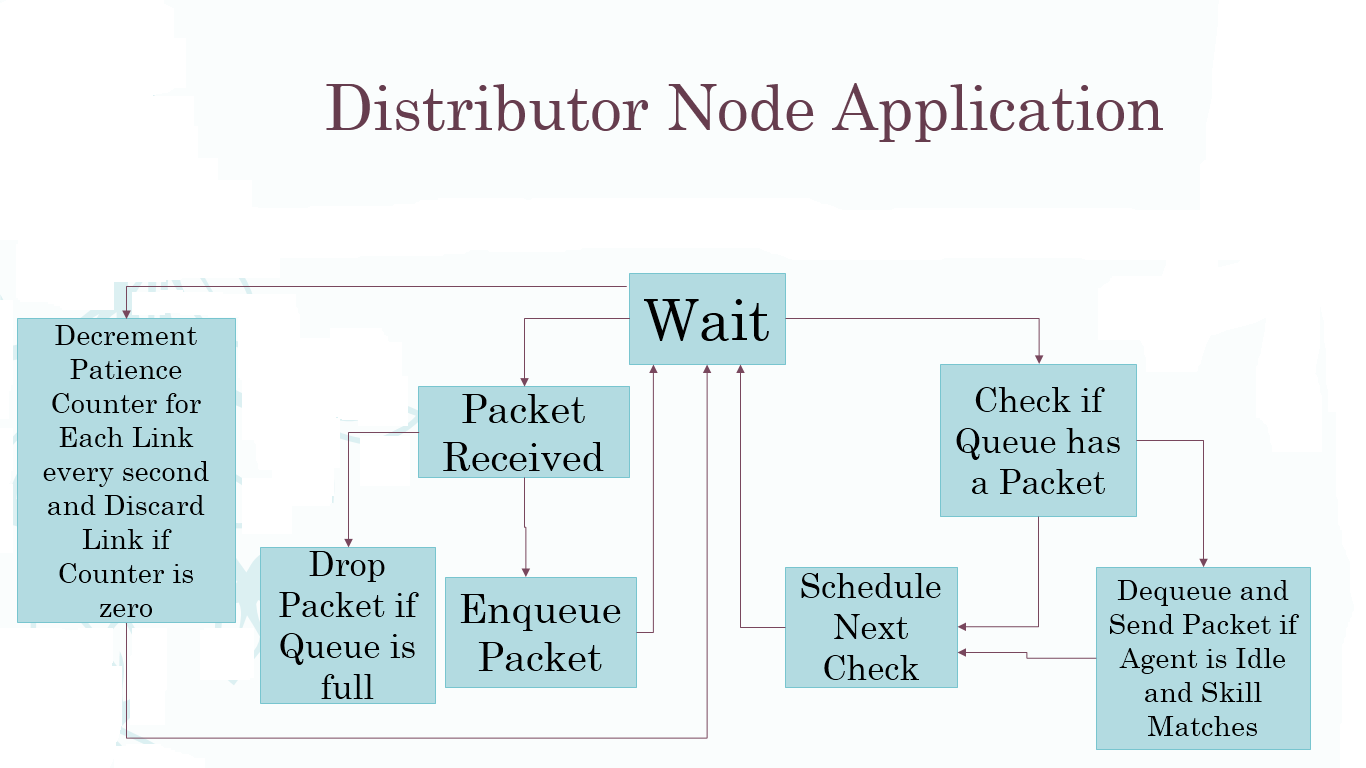
Erlang C is a simple multi-server queuing system; calls arrive according to a Poisson process at an average rate. We verified Erlang C Formula with experimental results for mean waiting time of callers in Queue using the Ger Koole calculator (“Erlang C Calculator.”, “Erlang X Calculator.”). Our goal is to verify Erlang X with additive variables which makes it close to a realistic call center through empirical analysis involving comparison of performance with real call center records (Mandelbaum 2002).

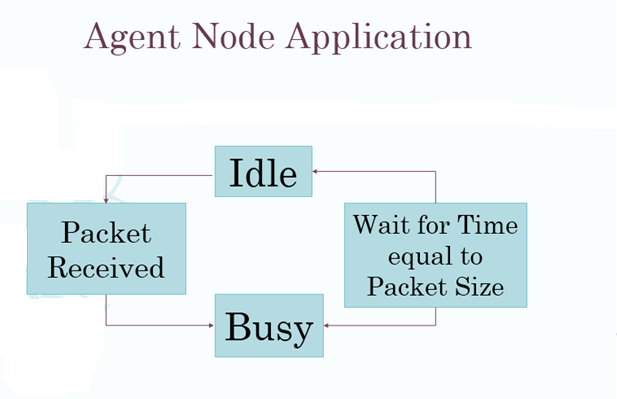
Generally, abandonment rates cannot be estimated directly using the Erlang C model because the model assumes no abandonment occurs. Our model will assume that each caller has a finite willingness to wait, and will abandon the queue or hang up if their wait time exceeds their patience. We plan to achieve this by assigning each caller a patience counter which is decremented after each second. Erlang X is modeled with limited available number of telephone lines thus calls will be dropped if the waiting queue is full. In addition to finite queue, we route our distributor in such a way that it can pick a blocking packet from anywhere in the queue and send it for service to prevent queue blockage (*Ns-3 Tutorial*, *Ns-3 Documentation*.). Furthermore, the agents in our model will also have multiple skills. There are two categories: Specialists, having only one skill and Generalists, having expertise in more than one skill (Koole 2007). We conclude by verifying experimentally that even with complex variables, we can achieve optimization in Erlang X just as that in Erlang C.

The strategy we have used is as follows: Decrement Patience Counter of each Link after every second. Remove Link from Queue if Timer is zero. Before Enqueue, check if queue is full. If queue is full, drop packet. Keep a Skill section in each Link to aid distribution of packets to operators. To prevent one packet from blocking entire queue, allow distributors to pick packet from any link of queue.

The working of Caller, Agent and Distributor Node is depicted in the Block Diagrams Below.

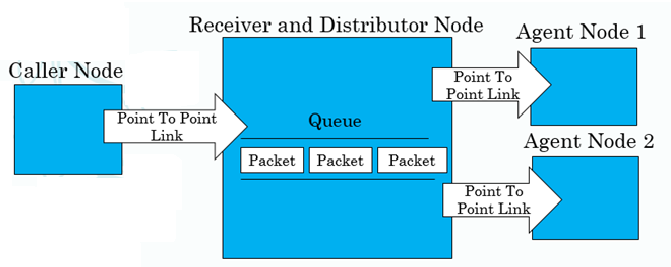






## Block Diagrams

*STRATEGY*



*System Model*

Change Model to include randomness

Limit queue, Patience counter, Skill section

Queue drop & block prevention mechanism

Change Node application mechanism

Perform experiments with new model

Represent and compare results

Evaluate Simulation Optimization

# Relevance towards Solution to Societal/Country Problems

The solution for optimizing call center performance metrics like mean Queue Waiting Time, Service Level, Total Problem Resolution and Agents Utilization by tuning Total Agents Employed, scheduling of agents and Dynamic Caller Routing scheme based on traffic intensity can be used to improve the real-life call center efficiency. Even though the behavior of a real call center is much more complex and the model simplifies the system by many assumptions about caller and agent behavior, we have tried to make the model more realistic by including variable call arrival rates, variable propagation delays, queue abandonments, call drops and inhomogeneous agents (with variable speed of service, hourly wages, experience and skills) (Gans 2003, Robbins 2015).

As a result, if we input the predicted traffic profile to the simulator, we can achieve the optimal number, performance ratings and schedules of the agents that a call center must hire to meet the desired performance metric goals. The accuracy, validity and applicability of the simulation predictions depends on how many iterations we carry out and how much detail we incorporate about the real call center into the simulation model.

The inherent simplicity and generic nature of the Queue System model lends itself for adaptation in many different environments besides call centers as well (Iversen 2001). Many other queue networks like shopping malls cash counters, pick and drop services, hospitals, printer stations, billing stations, ticket booths, banks, industrial workforce management etc. can benefit from the model. The system can be used to model any queue system with one or more servers that involves time and/ or resource management after a few modifications.

Many Queueing systems like call centers and billing stations are concerned with minimizing waiting time in queue i.e. finding the optimal number of agents needed to achieve the desired service level. The simulation can be run multiple times with different number of agents employed and mean waiting time recorded until the desired result is obtained. For time management, we will implement the fastest server first routing schemes. Another application includes minimizing queue abandonments by Longest Queue Wait First Priority routing scheme. For efficient and maximum utilization of agents we will implement the Skill based routing and Longest Idle First Priority routing (Robbins et al.). This is desired in banks, hospitals and international call centers where differentiated agents and services are provided hence agents specialized in a particular skill are needed.

# Project Goals and Objectives

1. Verify Erlang C and Erlang X models.
2. Calculate Service Level (percentage of calls answered in less than specific time limit), Total Problem Resolution, Net Promoter score, Quality Scores, Agent Salaries (hourly basis), Profit, Utilization of Agents (percentage of time they are busy), Average Caller Time in System, Probability of wait (Arrivals when queue is not empty), Service Rate (Total Calls Serviced / Total Time), Traffic Intensity (Arrival rate \* Service Rate) (“Call Centre.” *Wikipedia 2017*).
3. Plot Queue size versus time.
4. Introduce Virtual Response Unit using negative exponential random generator which waits for random time before enqueuing arrived call.
5. Using call center history (Mandelbaum 2002), make arrays of call durations, interarrival times, waiting times (VRU), waiting times (Service). Convert time format from hour: minute: second to seconds. All times should be in seconds. Calculate mean service time of each agent. Plot Service time vs time for any agent. Calculate mean arrival rate for each 10-minute interval and plot queue size vs time.
6. Make a structure for Agent Nodes to model exhaustion level, experience, speed (dependent on exhaustion and experience), Mean Service Time, wage, time since last service, breaks.
7. Implement Longest Idle First, Fastest Server First, skill based (Robbins et al.), lowest Cost and Longest Queue Wait First Routing schemes.
8. Introduce separate point to point helper for caller-distributor link. Model Delays for international calls for that point to point link using negative exponential random generator (*Ns-3 Tutorial*, *Ns-3 Documentation*.).
9. Simulate real life call traffic using call center records and compare performance metrics for simulation and Real-Life Data. Record data and plot results to explain any mismatch between actual results and predictions of simulation model.
10. Prepare Design and Simulation Presentation

# Proposed Timeline

|  |  |
| --- | --- |
| ***Date*** | ***Objective*** |
| September 18 – October 2 | Verify Erlang C and Erlang X models. |
| October 2 - October 16 | *Zarmeen Khan:* Calculate call center performance metrics and plot Queue Size versus time.  *Hira Tariq:* Introduce additional Call center agent performance measures that must be monitored.  *Ammarah Azmat:* Sorting of Real Life Call Center Data.  *Muhammad Shamaas:* Test performance of each Code. |
| October 16 - October 30 | *Zarmeen Khan:* Complete Coding for plotting results of simulation and Automatic Voice Response Unit.  *Hira Tariq:* Introduce Dynamic Call Routing Schemes and model Variable Delay for International Calls.  *Ammarah Azmat:* Calculate performance metrics from Real Life Call Center Data.  *Muhammad Shamaas:* Test performance of each Code. |
| October 30 - November 13 | Simulate real life call traffic using call center records and calculate performance metrics for simulation. |
| November 13 - November 27 | *Ammarah Azmat:* Record simulation data  *Hira Tariq:* Plot simulation results  *Zarmeen Khan:* Explain any mismatch between actual results and predictions of simulation model.  *Muhammad Shamaas:* Prepare Design and Simulation Presentation |
| November 28 - December 2 | Design and Simulation Group SPROJ-1 Presentation |
| December 9 | Submission of SPROJ-1 Design and Simulation Report after approval from Project Advisor |

# References

[1]. “Call Centre.” *Wikipedia*, Wikimedia Foundation, 17 Sept. 2017, en.wikipedia.org/wiki/Call\_centre. Accessed Sept. 2017.

[2]. “Erlang C Calculator.” *Call Center Optimization*, www.gerkoole.com/CCO/erlang-c.php. Accessed Sept. 2017.

[3]. “Erlang X Calculator.” *Call Center Optimization*, www.gerkoole.com/CCO/erlang-x.php. Accessed Sept. 2017.

[4]. Gans, Noah, et al. *Telephone Call Centers: Tutorial, Review, and Research Prospects*. 2nd ed., vol. 5, Amsterdam, Netherlands, 2003.

[5]. Iversen, Villy B. *Teletraffic Engineering*. Lyngby, Denmark, 2001.

[6]. Jerry, et al. *Discrete-Event System Simulation*. 4th ed., Prentice Hall.

[7]. Koole, Ger. *Call Center Mathematics*. Amsterdam, Netherlands, 2007.

[8]. Keshav, Srinivasan. *Mathematical Foundations of Computer Networking*. 2012.

[9]. Mandelbaum, A. Call Center Data. http://iew3.technion.ac.il/serveng/callcenterdata/index.html. 2002

[10]. *Ns-3 Tutorial*. 2017, www.nsnam.org/docs/release/3.26/tutorial/ns-3-tutorial.pdf. Accessed Sept. 2017.

[11]. *Ns-3 Documentation*, www.nsnam.org/docs/release/3.26/doxygen/index.html. Accessed Sept. 2017.

[12]. Perros, Harry. *Computer Simulation Techniques: The definitive introduction!* Raleigh, North Carolina, 2009.

[13]. Robbins, Thomas R. *Experience-Based Routing in Call Center Environments*. 2nd ed., vol. 7, Greenville, North Carolina, 2015.

[14]. Robbins, Thomas R., et al. *Evaluating the Erlang C and Erlang A Models for Call Center Modeling Working Paper*. Greenville, North Carolina.