Intelligent Network Analysis by Closed Queuing Models

Design Review

Student name & IDs:

Contents

List of Tables and Figures

1. Introduction

1.1 Definitions and abbreviations

Abbreviations:

SSP – Service Switching Point – accommodate terminals

STP – Signal Transfer Point – controls the traffic between the SSP's and the SCP

SCP – Service Control Point – holds the database and offers network services to SSP's.

Definitions:

M/G/1/K/K system – M – Exponential inter-arrival time to SCP

G – General distributed service time in SCP

1 – Single server (SCP)

K – System capacity

K – Size of population.

Average time in system (T) – Average time spent waiting in queue and the service time in SCP.

System state vector ($\bar{n}$) – a vector $\bar{n}$ in which n_i denotes the number of costumers in queue i + server i.

P($\bar{n}$) - a vector in which P(n_i) denotes the probability that the number of costumers in queue i + server i will be n_i. in the system's steady state.

delay – there is no delay between the devices.

Server utilization – the utilization of the SCP

μ - Average rate of costumers leaving SCP.

λ - Average rate of costumers arriving to SCP

Real λ - Actual rate of customer flow through the SCP

C – var/mean^2

1.2 Overview

With the wide deployment of intelligent network (IN) services, there is an urgent need to understand and solve teletraffic performance issues of the evolving network intelligence platform. This paper discusses a queuing system model for the performance analysis of IN call processing. The intelligent network is presented as a network of queues where the total number of customers (e.g., SSPs) is fixed, thus forming a closed queuing network. The IN distributed architecture is modeled as a finite source queuing model – M/G/1/K/K. The expected response time for that model is analyzed and computed. The numerical results and the corresponding curves are provided and, related to open questions, future work is summarized.

With this model we will examine the fallowing system parameters:

1. The average time in system (SCP) against server utilization with vary number of SSP's.
2. The probability to have k customers in system (SCP) with K=20 SSP's.
3. The probabilities to have 0 customers in system (SCP) with vary number of SSP's.
4. The average time is system (SCP) against different service time distributions.

2. Design and Flow

2.1 Overview

As described before the system built of three elements – SSP's, STP and SCP. The program uses three basic modules to create the system figure and connections between the elements as described below:

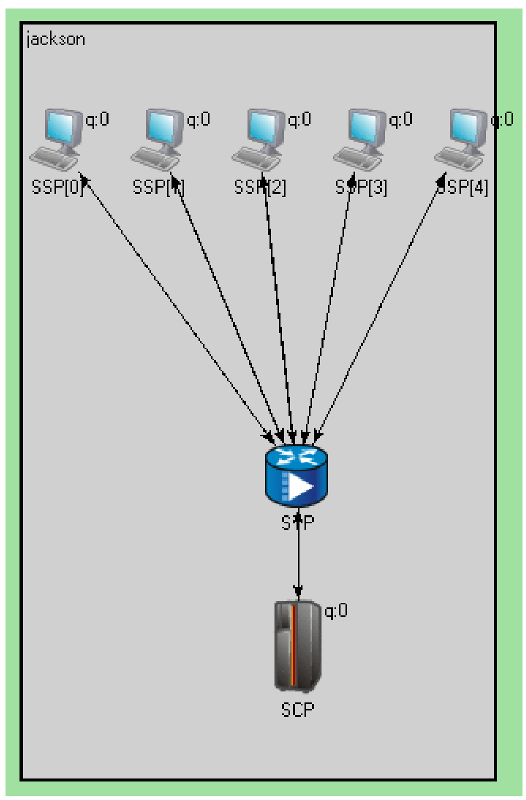
Fifo – Contains a queue and a single server. Each SSP is a part of a Fifo vector module. The service time at each server is i.i.d exponential with parameter **λi.** The "out" gate of each SSP is connected to a vector on "in" gates in the STP which is a Switch module.

Server – Contains a queue and a single server. The service time at each server is user-defined from deterministic, erlang-2, exponential and hyper-exponential. The "out" gate of each SCP is connected to a vector on "in" gates in the STP.

Switch – Made out of 2 vector gates, “in” and “out”, which are used to connect to each of the SSP’s. It also contains “scpIn” gate and “scpOut” gate to connect to the SCP. Messages received from the “in” vector are sent to the “scpOut” gate and messages from the “scpIn” gate are sent randomly to one of the “out” vector gates leading to a single SSP.

Diagram of the system

with 5 SSP's (K=5):



2.2 Modules overview

This part details the omnet modules and emphasis the section in which the statistics gathering is taking place. Because of the nature of the system it is necessary to do all the statistics gathering in the module who has the "all-out picture". This module is the Server.

Simple Module: FIFO (SSP)

Description: contains a bounded buffer and a single server with exponential service time

Variables: cQeueu – handles buffer operations fifo-wise

Functions: initialize() – initialize fifo parameters.

handleMessage(cMessage* msg) – if server is busy, msg is inserted into queue, otherwise, a thinking time is being calculated and after that the msg is passed out to the STP.

Statistics Gathering: None

Simple Module: SERVER (SCP)

Description: contains a bounded buffer and a single server with pre-defined service time

Variables: cQeueu – handles buffer operations fifo-wise

Functions: initialize() – initialize fifo parameters.

handleMessage(cMessage* msg) – if server is busy, msg is inserted into queue, otherwise, a serving time is being calculated and after that the msg is passed out to the STP.

Statistics Gathering: Gathers all statistics required for the final results: time in system, server utilization, real λ, probability of K customers in the system.

Simple Module: Switch(STP)

Description: acts as a network between SSP's and the SCP.

Variables: numGates – defines the number of I/O gates.

Functions: initialize() – initialize numGates.

handleMessage(cMessage* msg) – determines the origin of the msg and forwards it accordingly. Msg's from the SSP's are forwarded to the SCP and vice versa.

Statistics Gathering: None

Compound Module: Jackson

Description: contains all three modules, and represents the closed queuing network made out of SSP's, STP and SCP module.

2.3 Application Interfaces

The parameters defined by omnetpp.ini for each Run:

- service_time - defines the service time/mean service time for the SCP. For deterministic and erlang distributions it defines the service time and for exponential and hyperexponential it defines the mean service time.
- num_SSP – defines number of SSP in network. Default value is 20.
- num_init_jobs – initial number of customers in each SSP is always 1 and in SCP is always 0 but it can be changed from the ini file if needed.
- num_cells – number of cells in all buffers. Default is 100.
- isExpo – distinguishe between exponential distributions and the others in the aspect of service tume.

The parameters defined by user:

- The user can choose between 5 presets of the network distinguished by the distribution of the service time of the SCP.
- The presets are:
  - Run 1 – deterministic
  - Run 2 – erlang 2
  - Run 3 – exponential
  - Run 4 – hyper-Exponential SCP service time, C^2 = 1.5
  - Run 5 – hyper-Exponential SCP service time, C^2 = 2.0

2.4 Results and Test measurements

The figures below are of analytical nature and will be compared to simulation results.

The service time in figure 1 to 3 is exponentially distributed.

1. The average time in system (SCP) against server utilization with vary number of SSP's.

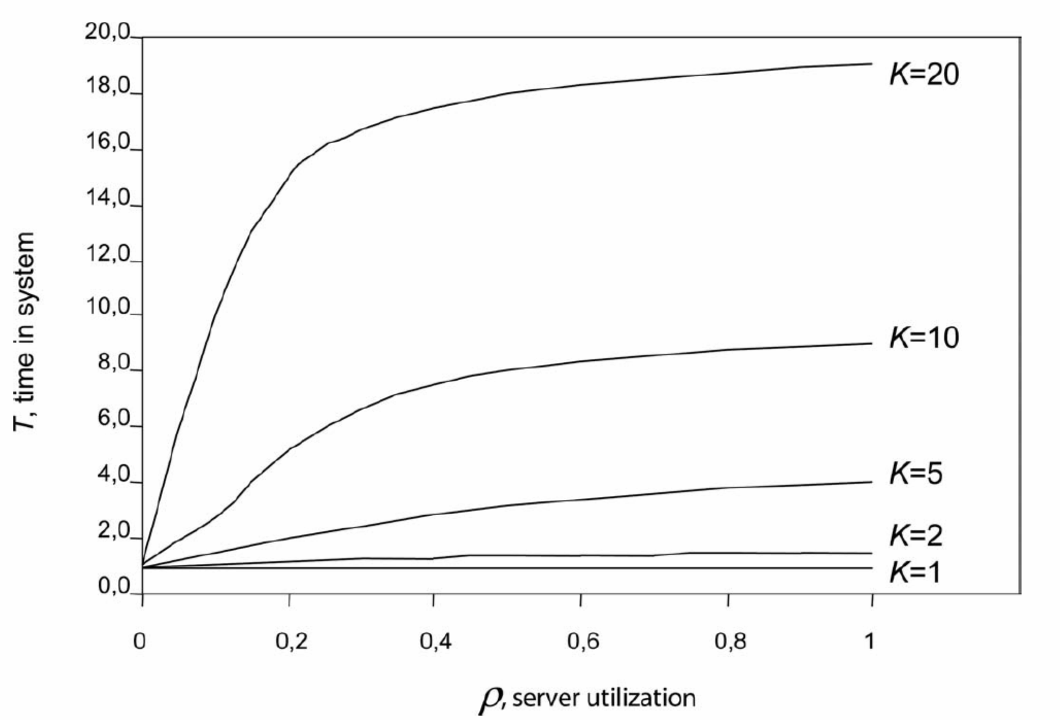


Figure 1

2. The probability to have k customers in system (SCP) with K=20 SSP's.

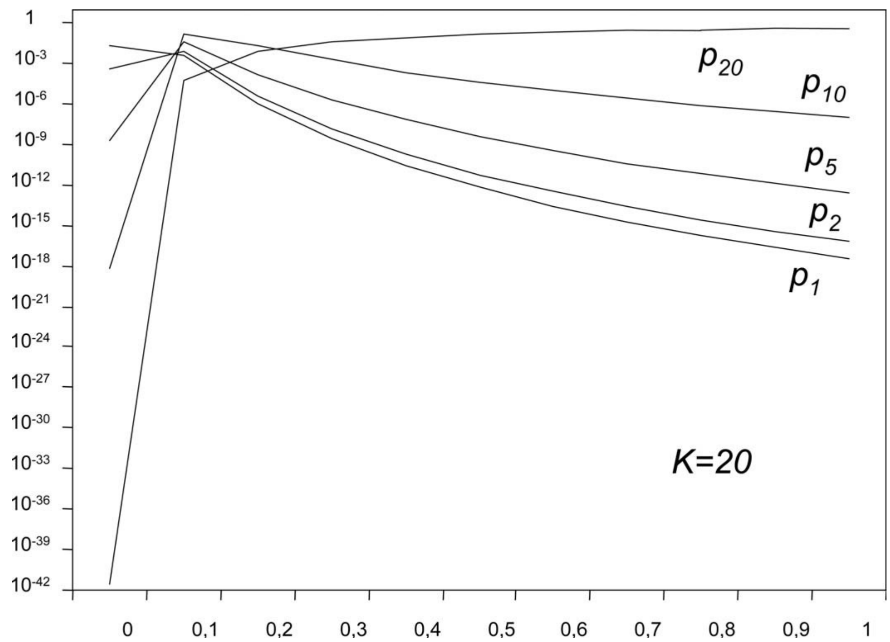


Figure 2

3. The probabilities to have 0 customers in system (SCP) with vary number of SSP's.

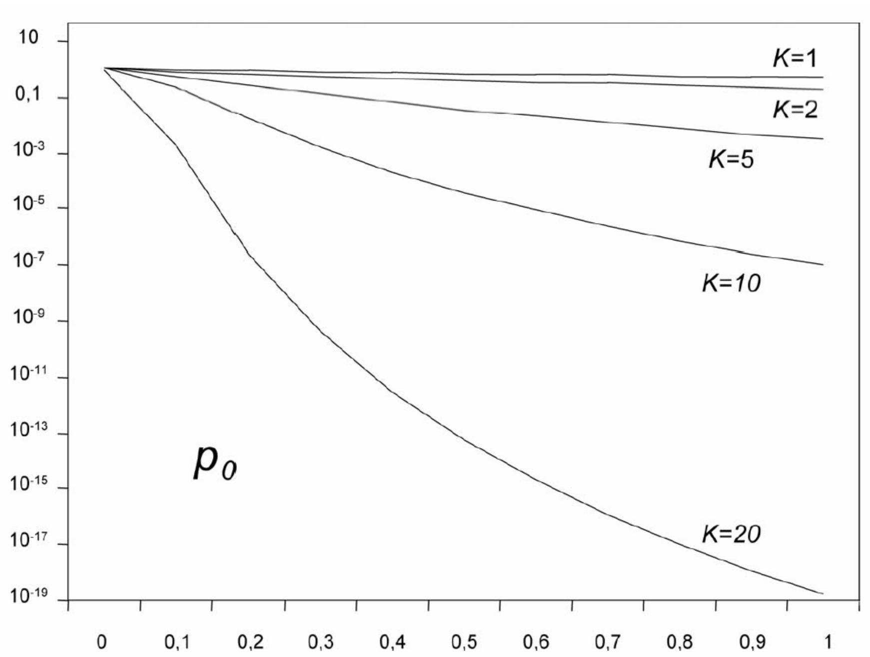


Figure 3

4. The average time is system (SCP) against different service time distributions.

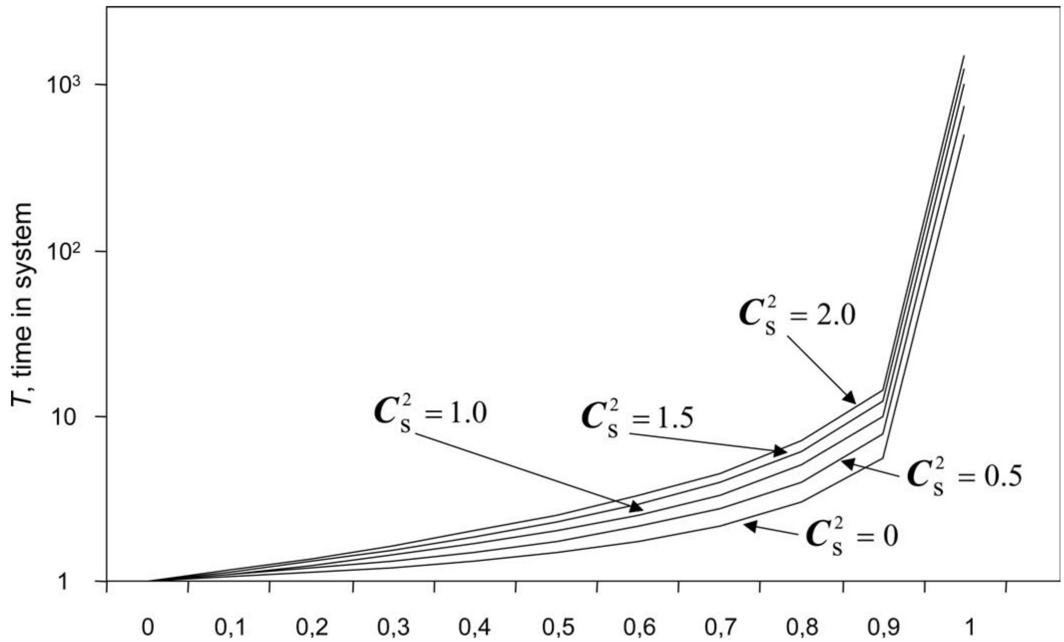


Figure 4

3. APPENDIX

3.1 Statistics gathering methods

1. Average time in system – sum of service times of all customers who got served divided by the total number of served customers.

   Analytic formula:
   $$T = \frac{K/\mu}{1 - p_0} - \frac{1}{\lambda}.$$

2. SCP utilization – sum of the time the server was busy divided by the total simulation time.

   Analytic formula:
   $$p_0 = \left[\sum_{k=0}^{K} \frac{K!}{(K-k)!} \left(\frac{\lambda}{\mu} \right)^k \right]^{-1}.$$

3. Real λ (actual rate of customer flow through the SCP) – total num of served customers divided by the total simulation time.

   Analytic formula:

   Recursion step:

   $$\boxed{\begin{aligned} \bar{T}_i[K] &= (1 + \bar{N}_i[K-1]) \cdot \frac{1}{\mu_i} \\ \bar{N}_i[K] &= K \cdot \frac{\hat{\lambda}_i \cdot \bar{T}_i[K]}{\sum_{j=1}^{M} \hat{\lambda}_j \cdot \bar{T}_j[K]} \\ \lambda_i[K] &= \frac{\bar{N}_i[K]}{\bar{T}_i[K]} \end{aligned}}$$

   In the equation in the middle, the $\hat{\lambda}_i$ are any solution to the flow equations $\lambda_i = \Sigma_j \lambda_j q_{j,i}$.

4. Probability of K customers in system – sum of times K customers were in the system divided by the total simulation time.

   Analytic formula:
   $$p_k = \frac{K!}{(K-k)!} \left(\frac{\lambda}{\mu} \right)^k p_0, \quad 0 \leqslant k \leqslant K,$$