Analytical Modeling and Performance Study of TGREP System

Mayank Singhal and Anirudha Sahoo
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay, Powai, Mumbai, India 400076
Email: {mayanksinghal, sahoo}@cse.iitb.ac.in

Abstract—Interworking of Voice over IP (VoIP) network with traditional Public Switched Telephone Networks (PSTN) is essential for wide adoption of VoIP services. Telephony gateways act as an interface between PSTN and IP telephony networks. These gateways use Telephony Gateway Registration Protocol (TGREP) to advertise their resource information to Location Servers which are responsible for routing calls to appropriate gateway. The periodic interval, known as update interval (UI), at which gateways advertise their resources plays an important role in deciding the performance of TGREP system in terms of call blocking probability. In this paper, we provide an analytical method to model TGREP system. To the best of our knowledge, this is the first study which provides accurate modeling of TGREP system. We validate our model by comparing its characteristics with that of a discrete event simulator developed by us. Using the model, we then analyze the performance of TGREP system as various operating parameters, like update interval, load, call arrival rate and average call length change. This study gives very useful insight into the system. For example, our study shows that a TGREP system should never run above relative load of 1.0 in order to keep the blocking probability below 0.1. Our study also shows that when system is running with light load, update interval can be arbitrarily large. Based on our study, we have provided a reference table which will be useful for TGREP system administrators in setting update interval based on the system configuration and load.

I. INTRODUCTION

With falling high speed Internet prices and availability of technology with QoS guarantees, Voice over IP (VoIP) technology is being deployed widely. Like in Internet services, there are different service providers for VoIP. Thus, the architecture of VoIP has various Internet Telephony Administrative Domains (ITADs), each maintained by a different service provider. As these deployments grow in size, maintaining reachability information while keeping business relationships between different service providers become difficult with static configuration. This has led to design of Telephony Routing over IP (TRIP) which propagates reachability information dynamically across ITADs [1], [2]. In TRIP, there are Location Servers (LSs) which exchange reachability information with LSs in other ITADs.

One of the primary requirements from the VoIP service is backward compatibility with traditional Public Switched Telephone Networks (PSTN). Users of VoIP services should have the flexibility to call PSTN endpoints and vice versa. VoIP networks are packet switched whereas PSTN is circuit switched. This adds a requirement of specialized devices, called Telephony Gateways, that act as an interface between these two very distinct networks. VoIP service providers have deployed many gateways to route calls to PSTN. So, a given call can be successfully completed by many possible candidate gateways. The decision of choosing the appropriate gateway is made by LSs. This choice is based on different set of parameters, the most important of which are:

- **Business Relationships**: While all the gateways can potentially finish a call, some of them may not be considered because of high monetary costs that may be associated with them due to business relationships between the service provider that owns the gateway and the one whose customer has initiated the call.
- **Compatibility**: VoIP calls may involve different signaling protocols and features that the user can choose from. A particular call may require a distinct set of protocols and features to be supported by a gateway. This results in discarding of some of the potential gateways as ineligible candidates for finishing the call due to protocol and feature incompatibility.
- **Gateway Capacity and Available Circuits**: Gateways can only support a limited number of simultaneous calls as they consist of a fixed number of PSTN circuits, which is known as the Gateway Capacity. Some of these circuits may already be serving other calls. A new call can only be accepted if there is a idle circuit. Since there is no queueing at the gateway, calls that are routed to a gateway when it has no free circuits are dropped. Number of available circuits is an important parameter for a call to be successfully completed.

An LS need to choose a gateway based on some or all of the above parameters such that the probability that a call is blocked at the gateway is small.

For better administration, ITADs are further divided into smaller units called Points of Presence (POPs). Each POP consists of an LS and one or more gateways. The LS acts as a proxy for the gateways and communicates with other POPs and ITADs. An LS may have the resources (with the gateways in its POP) to finish the call. Otherwise, the LS routes the call to another POP or another ITAD that has the capability to serve the call. This routing requires information to be exchanged not only between POPs inside ITAD under the same administrative domain but also with other ITADs. Telephony Routing over IP (TRIP) [1] is an IETF Standard that governs this exchange of information.

For an LS to choose an appropriate gateway, it needs to get some information from the gateways in its POP. This gateway to LS information exchange happens using a different, but related protocol called Telephony Gateway REGistration
Protocol (TGREP) [3]. Gateways update the LS using periodic message, called UPDATE message, with latest resource and performance related information. The period of these update messages is called Update Interval. The Update Interval is a configurable parameter that has a significant impact on the performance of the system. One of the primary measures of performance for TGREP systems, which is also used extensively in this study, is the Call Blocking Probability. Call Blocking Probability is defined as the probability that a call in the TGREP system will be dropped, because of it being routed to a gateway with no free circuits available. There has been very little study done on TGREP systems. Hence, the value of Update Interval is normally set to default value preconfigured by manufacturers without any consideration of the system configuration and load. For some system configurations, this may be fine. But for some other this may result in high call blocking probabilities that could otherwise have been avoided if the analysis of the system was available. In this paper, we provide an analytical method to model TGREP systems using known stochastic models in the literature. To the best of our knowledge, this is the first study which provides accurate modeling of TGREP system. We validate our model by comparing its characteristics with that of a discrete event simulator developed by us. Using the model, we then analyze the performance of TGREP system (in terms of blocking probability) as various operating parameters, like update interval, load, call arrival and average call length change. This study gives very useful insight into the system. For example, our study shows that a TGREP system should never run above relative load1 of 1.0, otherwise the call blocking probability will be quite high2. Our study also shows that when system is running with light load, update interval can be arbitrarily large, i.e., updates can be completely stopped. This finding opens up scope for adaptive update interval where update intervals can be changed suitably as per the system load. Based on our study, we have provided a reference table which, we hope, will be useful to TGREP system administrators in setting update interval based on the system configuration and load. The main contributions of this paper are:

- There is not much literature on performance study of TGREP system. To the best of our knowledge this is the first study which presents an analytical model that captures the working of TGREP system accurately.
- The current industry practice is to set update interval to its default value. But this may not be always a suitable setting. We have carried out extensive simulation experiments to get insight into how update interval impacts the performance of TGREP system in terms of call blocking probability, which would be helpful in deciding the value of update interval for a given configuration.
- Based on our study, we provide a reference table of update interval for different configuration of TGREP systems which would be useful for TGREP system administrators.

The rest of the paper is organized as follows. Section II presents an overview of TGREP system, Section III outlines the related work in this area of research. Section IV has the detailed analytical modeling of TGREP system followed by simulation experiment setup and results in Section V. We conclude the paper in Section VI with some of the important findings from this study.

II. Overview of Operation of TGREP System

As mentioned before, VoIP service architecture is divided into various ITADs each governed by a separate administrative authority. The ITADs are further divided into POPs which contain one or more Location Servers and Gateways. Location Servers are responsible for routing calls whereas gateways act as interfaces between PSTN and IP networks. Communication between different Location Servers happens through TRIP protocol, which has two flavors. E-TRIP is used for intra-domain information exchange whereas I-TRIP is meant for inter-domain information exchange [1]. Most of this information originates from the gateways. The call routing is governed by various signaling protocols that are beyond the scope of this study. Figure 1 shows the architecture of a typical VoIP deployment.

When a VoIP user makes a call to connect to a PSTN endpoint, the call is routed to a Location Server that has the capability of finishing the call through one of gateways in its POP. The LS should select a gateway such that the Call Blocking Probability is minimum. There have been few gateway selection algorithms suggested which can be found in [4]. But to select an appropriate gateway, the LS needs to know the state of the gateways. This information is communicated to LS by the gateways using TGREP. Location Server can be logically divided into two parts: the Egress LS and the Ingress LS. Egress LS participates in information exchange with other location servers using TRIP whereas Ingress LS acts as TGREP Receiver. Gateways act as TGREP Senders. TGREP standard defines few messages for its operation out of which UPDATE message is of importance to our study. Gateways use UPDATE message to send their state information to LS. This message consists of several attributes, the most important of which are as follows.

- **TotalCircuitCapacity:** This represents the total number of PSTN circuits on the gateway. This attribute is fairly static; it can change when trunks are adds to or removes from the gateway.

1Relative load is defined later in the paper.
2We assume that call blocking probability above 0.1 is not acceptable in practical systems.
• **AvailableCircuits**: This is the number of free circuits at a given time. As calls are established and disconnected, this attribute changes.

• **CallSuccess**: contains number of calls succeeded and total number of calls attempted in an interval.

• **Prefix**: Prefix list of destination numbers the gateway can complete calls to.

• **TrunkGroup**: List of trunk groups on the gateway used to complete calls.

• **Carrier**: List of carriers that the gateway uses to complete calls.

In this paper, we will only be concentrating on **TotalCircuit-Capacity** and **AvailableCircuits**. The **UPDATE** messages are not sent instantaneously as the gateway state changes. This is because constructing and transmitting them has overhead on the gateways. Hence, the **UPDATE** messages are sent periodically. A large update interval is good for reducing gateway overhead, but may lead to worse performance because the system may improve performance at the cost of incurring larger overhead. Thus, the choice of update interval is very important for system performance.

### III. RELATED WORK

Internet Standards for TGREP and TRIP are fairly recent. For this reason there is very little prior work on their modeling and performance analysis. In [5], performance analysis of TRIP was done using simulations. This work is related closely to the problem that this study aims to address. The study was done prior to the introduction of TGREP standard, when TRIP was used as the protocol of communication between gateways and location server as well. Some of the analysis performed in the study will now be applicable to TGREP instead. Among the various experiments conducted in [5], analysis of impact of propagation delays on call blocking probability extends easily to TGREP. Propagation delays resulting due to calls being routed from one LS to another as well as due to routing from LS to gateway and their impact on Call Blocking Probability were studied. It was noted that propagation delay does not impact overall call blocking probability but it does impact the location of call blocks, when there is propagation delay from LS to gateway. On increasing delay to the gateway, probability of call blocking on gateway increases and the probability of call blocking on the LS decreases. Other experiment involved analysis of call rerouting between different location servers, comparison of TRIP enabled network to SIP networks and effect of trunk failures.

There have been very limited studies done on the modeling and performance analysis of TGREP systems. In [6] authors have studied TGREP and proposed strategy to minimize the delay in setup of the call. It is argued that high delays will result in lower revenues as the VoIP user may get impatient and terminate the call. **Forking** is suggested as a possible solution to decrease the setup time, in which multiple gateways compete against each other to complete the call. This may result in higher load in the system as all except one of the competing gateways will finally serve the call.

In [4] and [7], the authors did a comparative study of various deterministic as well as probabilistic gateway selection algorithms. It was reported that **Probabilistic Selection Based on Absolute Available Circuits (PSBAAC)** performance is the best among the ones that were analyzed. In [4], **M/M/T/T** queuing abstraction was proposed for modeling gateways which was utilized later in [8]. The goals of [8] was to provide an analytical model for TGREP system and use the proposed model for finding optimal operating value of update interval. it used only steady state analysis of CTMCs for abstracting the system. But TGREP system cannot be analyzed by only using steady state analysis. The periodic updates require that the system be analyzed in a finite duration, i.e., the transient behavior of the system has to be modeled. In this study, we accurately model TGREP system by abstracting it as a DTMC across update interval and as a CTMC within an update interval. Within update interval, transient behavior of the CTMC is modeled to accurately capture the behavior of the system.

### IV. ANALYTICAL MODELING OF TGREP SYSTEM

#### A. System Model

We model TGREP system as consisting of two gateways \( G_1 \) and \( G_2 \) of total capacities \( T_1 \) and \( T_2 \) respectively\(^3\). Thus, the two gateways can successfully complete maximum of \( T_1 \) and \( T_2 \) calls respectively. Arrival of calls to the location server is Poisson distributed with rate \( \lambda \). Call length is exponentially distributed with mean \( 1/\mu \). Update interval is assumed to be the same for both the gateways. Additionally, it is assumed that the LS receives \( \text{UPDATE} \) messages from the gateways at the same time. We represent number of busy circuits (i.e., number of ongoing calls) on gateway \( G_1 \) and \( G_2 \) by \( i_1 \) and \( i_2 \) respectively. When calls arrive to LS, the LS uses a gateway selection algorithm to direct the call to a particular gateway.

Our model is agnostic to any gateway selection algorithm and hence provides flexibility to use any gateway selection algorithm.

We denote update interval as \( U \) which is periodic. We denote the states of the two gateways as an ordered pair \((i_1, i_2)\). Let \( P_a(i_1, i_2, U) \) denote the probability that the gateways are in states \((i_1, i_2)\) as per the update received by the LS. Let \( P_b(i_1, i_2, U) \) be the call blocking probability when the last known gateway states (at the LS) were \((i_1, i_2)\) in the current update interval.

We list Various notations used in this analysis in Table I for ease of reference.

#### B. Modeling Methodology

We split the study of the TGREP system in two parts to make it easier to model. The first part studies the system across update intervals whereas the second part studies the system within an update intervals.

The gateway states transition from \((i_1, i_2)\) to say, \((i'_1, i'_2)\) across update intervals which is modeled as a DTMC. Within an update interval, the states of the gateways change as calls arrive or depart. This is modeled as a CTMC.

After each update, dispatching of calls to the gateways are done solely based on the current state information (by

---

\(^3\)We present our model with two gateways to keep the analysis simple. Our model can easily be extended to larger number of gateways.
the Gateway Selection Algorithm). Hence, given an update to the LS, the system state in future is independent of any of the previous updates and is only dependent on the state information carried in the current update. Thus, the gateway state transitions across update interval can be modeled as a DTMC.

Arrival to LS is assumed to be Poisson. The LS directs the arrivals to the two gateways based on the last state information. Thus, the individual arrival to the gateways is also Poisson. The call duration is exponentially distributed. Hence, the state transitions, which happen due to call arrival and departure, within an update interval is Markovian. Thus, the state transitions within an update interval can be modeled as a CTMC. These two models are depicted in Figure 2.

Using the Law of Total Probability, the Call Blocking Probability of the system, $P^b_{total}$, for a given update interval $U$ can be calculated as:

$$P^b_{total}(U) = \sum_{i_1=0}^{T_1} \sum_{i_2=0}^{T_2} P_u(i_1, i_2, U) P_b(i_1, i_2, U)$$  \hspace{1cm} (1)

Next, we compute each term in the right side of Equation (1) to get the total blocking probability.

C. Computation of $P_u(i_1, i_2, U)$

As described earlier, $P_b(i_1, i_2, U)$ is the probability that of gateways reporting state $(i_1, i_2)$ when update interval is $U$. So this is the steady state probability of an update reporting the gateways to be in state $(i_1, i_2)$. To perform steady state analysis of DTMC, it is required to find the transition probability matrix for this DTMC. The methodology to find the transition probability $C_{vw}(U)$ is described in Section IV-C2.

Let us assume that the successive time steps of DTMC is given by $\{X_n : n \in \mathbb{N}\}$. The DTMC has state space having $(T_1+1)(T_2+1)$ states and its transition probabilities are given by:

$$P\{X_{n+1} = w | X_n = v\} = C_{vw}(U)$$  \hspace{1cm} (2)

Note that the transition probabilities are dependent on the value of update interval $U$ since it dictates what would be the system states at the end of the update interval.

1) Steady State Analysis of DTMC: The states of the DTMC are represented by an ordered pair $(i_1, i_2)$ representing the number of busy circuits reported in the last updates from gateway $G_1$ and $G_2$ respectively. Hence, the state space of the DTMC contains $(T_1+1)(T_2+1)$ states from $(0, 0)$ to $(T_1, T_2)$. To represent states in a single dimension we use index $k$. The conversion from one dimension to two dimensions is easily done by equations:

$$i_1 = \lfloor k/(T_1 + 1) \rfloor$$  \hspace{1cm} (3)

$$i_2 = k \mod (T_2 + 1)$$  \hspace{1cm} (4)

We use the conventional steady state analysis [9] to compute the limiting probability vector $\vec{P}$ using equation:

$$\vec{P} = \vec{P} \mathbf{C}(U)$$  \hspace{1cm} (5)

$$\Pi_k \geq 0$$  \hspace{1cm} (6)

$$\sum_{k=0}^{(T_1+1)(T_2+1)-1} \Pi_k = 1$$  \hspace{1cm} (7)

Where $\Pi_k$ denotes the limiting probability of $k^{th}$ state (in the vector $\vec{P}$) of the DTMC. $\mathbf{C}(U)$ is the transition probability matrix of the DTMC.

2) Computation of Transition Probabilities of the DTMC:

To calculate $\mathbf{C}(U)$, we need to analyze the behavior of the system within an update interval. We can calculate the elements of the matrix in a row wise fashion. Each row $v$ of the matrix $\mathbf{C}(U)$ represents one step transition probabilities from state $v$ to all states $w \in \{0, 1, 2, ... \}$. The row corresponding to state $v$ has the index $k$ that $v$ is corresponding to single dimension for our analysis.

As mentioned before, the behavior of the gateways is independent of each other (during an update interval) after the arrival rates are set by the Gateway Selection Algorithm. We consider a transition step of the DTMC from state $(i_1, i_2)$ to $(i'_1, i'_2)$. Thus, the starting state of Gateway $G_j$, $j \in 1, 2$ is given by $i_j$. Now, using transient analysis we can calculate the probability distribution vector of different gateway states. The size of this vector is $(T_j + 1)$ for the gateway $G_j$. The $w^{th}$ element of the $v^{th}$ row of transition matrix $\mathbf{C}(U)$ can be calculated as the product of probabilities of the two gateways being in the states which corresponds to this element of $\mathbf{C}(U)$ as given in Equation (8).

$$C_{vw}(U) = P^i_{1v}(U)[i'_1] P^i_{2w}(U)[i'_2]$$  \hspace{1cm} (8)

$P^i_{jv}(U)$ is the transition probability vector of gateway $G_j$ with starting state $i_j$ when update interval is $U$. $P^i_{jv}(U)[i'_j]$ represents the $i'_j^{th}$ element of the vector. Note that in the above equation, $v$ and $w$ correspond to single dimension equivalent
of \((i_1, i_2)\) and \((i_1', i_2')\) respectively (which can be computed using Equations (3) and (4)). To compute \(P_{ij}^>(U)\) we use uniformization method of transient analysis of CTMC. We use the theory directly from [10] and adapt it to our notations. The transition state probability vector \(P_{ij}^>(U)\) is given by [10]

\[
P_{ij}^>(U) = \sum_{i=0}^{\infty} \Pi_j^>(i) e^{-q_{ij}^U} (q_{ij}^U)^i \frac{i^!}{i!}.
\] (9)

The scalar quantity \(q_{ij}^U\) is calculated from the Generator Matrix \(Q_j\) of corresponding to Gateway \(G_j\) starting at state \(i_j\) as follows [10].

\[
Q_j^i = \{q_{ij}^m(m,n)\}
\] (10)

\[
q_{ij}^U \geq \max_{0 \leq i \leq T_j} \left| q_{ij}^U (i,i) \right|
\] (11)

The Generator Matrix \(Q_j^i\) is given by [10]:

\[
Q_j^i = \begin{bmatrix}
-\lambda_j & \lambda_j & 0 & 0 & \ldots & 0 & 0 \\
\mu & -(\mu + \lambda_j) & \lambda_j & 0 & \ldots & 0 & 0 \\
0 & 0 & \ldots & 0 & 0 & \ldots & T_j \mu \\
\end{bmatrix}
\] (12)

\(q_{ij}^U\) is also required to calculate the transition probability matrix \(Q_{ij}^U\) used for Uniformization of the CTMC as given below:

\[
Q_{ij}^U = \frac{Q_j^i}{q_{ij}^U} + I
\] (13)

Here, \(I\) is a \((T_j + 1) \times (T_j + 1)\) Identity matrix. Now the row vector \(\Pi_j^>(i)\) used in Equation (9) can be calculated iteratively as shown below [10].

\[
\Pi_j^>(0) = P_{ij}^>(0)
\] (14)

\[
\Pi_j^>(i) = \Pi_j^>(i-1)Q_{ij}^U
\] (15)

where \(P_{ij}^>(0)\) is the initial probability vector, which is determined from the starting state of the gateway at the beginning of Update Interval. The elements of the initial probability vector are as given below.

\[
P_{ij}^>(0) = \{p_{ij}^j(m)\}
\] (16)

\[
p_{ij}^j(m) = \begin{cases} 
1 & \text{if } m = i_j \\
0 & \text{otherwise}
\end{cases}
\] (17)

The infinite sum in Equation (9) can be truncated on both ends with lower limit \(l\) and upper limit \(k\) with error bound \(\varepsilon\) [10]. The lower limit \(l\) is found such that it is the largest non-negative integer satisfying the following inequality [10].

\[
e^{-q_{ij}^U} \sum_{i=0}^{l-1} (q_{ij}^U)^i \frac{i^!}{i!} \leq \frac{\varepsilon}{2}
\] (18)

Similarly \(k\) is the smallest non-negative integer such that [10]

\[1 - e^{-q_{ij}^U} \sum_{i=0}^{k-1} (q_{ij}^U)^i \frac{i^!}{i!} \leq \frac{\varepsilon}{2} \text{ if } k \geq 0
\] (19)

Now \(u^{th}\) element of the \(v^{th}\) row of Transition Matrix \(C(U)\) can be calculated using Equation (8).

D. Calculation of Call Blocking Probability \(P_b(i_1, i_2, U)\)

To calculate \(P_b(i_1, i_2, U)\), we need to understand the system behavior within an update interval. Arrival and departure of calls to the gateways within an update interval change the state of the gateway. Thus, within an update interval the system can be modeled as a CTMC (see Figure 2). A call arriving to the LS is directed to one of the two gateways based on the gateway selection algorithm. Since the arrival to LS is Poisson, the fraction of arrival to individual gateways are also Poisson. The behavior of gateway between two updates is independent of each other as state of one of these gateways cannot affect the other gateway. Thus, the two gateways (their CTMC models) can be analyzed independently. Note that in the case of analysis of DTMC the states of gateways are interdependent because transition of gateway, say \(G_1\), from state \(i_1\) to state \(i_1'\) (at the end of the update interval) is dependent on what was the arrival rate to \(G_1\), which in turn, depends on the state \(i_2\) of gateway \(G_2\).
Analysis of CTMC of individual gateways has to be done in a duration \( U \), after which the CTMC parameters may change because the next update may report different gateways states which, in turn, would change arrival rate to the gateway. Hence, transient behavior of CTMC should be studied in an interval of \( U \).

Using the law of total probability, call blocking probability for the system is the sum of call blocking probabilities at the individual gateways if the call is routed to that particular gateway. Hence, \( P_b(i_1, i_2, U) \) is given by

\[
P_b(i_1, i_2, U) = p_1(i_1, i_2)P_{b_1}(i_1, U, \lambda_1) + p_2(i_1, i_2)P_{b_2}(i_2, U, \lambda_2). \tag{20}
\]

Where \( p_j(i_1, i_2) \) is the gateway selection probability of \( j \)th gateway when the last known state was \( (i_1, i_2) \). \( P_{b_j}(i_j, U, \lambda_j) \) is the Call Blocking Probability for the \( j \)th gateway starting with a state \( i_j \) in time interval \( U \) having arrival rate \( \lambda_j \).

1) **Gateway Selection Probabilities**: In this study, we have used Probabilistic Selection Based on Absolute Available Circuits (PSBAAC) as the gateway selection algorithm [4].

Hence gateway selection probabilities are given by [4]:

\[
p_j(i_1, i_2) = \begin{cases} 
\frac{1}{2} \, & \text{if } i_1 = T_1 \text{ and } i_2 = T_2 \\
\frac{1-i_1/T_1}{1-i_2/T_2} & \text{otherwise}
\end{cases} \tag{21}
\]

which can be used in Equation(20).

Call arrival rates \( \lambda_j \) to gateway \( G_j \) when can be calculated as:

\[
\lambda_j = p_j(i_1, i_2)\lambda \tag{22}
\]

Note that \( \lambda_j \) should have been denoted as \( \lambda_j(i_1, i_2) \), because it can potentially change after every update interval (which reports the states as \( (i_1, i_2) \)). But we leave out \( (i_1, i_2) \) to keep the notation simple.

2) **Transient Analysis of CTMC**: Gateway \( G_j \) is modeled as a \( M/M/T_j/T_k \) queue. The arrival to the gateway is Poisson distributed with rate \( \lambda_j \) given by (22). Average call holding time is \( 1/\mu \). The corresponding CTMC has \( T_j + 1 \) states. The transition rates for the CTMC are shown in the Figure 3.

![Logical Gateway](image)

Fig. 3. Gateway as \( M/M/T_j/T_k \) Queue

We use uniformization for the transient analysis of the CTMC. The method is adapted from [11] and [10] which is based on reduction of the CTMC into a DTMC subordinated to a Poisson process. To compute the probability that a call is blocked on arrival to a gateway, we use the property of Poison Arrival called Poisson Arrival Sees Time Average (PASTA) [12]. Due to this property, the probability of a call arriving to a gateway being blocked in duration \( U \) is equal to the fraction of time gateway stays in the state of all circuits being busy. To compute the fraction of time gateway stays in the state of all circuits being busy, we need to first find the duration for which the gateway stays in that state (starting from some last known state). Using our notations, for gateway \( G_j \) this is the duration for which the gateways stay in the state \( T_j \) in an interval \( U \), given the starting state was \( i_j \). This is nothing but the **Occupancy Time** of the gateway in the state \( T_j \) in an interval \( U \), given the starting state was \( i_j \) and is denoted by \( m_{i_j, T_j}(U) \). This is the \( (i_j, T_j) \)th element of Occupancy Matrix \( M^U_{i_j}(U) \). We use the method presented in [11] to compute this occupancy time. We present the method directly from [11] (adapted to our notations).

\[
M^U_{i_j}(U) = m_{v,w}(U) \quad \text{for } v, w \in \{0, 1, 2, ..., T_j\} \tag{23}
\]

\[
M^U_{i_j}(U) \text{ can be calculated using Uniformization [11] as }
\]

\[
M^U_{i_j}(U) = \frac{1}{r^U_{i_j}} \sum_{k=0}^{\infty} P(Y^r_{i_j} > k)\hat{P}^k_{i_j} \tag{24}
\]

Where:

\[
Y^r_{i_j} \sim \text{Poisson}(r^r_{i_j} U) \tag{25}
\]

\[
P^r_{i_j} = \{p^r_{i_j}(v, w)\} \tag{26}
\]

\[
p^r_{i_j}(v, w) = \begin{cases} 
1 - \frac{r^r_{i_j}(v)}{r^r_{i_j}(v)} & \text{if } v = w \\
\frac{r^r_{i_j}(v)}{r^r_{i_j}(v)} & \text{if } v \neq w.
\end{cases} \tag{27}
\]

\[
r^r_{i_j}(v) = \max_{0 \leq v \leq T_j} \left( r^r_{i_j}(v) \right) \tag{28}
\]

\[
r^r_{i_j}(v) = \sum_{w=0}^{T_k} r^r_{i_j}(v, w) \tag{29}
\]

Note that \( r^r_{i_j}(v, w) \) is the \( (v, w) \)th element of the Rate Matrix \( R^r_{i_j} \). \( R^r_{i_j} \) can be obtained from the Generator Matrix \( Q^r_{i_j} \) as follows [11]

\[
r^r_{i_j}(v, w) = \begin{cases} 
q^r_{i_j}(v, w) & \text{if } v \neq w \\
0 & \text{otherwise.}
\end{cases} \tag{30}
\]

Generator Matrix \( Q^r_{i_j} \) is given by Equation (12). So \( m_{i_j, T_j}(U) \) can be used to calculate \( P_{b_j}(i_j, U, \lambda_j) \) as follows.

\[
P_{b_j}(i_j, U, \lambda_j) = \frac{m_{i_j, T_j}(U)}{U} \tag{31}
\]

Using this value of \( P_{b_j}(i_j, U, \lambda_j) \), \( P_b(i_1, i_2, U) \) can be computed using Equation (20). Finally, computation of total call blocking probability can be done using Equation (1).

V. EXPERIMENTS AND RESULTS

In this section, we present details of our experiments and the corresponding results. We have implemented the analytical modeling presented in the previous section. We also have built a discrete event simulator to study TGREP system. The simulator also helps us to validate the analytical model developed for TGREP.
A. Implementation of Analytical Model and Development of Simulator

We have implemented the analytical model of TGREP system presented in the previous section using Java. This analytical modeler helps us study how various operating parameters affect the performance of TGREP system. We also have built a discrete event simulator using Java. Java libraries JAMA [13], MTJ [14] and EJML [15] were used for various matrix operations and jMarkov [16] was used for quickly abstracting Markov Chains in analytical modeling.

All simulations were run for a duration such that one hundred thousand calls arrive at the Location Server. The average call length was set to 180 seconds. The number of gateways, their capacities and gateway Selection Algorithm are configurable. But in this paper we have used PSBAAC as the gateway selection algorithm in all the experiments.

In traditional telephony system, load on the system is measured in erlang. But for our purpose, erlang load may not give the correct picture for comparison. Hence we introduce a related metric called Relative Load (for the system) which is defined as follows.

\[
\text{Relative Load} = \frac{\text{Erlang Load}}{\text{Total Number Of Circuits}}
\]  \hspace{0.5cm} (32)

where Erlang Load is given by \( \frac{\lambda}{\mu b} \) and Total Number of Circuits is given by \((T_1 + T_2)\). Thus, Relative Load can be used to compare (fairly) systems of different sizes.

B. Configurations of TGREP System

We have considered various configurations for our experiments as explained in this section.

1) Symmetrical Systems: First, we consider the two gateways to be symmetrical with respect to their capacities. For this configuration, we have considered three different system configurations.

- **Low End System** For the low end system we consider a symmetrical system where each of the gateways has just 23 circuits (one T1 trunk).

- **Medium Range System** For this configuration we consider a symmetrical system where each of the gateways has 46 circuits (two T1 trunks).

- **High End System** Here the gateways consist of 92 circuits each (four T1 trunks).

2) Asymmetrical Systems: The number of circuits on the gateways may not always be equal in practice. Hence, we run our experiments for asymmetrical system in which one of the gateways has twice the number of circuits than the other.

- **Low - Med Asymmetrical System** This system consists of one gateway with forty six circuits and the other gateway with only twenty three circuits.

- **Med - High Asymmetrical System** In this configuration, the two gateways have forty six and ninety two circuits respectively.

C. Impact of Update Interval on Call Blocking Probability

The first experiment we conducted was to study the impact of Update Interval (UI) on call blocking probability \( (P_b^{total}) \). The results (both analytical modeling and simulation based) for the three symmetrical configurations are shown in Figure 4, Figure 5 and Figure 6 respectively. Analytical and simulation based values match very closely and hence the two curves (for a given relative load) cannot be distinguished from each other.

It can be seen from these figures that the performance graphs of the three symmetrical configurations have very similar shape. For a given UI, the call blocking probability, as expected, is lower for TGREP system with higher total capacity. We make some interesting observations from these graphs. For relative load of 0.6 or lower, the call blocking probability is almost zero and does not depend on UI. This indicates that when TGREP system is running at a very low load, then gateways can stop sending updates to the LS without having an adverse impact on the performance. These graphs can be used to choose operating UI of a gateway, when the expected relative load to the system is known, to keep the call blocking probability below a certain value.

The results for asymmetrical system are shown in Figure 7 and 8. The behavior of the system resembles closely to that in the symmetrical case.

D. \( P_b^{total} \) vs Relative Load

We then study the impact of relative load on blocking probability. Since our analytical model matches very closely with simulation, we do not run experiments for both. We only carry out experiment with our analytical modeler.

For symmetrical configuration, we ran experiments only for low and medium end systems. We could not run the experiments for high end system because of its high memory requirement. The results for low and medium end symmetric system are shown in Figure 9 and 10 respectively. These curves can also be used to find operating parameter for the system. For example, looking at Figure 9, it is clear that the system should not be run beyond relative load of 1, to keep blocking probability below 0.1. We assume that practical systems would actually have blocking probability much less than 0.1.

Earlier studies [4] had shown that the \( P_b^{total} \) vs Load resembles an S-curve. The results in [4] were obtained from
Fig. 5. $P_b^t$ vs $U$ for a Medium End Symmetrical System

Fig. 6. $P_b^t$ vs $U$ for High End Symmetrical System

Fig. 7. $P_b^t$ vs $U$ for a Low-Med Asymmetrical System

Fig. 8. $P_b^t$ vs $U$ for a Med-High Asymmetrical System

Fig. 9. $P_b^t$ vs Relative Load for a Low End Symmetrical System

Fig. 10. $P_b^t$ vs Relative Load for a Med Range Symmetrical System
simulation. Our experiment shows the same performance curve can be obtained from our analytical modeler.

Results for asymmetrical system is shown in Figure 11. We show the result for Low-Med asymmetrical system. The performance of TGREP system is very similar in this configuration.

E. Dependence of Blocking Probability on $\lambda$ & $\mu$

In traditional telephony systems, the call blocking probability depends on the erlang load (e.g., Erlang B formula). When the call arrival rate ($\lambda$) and call holding time ($1/\mu$) are changed such that the load does not change (this can be done by scaling both $\lambda$ and $\mu$ by the same factor), the blocking probability does not change. We wanted to test the same property for TGREP system.

We carried out experiments for low and medium end symmetrical systems and for low-mid range asymmetrical systems. The relative load is kept constant at 0.75. But $\lambda$ and $\mu$ were multiplied by a common scaling factor such that the relative load remained at 0.75. In another words, the ratio $\lambda/\mu$ did not change.

Figures 12 and 13 show the results for low and medium end symmetrical system. Both the graphs have similar behavior. In traditional telephony system, for different scaling factors there would be one curve (since erlang load does not change). But for TGREP system, we see that the blocking probability changes as the scaling factor change (for a given UI). These set of graphs show TGREP system does not behave like a traditional Markovian queuing system. For traditional $M/M/T/T$ queue call blocking probability is dependent just on the load and can be calculated using Erlang B formula. But in TGREP system the call blocking probability is dependent on individual value of call arrival rate and average call length.

Figure 14 shows the corresponding graph for low-mid range asymmetrical system, which has very similar behavior as symmetrical system.

F. Reference Table for TGREP System Administrators

In practice, system administrators use a default value of update interval (25 seconds as per [17]) for TGREP deployments regardless of the configuration of the system. In our experiments we have shown that call blocking probability is dependent on update interval and that the default value may not always give the desired performance. Here, we provide two tables which can be used by TGREP system administrators to set the UI values depending on the system configuration. The tables give an update interval value which will result in desired call blocking probability for a given system size and relative load. These $U$ values were obtained using Brack- etingNthOrderBrentSolver, a root finding algorithm available in [18] which is a modification of Brent’s Method [19]. Only the update intervals that are in the range (1, 300) have been reported. Table II gives the update interval values for expected blocking probability of 0.05, whereas Table III gives the corresponding values for expected blocking probability of 0.1.

VI. CONCLUSION

In this paper, we have presented accurate analytical modeling of TGREP system which is abstracted as a DTMC across update interval and CTMC within an update interval. We used transient behavior of CTMC and steady state behavior of DTMC to compute call blocking probability. We analyzed the system performance as various operating parameters changed. This study provides many helpful insight into the system. Some of the important observations from the study are (1) TGREP should not be run beyond relative load of 1.0 in a
two gateway system each having upto 92 circuits and (2) At low relative load (below 0.6), update interval does not make any difference in the performance and hence updates can be stopped completely.

Based on our study, we have provided a reference table of update interval values for different configurations which can be used by TGREP system administrators to run the system at a good performance level.

**REFERENCES**


---

**TABLE III**

<table>
<thead>
<tr>
<th>RL</th>
<th>(T₁, T₂)</th>
<th>(23, 23)</th>
<th>(23, 46)</th>
<th>(46, 46)</th>
<th>(46, 92)</th>
<th>(92, 92)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.60</td>
<td></td>
<td>&gt; 300</td>
<td>&gt; 300</td>
<td>&gt; 300</td>
<td>&gt; 300</td>
<td>&gt; 300</td>
</tr>
<tr>
<td>0.65</td>
<td></td>
<td>&gt; 300</td>
<td>&gt; 300</td>
<td>&gt; 300</td>
<td>&gt; 300</td>
<td>&gt; 300</td>
</tr>
<tr>
<td>0.70</td>
<td></td>
<td>&gt; 300</td>
<td>&gt; 300</td>
<td>&gt; 300</td>
<td>&gt; 300</td>
<td>&gt; 300</td>
</tr>
<tr>
<td>0.75</td>
<td></td>
<td>214.35</td>
<td>228.27</td>
<td>223.72</td>
<td>227.98</td>
<td>224.87</td>
</tr>
<tr>
<td>0.80</td>
<td></td>
<td>155.36</td>
<td>170.16</td>
<td>169.86</td>
<td>173.93</td>
<td>170.70</td>
</tr>
<tr>
<td>0.85</td>
<td></td>
<td>105.55</td>
<td>123.06</td>
<td>127.38</td>
<td>131.66</td>
<td>129.74</td>
</tr>
<tr>
<td>0.90</td>
<td></td>
<td>59.88</td>
<td>81.62</td>
<td>90.85</td>
<td>96.00</td>
<td>90.03</td>
</tr>
<tr>
<td>0.95</td>
<td></td>
<td>18.60</td>
<td>43.18</td>
<td>56.73</td>
<td>64.07</td>
<td>66.77</td>
</tr>
<tr>
<td>1.00</td>
<td></td>
<td>&lt; 1</td>
<td>8.39</td>
<td>22.05</td>
<td>33.72</td>
<td>39.61</td>
</tr>
</tbody>
</table>