VoIPTG - v1.0.0

VoIP Traffic Generator

Contents

Chapter 1

Introduction

1.1 VoIPTG - VoIP Traffic Generator

VoIP Traffic Generator is a synthetic generator capable to generate traces close to real world VoIP traffic. For convenience, we call the Traffic Generator VoIPTG in this document.

1.2 Why Synthetic Generator

IP Telephony has become very popular and SIP-based telephony systems are almost substituting the traditional PSTN systems by being the backbones of major operators. Lower call price, easier automation and faster calling procedures make the SIP based telephony most popular but at the same time it introduces threat, similar to that of email spam known as Spam over Internet Telephony (SPIT). SIP based VoIP traffic does not follow any generic model to describe its characteristics like traditional telephony. Analysis on VoIP (Voice over Internet Protocol) traces shows the difference in the characteristics of VoIP traffic from the traditional POTS and ISDN. Difference in characteristics also means different parameters to tune the network monitoring tools. Moreover, a key prerequisite for the development of effective SPIT prevention systems is an adequate knowledge of VoIP traffic pattern. Again until now, SPIT attacks are only potential and in the literature are still not documented. In this stage, the only way to measure effectiveness of the SPIT prevention system is to use a simulative approach.

This state introduces the necessity for a new consideration on the SIP based VoIP traffic characteristics. Again a new analysis tools has become most important for the management of SIP based telephony systems as well as for the prevention of SPIT. Available traffic generators fail to capture the complex statistical properties of SIP based VoIP Traffic. Explicitly, they fail to characterize the more sophisticated traffic parameters addressing the user behavior. Characterizing the overall traffic volume in terms of interarrival time and call duration is a simple task as it ignores the complexity of users behavior. But modeling the user perspective requires additional architectural decisions for the stochastic generator.

Considering this scenario, we have developed synthetic generator VoIPTG with characterization of users capable to give a solid idea of the system behavior for tuning of VoIP traffic control and anti-SPIT systems.

Chapter 2

VoIPTG Architechture

2.0.1 VoIPTG Flow

Based on analysis on real trace, synthetic traffic generator VoIPTG is generated. The logical flow of VoIPTG is found in fig 2.1.

The input of VoIPTG is the configuration file containing the input parameters of the traffic generation and the output is in form of standard CDRs which can be read by any call monitoring system. The generation of a trace begins with the automatic generation of user domain with users identifiers. Only identifiers are generated, not instances of a user entity, so that configuration of users characteristics is not required. User domain includes Caller Set with N_u callers and Receiver set with N_r receivers. In the present version of VoIPTG, mainly focuses for sophisticated modeling of the callers, while the precise modeling of callee is ignored.

After the generation of user domain, based on the configuration parameters, a succession of arrival times for the call is generated; these times are distributed following a homogeneous poissons process, since this kind of stochastic process has characteristics that better fit with the simulation of a call's flow. The traffic generation procedure is as follows: a new call is characterized by its (global) interarrival time from the preceding one, its call duration, and finally it is assigned to one of the users identifiers available as caller and one as callee. Figure 2.1 shows the main stocastic characterization of the calls by blocks (Interarrival Time, Call Duration, Assign call to Caller and Assign call to Callee). The working procedure for these stochastic blocks are independent of each other and they can take different distributions.

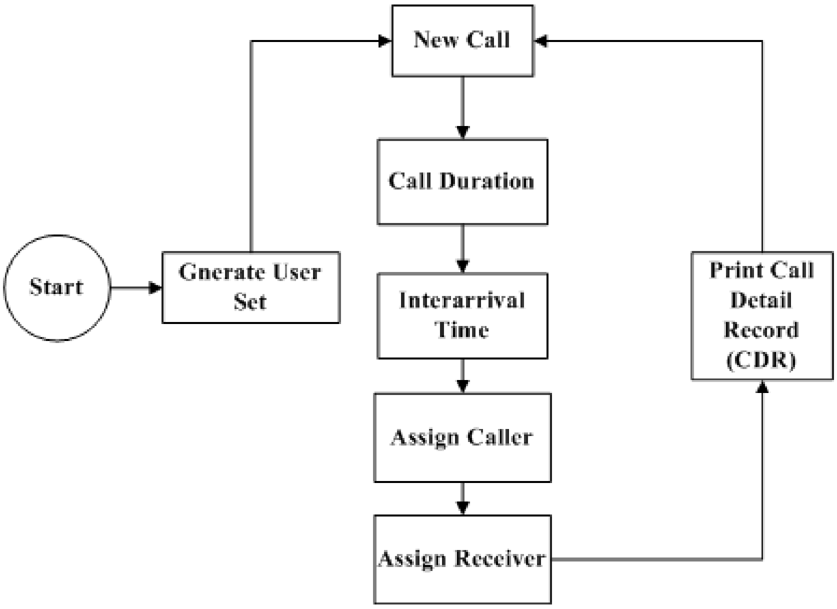


Figure 2.1: Workflow of VoTIP Traffic Generator.

The following main phases of the generation process are repeated for every call being generated:

1. generation of the arrival time of call.
2. Evaluate the duration of the call;
3. selection of the caller to assign at the current call;
4. selection of the callee for the call;

VoIPTG allows the user to choose the starting time of the observation and it starts generating call from defined starting time. Each call is identified with specific Call ID. VoIPTG is written in Java and as it may need to generate a huge number of calls, the call information is stored in batch to avoid the memory allocation problem.

2.0.2 User Domain for VoIPTG

As for traffic generation, special consideration is given on ‘volume per-user” analysis, so user domain is very important here. VoIPTG maintains two user sets to indicate the caller set who generates the call and the receiver set who receives the generated call. These two user sets are not disjoint and the configuration file also includes the parameter to set the overlapping ratio between these two user sets. This concept can be extended to the concept of internal and external user. In our Traffic Generator we have introduced two user set namely,

Caller Set (Source-Users): Users that generate calls.

Receiver Set (Receiver-Users): User that receive the generated call.

As these two user set have overlapping so the intersectiong of these two use set can be mapped as internal user.

2.0.3 User generation

The user generation process generates the users following the standard internet mailing address naming convention:

username@subdomain.domain.top-level-domain

The sub-domain generation of the domain is optional, enabled by default in the code. The top level domain list is taken from the Internet Assigned Number Authority's (IANA) official website. The user generation process is divided into two sub-processes:

Domain Generation: The first process is the domain generation where VoIPTG generates a certain number of domains, by default 1000 domains. In generating a domain, the program first chooses a top level domain randomly from the top level domain list and stores it in a variable. Then the program checks whether the sub-domain option is enabled. If the sub-domain option is enabled, the VoIPTG then defines the length of subdomain characters between 1 and the maximum sub-domain characters randomly, and uses this characters length to generate a sub-domain. By default the maximum sub-domain characters is 4. Each character for the sub-domain is chosen randomly from a set of characters [a-z, 0-9]. Then the main domain is generated by the same fashion as the sub-domain. The default characters length is 10. After generating these three elements, the program constructs the domain name

Before MD5 hashing	After MD5 hashing
gpMa@nh.fb95fd1xl.de	275e8802f2bdf44e8ffaeac2f3@a3192533a2ff47d8dccd703c3f
So3TfzQ@trz.3smhq.it	09074601fafe4ddf1f5d6b23f4@d66f8c4326adb53cf8550c5348
62DY1Ncj@55.m9my.com	60366c4eca8d9ecb13da2cf38e@1ae56f04272a20e7b6e05a9642

Table 2.1: User Generation

by concatenating them following the standard domain naming convention with "." as the separating character, and hashes this created domain name using an MD5 hash algorithm for anonymity reason.

Username Generation: The second process is the username generation where the program generates a certain number of usernames following the input from the main input file. When generating a username, VoIPTG defines the length of username characters between 1 and the maximum username characters randomly. The default maximum username characters is 10. Each character of the username is chosen randomly from a set of characters [a-z. A-Z, 0-9]. Unlike the domain's set of characters, the username's set of characters is case sensitive. The generated username is then hashed using the MD5 hasing algorithm for anonymity reason. After the two processes, it concatenates each username with a domain by randomly selecting a domain from the created domains pool. The "@" character is used to separate the username and domain while concatenating to comply with the standard internet mailing address naming convention. All users generated are written to a text file, a user per line. From all the input parameters used for the user generation process, only the number of users to be generated is read from the main input file. The rest of the parameters are defined inside the code in variables. Examples of generated users before and after being hashed using the MD5 algorithm is shown in Table 2.1.

2.1 Output of VoIPTG

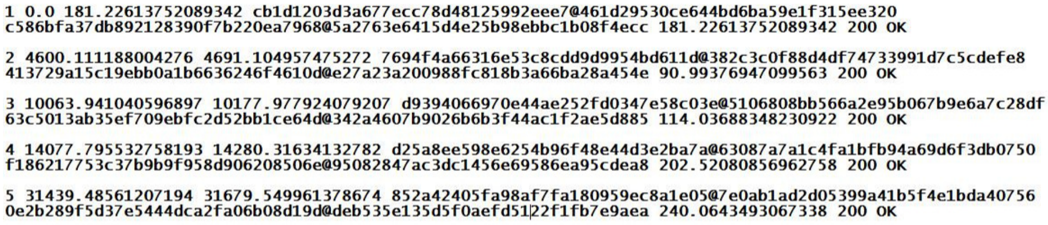
The output of the simulator is a list of calls refers to communication between two SIP user for the purposes of a multimedia conversation. In VoIPTG, Each call is identified with a Call ID. Calls are also associated with detail information such as call duration, starting time and ending time of the call, caller and receiver. The

Call Information	**Details**
Call ID	Sequence number of the generated call(long number)
Start Time	Start time of the call (second)
End Time	End time of the call (second)
Call Duration	Duration of the call (second)
Caller	User from user domain (sip address)
Receiver	User from user domain (sip address)
Response	”OK”
Response Code	200 (Code 200 indicates the established call)

Table 2.2: Output pattern of Traffic generator (Information of generated call)

output of the traffic generator includes the information for each call described in table 2.2.

A segment of the output file “call_Generation.txt” containing generated calls looks like figure 2.2.



```
1 0.0 181.22613752089342 cb1d1203d3a677ecc78d48125992eee7@461d29530ce644bd6ba59e1f315ee320
c586bfa37db892128390f7b220ea7968@5a2763e6415d4e25b98ebbc1b08f4ecc 181.22613752089342 200 OK

2 4600.111188004276 4691.104957475272 7694f4a66316e53c8cdd9d9954bd611d@382c3c0f88d4df74733991d7c5cdefe8
413729a15c19ebb0a1b6636246f4610d@e27a23a200988fc818b3a66ba28a454e 90.99376947099563 200 OK

3 10063.941040596897 10177.977924079207 d9394066970e44ae252fd0347e58c03e@5106808bb566a2e95b067b9e6a7c28df
63c5013ab35ef709ebfc2d52bb1ce64d@342a4607b9026b6b3f44ac1f2ae5d885 114.03688348230922 200 OK

4 14077.795532758193 14280.31634132782 d25a8ee598e6254b96f48e44d3e2ba7a@63087a7a1c4fa1bfb94a69d6f3db0750
f186217753c37b9b9f958d906208506e@95082847ac3dc1456e69586ea95cdea8 202.52080856962758 200 OK

5 31439.48561207194 31679.549961378674 852a42405fa98af7fa180959ec8a1e05@7e0ab1ad2d05399a41b5f4e1bda40756
0e2b289f5d37e5444dca2fa06b08d19d@deb535e135d5f0aefd5122f1fb7e9aea 240.0643493067338 200 OK
```

Figure 2.2: Output file of VoIP Traffic Generator

Chapter 3

VoIPTG Features

Analysis on VoIP traces shows that SIP based VoIP traffic does not follow any generic model to describe its characteristics like traditional telephony. So identification of metrics/parameters for characterizing SIP based VoIP traffic is utmost important. Traditional traffic models mainly focus on interarrival times and call duration in order to properly dimension the network and estimate revenues. But in modern IP telephony network dimensioning is not an issue and capacity in IP-based systems is large. Again SIP-based signaling can support millions of users with possibility of parallel calls from the same users. This flexibility brings in the necessity of identifying new parameters for characterizing complex user behavior in the network. This discussion concludes the following metrics/parameters to characterize the IP telephony.

- Call Duration
- Interarrival Time.
- Volume per Caller
- Volume per Receiver

We have analyzed telephony traces (Call Detailed Records (CDR)) to find the approximate models/patterns for the parameters characterizing IP telephony. Present VoIPTG implementation mentions the distributions/patterns defined in 3.1 for characterizing VoIP traces.

Parameter	Implemented Distribution
Call Duration	Exponential Distribution, Log Normal Distribution
Interarrival Time	Exponential Distribution
Volume per Caller	Call generation with Delta rate (Delta function), Uniform(Max-Min), Zipf
Volume per Receiver	Call receive with Delta rate

Table 3.1: List of implemented distribution for fundamental parameters for describing VoIP Traffic

3.1 Call duration Distribution

Call Duration Distribution is very significant to analyze VoIP Traffic. It is anticipated that characteristics of call duration for VoIP calls will be quite different from traditional phone calls. In traditional telephony, Call Duration is generally assumed to be exponentially distributed. But analysis on VoIP Traffic shows that VoIP Call Duration distributions are not always exponential rather can be approximated by log-normal distribution. We have implemented Exponential and Log-Normal distribution as call duration distributions.

3.1.1 Exponential Distribution for Call Duration Distribution

Call duration of the generated calls is exponentially distributed following an average call duration. Figure 3.1 shows the distribution of the call duration from generated call by VoIPTG. This trace is generated over 2 weeks observation time over 10000 users and the average call duration is 180 seconds.

3.1.2 Log Normal Distribution for Call Duration Distribution

For implementing log-normal distribution we need to define the median. Log normal distribution has infinite support but as the call durations are finite so we need to avoid the infinite property and need to define a truncation point. So with the defined median and truncation point VoIPTG will distribute the call durations following lognormal distribution where the duration of the generated calls remain

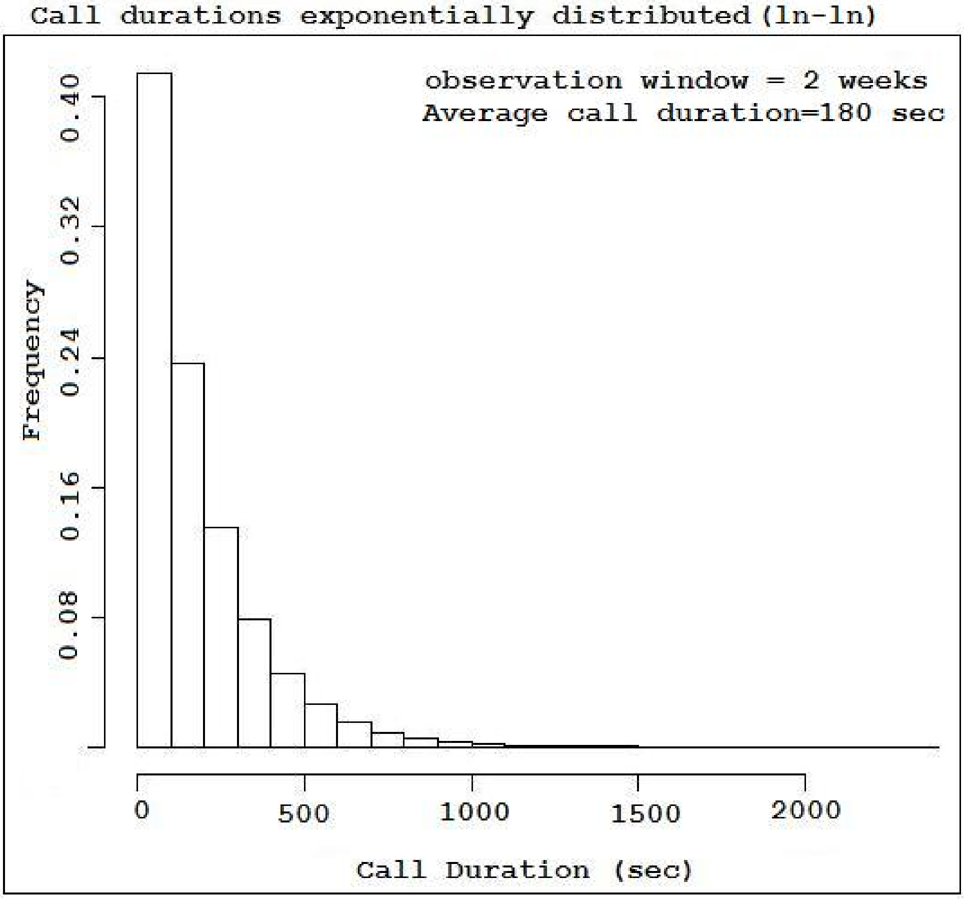


Figure 3.1: Exponential distribution for call duration

below the truncation point. Figure 3.2 shows that duration of the generated calls is following lognormal distribution.

3.2 Interarrival time Distribution

The interarrival time is a poisson process which can perfectly be described by exponential distribution. Interarrival time of the traces in traditional telephony are also exponentially distributed. Traffic analysis on IP telephony discoves that in inter-arrival time of the calls follows the exponential distribution.

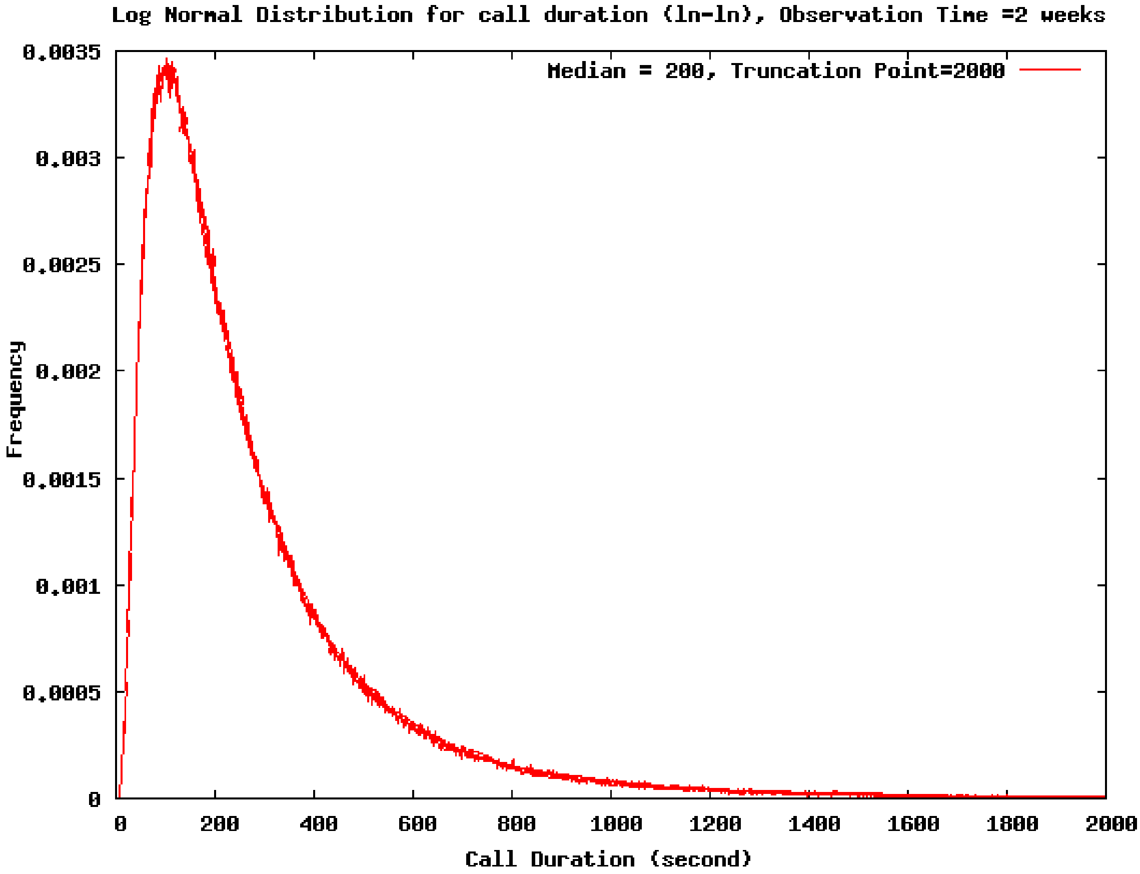


Figure 3.2: Log Normal distribution for call duration

3.3 Volume per Caller

The aim of this part is the characterization of the more sophisticated traffic parameters addressing the user behavior. Modeling the user perspective requires additional architectural decisions and it introduces dependencies among the input parameters. For modeling the user behavior it is hard to follow the straightforward approach which is the characterization of each single user. Characterizing single user can be a good approach for small number of users but with millions of users this concept will fail. So inspite of per user characterization we tried to characterize the common behavior of the user domain with complex statistical distributions.

This section deals with the selection of a *userid* from a set of callers and assigning it to the call's FROM field in the generation phase. The user selection have to be done in a way such as at the end of the calls generation, the distribution of the users's outgoing call rates follow a probability distribution configured as parameter.

The current version of the generator implements four types of distribution of the outgoing traffic:

- Calls distributed with delta rate (Delta function).
- Uniform (Min-Max) - Generation rates of the callers distributed uniformly between *min* and *max* values.
- Exponential Distribution.
- Zipf Distribution.

3.3.1 Calls generated with Delta rate (Delta function)

This is the simplest pattern to characterize the user behavior in IP telephony. Although from the analysis of VoIP calls we have seen that this distribution does not always characterize real world traffic, we have decided to implement it on behalf of the completeness of the generator. Here each user in the domain has identical call generation probability which can be defined as delta. To get this kind of traffic, indeed, the caller of generated calls are chosen uniformly among a set of users. At the end of the generation process, the representation of the probability density function of the call rates theoretically, should be a delta function. In practice, since we generate a relative small number of calls, its shape is more similar to a bell.

Here we have assigned a fixed call generation rate to for each user. If the total number of user N_u, We can assign the probability of call generation by each user as:

$$p_u = \frac{\lambda}{\sum\limits_{i \in N_u} \lambda} \tag{3.1}$$

And the λ can be defined as:

$$\lambda = \frac{1}{N_u} \tag{3.2}$$

Figure 3.3 shows that the users generate call at a rate of 60 calls per week. Here the user domain contains 10000 users.

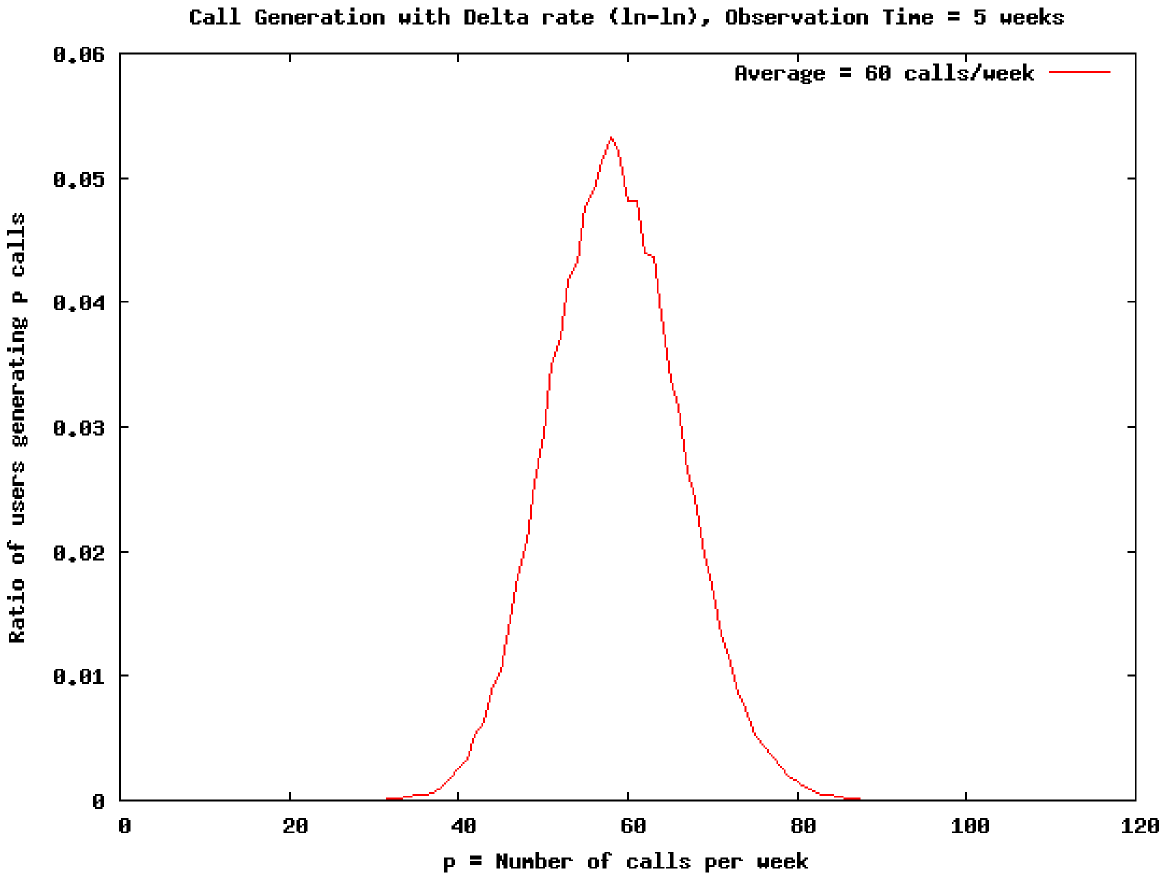


Figure 3.3: Call generation with delta rate

3.3.2 Call rates following Uniform Distribution (Min-Max)

We have implemented Uniform Distribution as Volume per Caller distribution where call rates of the users, uniformly distributed between a minimum (a) and a maximum value (b. In continuous domain the probability density function of the uniform distribution is:

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } x \in [a, b] \\ 0 & \text{otherwise} \end{cases}$$

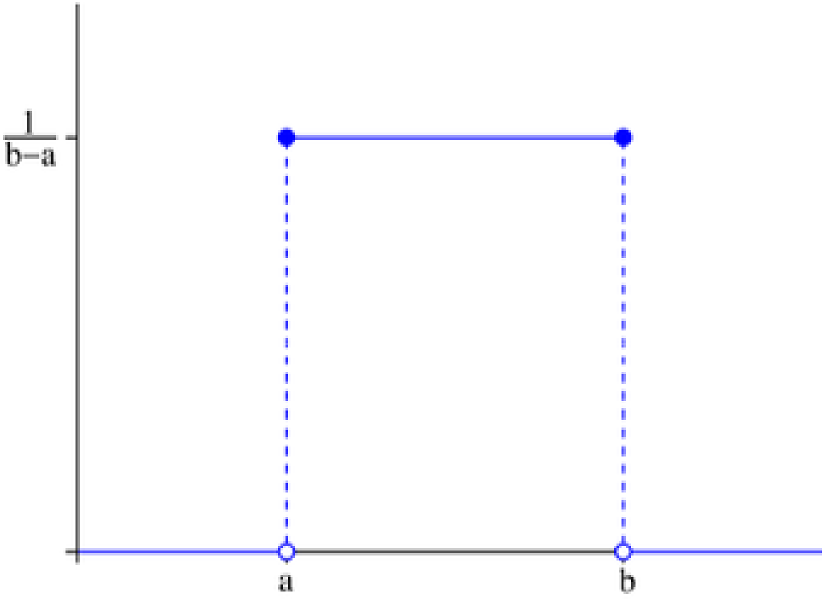


Figure 3.4: Uniform distribution, $x \in [a, b]$

Since we work with a finite support (with a finite number of users) we can approximate the p.d.f. assigning at every user a different call rate in the range $[a, b]$. In this way we have N_u (which is the number of users) call rates λ_i uniformly distributed between a and b. If the total number of user N_u, We can assign the probability of call generation by the users as:

$$p_{u,i} = \frac{\lambda_i}{\sum\limits_{i \in N_u} \lambda_i} \tag{3.3}$$

Theoritically the average of the call generation rate for all the users should be equal to the global call generation rate λ.

$$\lambda = \sum_{i \in N_u} \lambda_i \tag{3.4}$$

Theoritically, the curve found in figure 3.5 should be symmetric, but we have noticed asymmetric behavior. We have tried to figure out the reason. To understand this process, we have implemented a bimodal distribution where users are divided into two groups following two different call generation rates. In this ex-

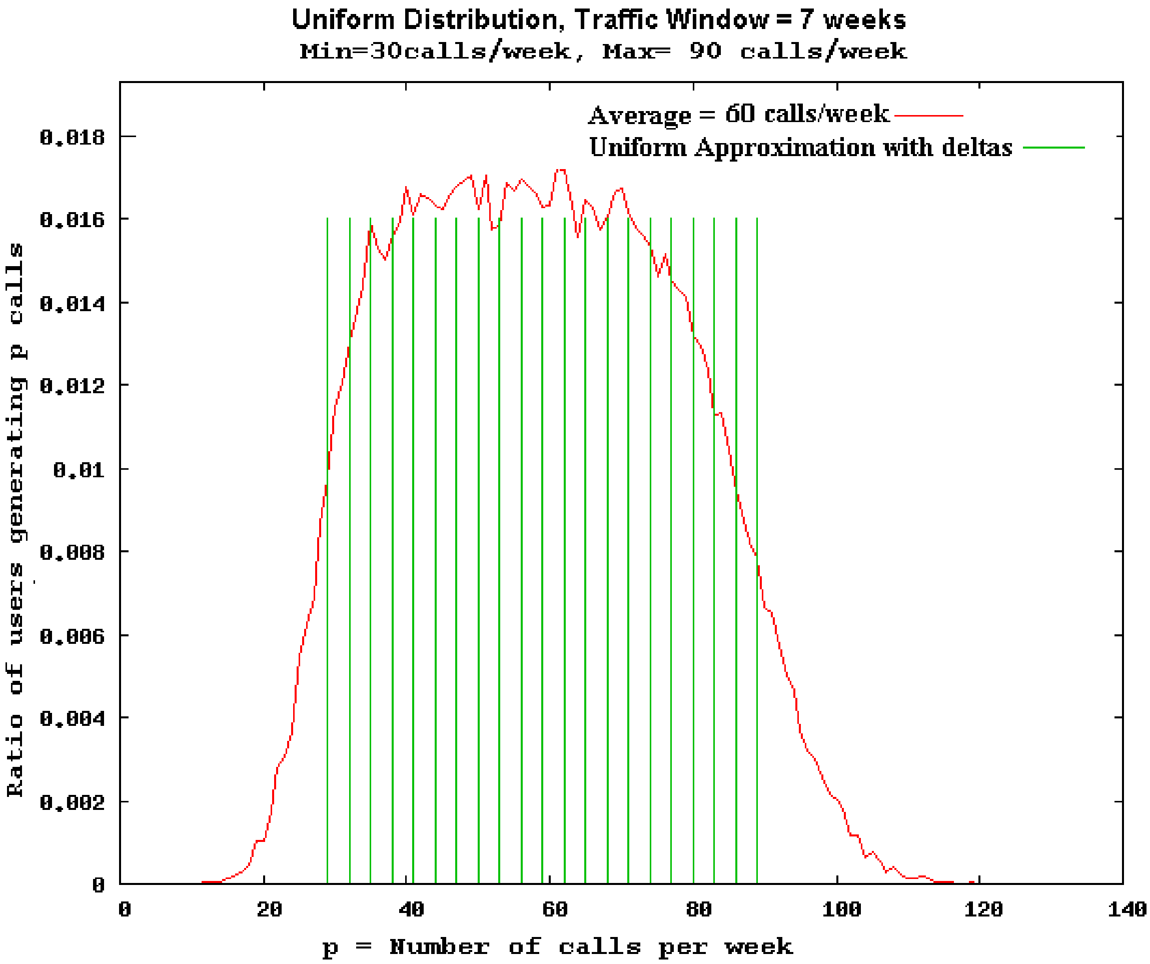


Figure 3.5: Uniform distribution, $x \in [30, 60]$ calls/week

ample, the user domain contains 10000 users and the call generation rate for first 5000 users are 30 calls/week and the rest 5000 users generates 90 calls per week. The probability density function curve from the generated calls is found in figure 3.6.

Here, we have implemented two delta functions and get two bell shaped curves. Theoritically, the properties of these two bell shaped curves should be same. If we measure the probability of call generation below 30 call/week and greater that 90 calls/week, then they should be identical. But the measured data set shows that these two probabilities are not identical and it is high for call rate x2 which is 90 calls/week. This example of bimodal distribution helps us to conclude that

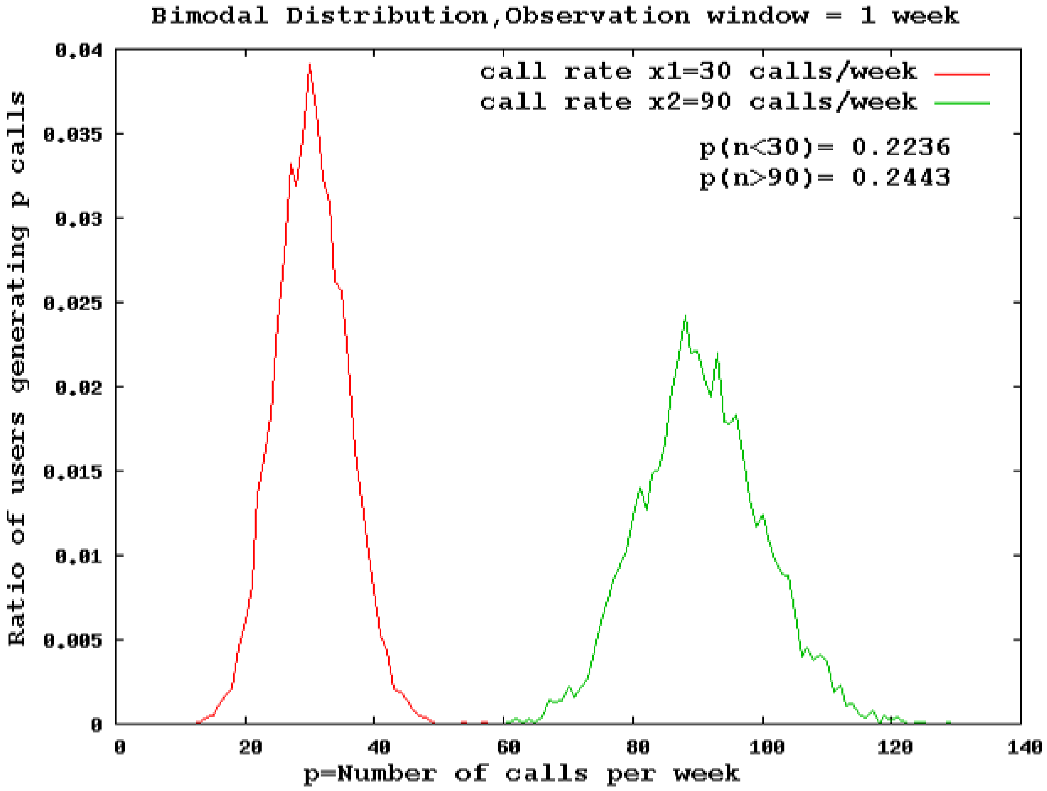


Figure 3.6: Bimodal distribution

the variability of the number of generated call increases with the increasing call generation rates. Actually implementation of the uniform (Min-Max) distribution is nothing but assigning different call rates in a range for users in domain which indicates implementation of several delta functions. As the call rates increases the variability of the generated calls also increases with the average call rate and this leads to the asymmetric behavior of the curve.

Call rates following a Exponential distribution

In real-world scenarios, the assumption of call generation in a constant rate is rarely satisfied. So we have proceeded to non-uniform distributions. We have started with implementation of Exponential Distribution as Volume per Caller distribution. In continuous domain the probability density function of the exponential distribution is:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases}$$

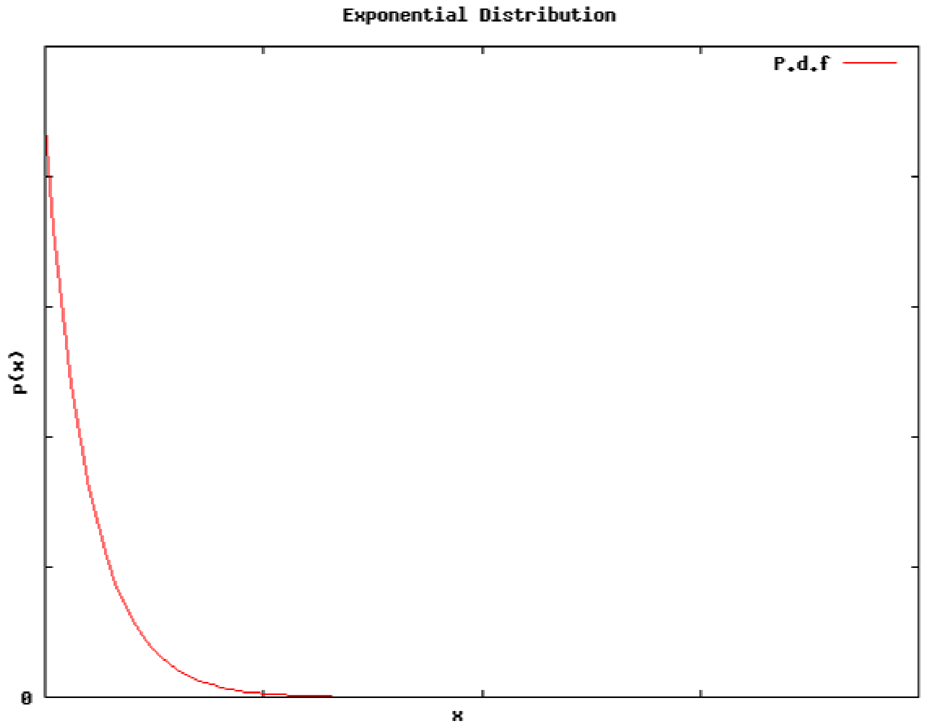


Figure 3.7: Exponential Distribution

Since we work with finite number of users and calls, we can avoid the infinite support of exponential distribution. We have truncated the c.d.f at the point which indicates the 99% of the c.d.f and approximate the p.d.f. assigning at every user a different call rate. The average call rate is evaluated from global call rate and user domain (number of caller). Figure 3.8 shows generated traffic by VoIPTG over 7 week observation time with 100000 users where the call distributed over the users is exponentially distributed.

Call rates following a Zipf distribution

It is natural that in a telephone network, some users receive more calls than others. Indeed, real-life data have verified the conjecture that the number of calls generated by users in a domain follows power-law property. So we implemented this idea by

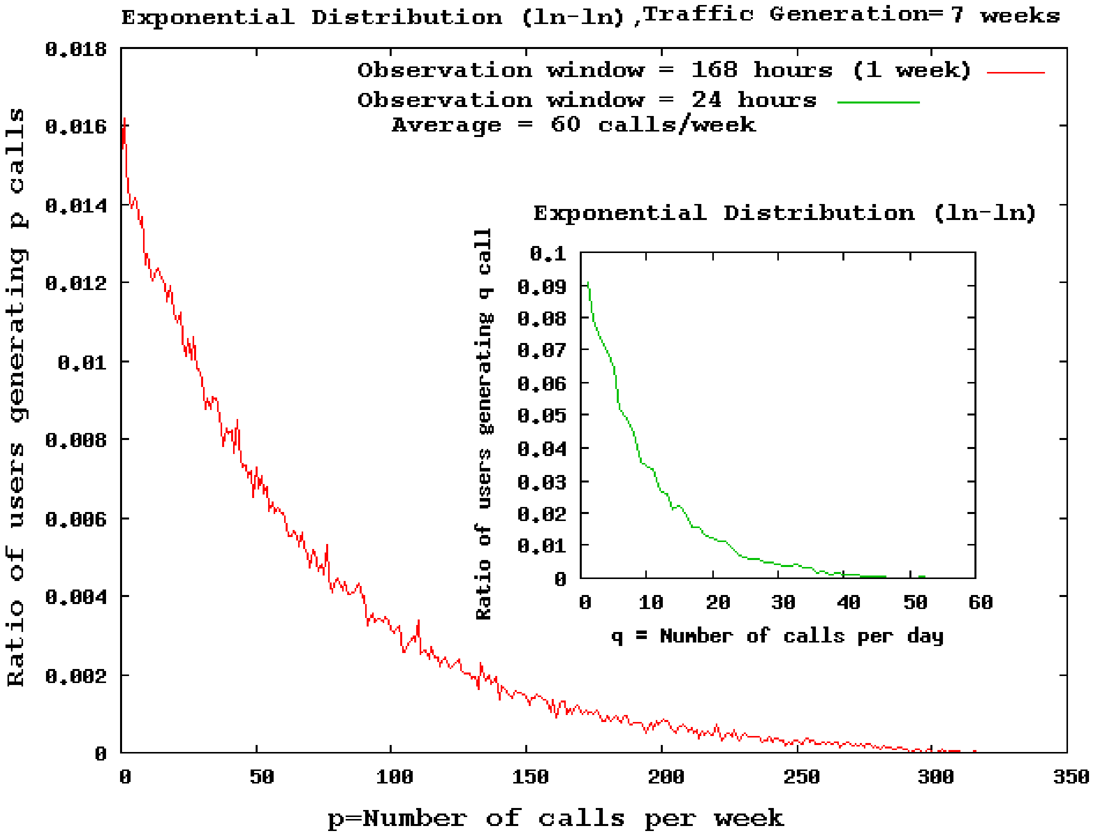


Figure 3.8: Calls over the users exponentially distributed.

modeling zipf distribution for call distribution over the caller. In our model we consider number of call as the event and number of users generating corresponding call as the frequency. If we consider simplest Zipf's law then, given that number of calls are sorted from most frequent to least frequent, the second most common frequent call will occur 1/2 as often as the first one.

Zipf distribution is the discrete version of "'Pareto Distribution"' which has a infinite support. But we are interested about Zipf Distribution because in VoIPTG a finite maximum number of calls is generated by the users in the observation period. So we first need to define the maximum number of call which possibly will be generated in the observation time by the users. This is defined as a parameter named "'Zipf Truncation Point"' in the configuration file.

Considering N as the maximum number of generated call in observation time, k as the rank of the call measured with corresponding frequency and s is the scale parameter for zipf distribution, the Probability Density function of Zipf's law can be defined in following way:

$$f_{k;s,N} = \frac{1}{k^s \times H_{N,s}}$$

Where, $H_{N,s}$ is the N^{th} generalized harmonic number.

We assume the number of calls are ranked sorted according to their frequency(Number of user generating the call)in desending order upto the define truncation point (Maximum number of generated call by user in observation time) N. For every call we can evaluate the probability of user generating that call from the above equation. Multiplying the total number of call with the probability we can evaluated the number of users generating the call. Once we have evaluated the number of users for each call, it is easy to evaluate the total number of call generated in the observation time. Then we can easily measuer the call generation probability of each user in the observation time.

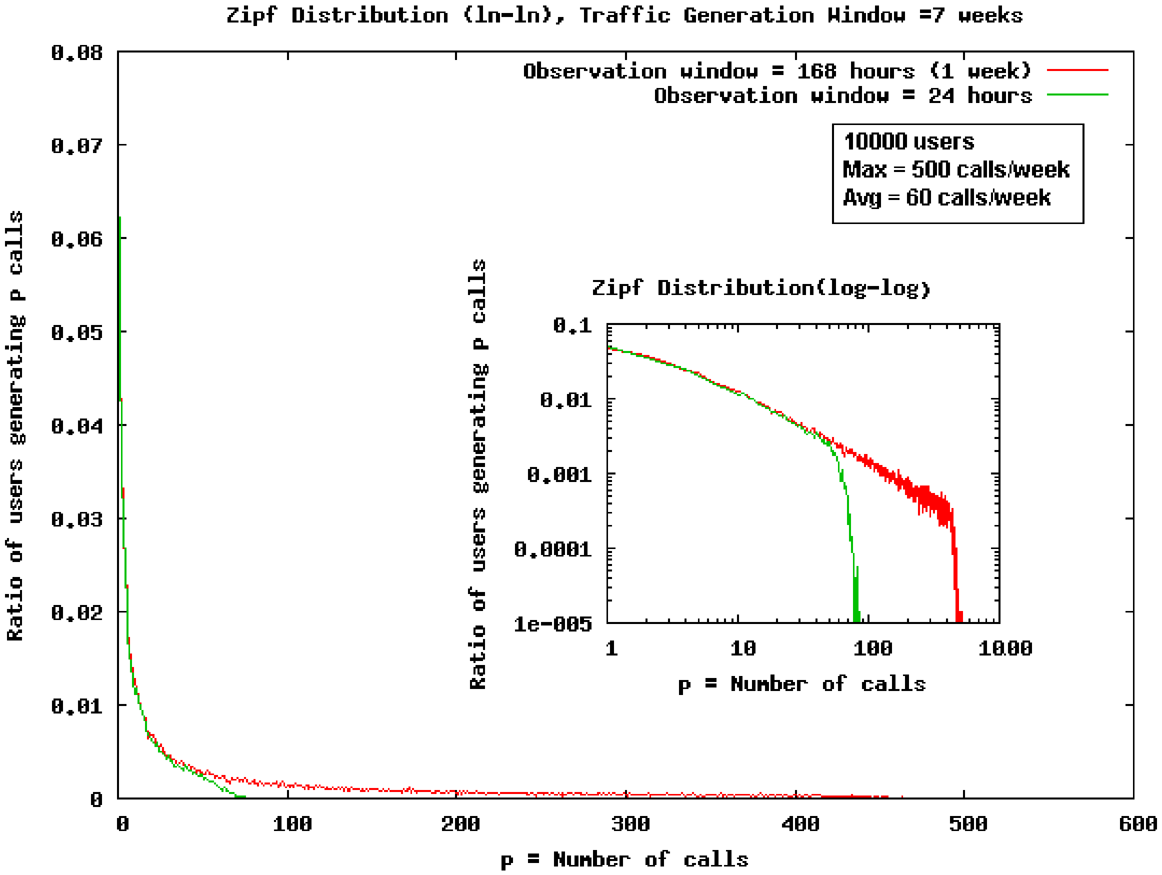


Figure 3.9: Zipf distribution for call distribution over the users.

Chapter 4

VoIPTG - User Manual

This User Manual of VoIPTG includes the short description on using the synthetic generators. It includes the installation procedure and the description over configuration parameters.

4.1 Installing VoIPTG

Users can download VoIPTG application from :

https://net-server.dit.unitn.it/svn/voipseal-new/VoIPTG/Download/.

Installation procedure of the application and the generation step of traffic is found

https://net-server.dit.unitn.it/svn/voipseal-new/VoIPTG/Download/.

4.2 How to Generate Traffic using VoIPTG

After successful installation of the application, user can generate traffic. The first step of the traffic generation with VoIPTG is to define the input parameter. A sample of the configuration file with the input parameters named as ”‘input.properties”’ is found the same directory where the application is installed. Detail desccription and guideline for configuration parameters are found in subsection 4.3. The traffic will be generated automatically following the configuration file and generated traffic is stored in a text file to the specified output location. Samples of the output for specific configuration of the parameters are also found with the application to get idea about the traffic generation process.

4.3 Configuration Parameters for VoIPTG

Configuration parameters for VoIPTG are stored in a file with '.properties" extension which is used for files mainly used in Java related technologies to store the configurable parameters of an application. The general structure of the configuration parameter is as follows:

1. The parameters are defined in following format:

   <PARAMETER NAME> = <VALUE>

2. The comments are begin with # character.

Description and guidelines for the configuration parameters are given in the following subsections.

4.3.1 Parameter for Configuring Output File name and location

Users of VoIPTG can choose the name and location of the output file to store the generated traffic information. Parameters for configuring format, name and location of the output file are defined in this section.

Parameter Name : OUTPUT_FORMAT

- Description - This parameter allows to format the call record as desired with the necessary information such as the start time, end time, duration of call, caller and the receiver. Present VoIPTG implementation supports following two output formats.

  UNITN- This format separates fields of call information by comma character (,). Start and end times of call are expressed in milliseconds while call duration in seconds. It arranges the generated call information in following fashion:

  Caller, Receiver, Call Start Time, Call End Time, Call Duration, Response Code(200), Response String (OK).

  NEC- This format separates fields of call information by comma character (,). Start, end times and call durations are expressed in seconds. It arranges the call information in following format:

Caller, Receiver, Call Start Time, Call End Time, Call Duration, Response Code(200), Response String (OK).

- Data Type - Integer
- Range - 1 : UNITN FORMAT, 2 : NEC FORMAT
- Default Value - 1
- Example - OUTPUT_FORMAT=1. This indicates that the generated calls are stored in the output file following UNITN output format.

Parameter Name : OUTPUT_DIRECTORY

- Description - Location/Path of the output file to be stored.
- Data Type - String
- Default Value - Path of the directory where the jar file of the application (Traffic Generator)is located.
- Example - C:\\TrafficGenerator\\NewCall\\ [Double \\ is needed to define the path.]

Parameter Name : OUTPUT_FILE_NAME

- Description - Name of the output file.
- Data Type - String
- Default Value - Calls.txt
- Example - Calls.txt

4.3.2 Parameter for Configuring User Sets

VoIPTG maintains two user sets to indicate the callers who generate the calls and the receivers who receive the generated calls. These two user sets are not disjoint. The overlapping percentage between the caller and receiver set is defined by overlapping percentage parameter. Based on the overlapping percentage, the total number of common users for both sets are calculated from the Caller set. Then the common users are selected randomly from the Caller set and added in the Receiver set. In other words, these users common to both Caller and Receiver

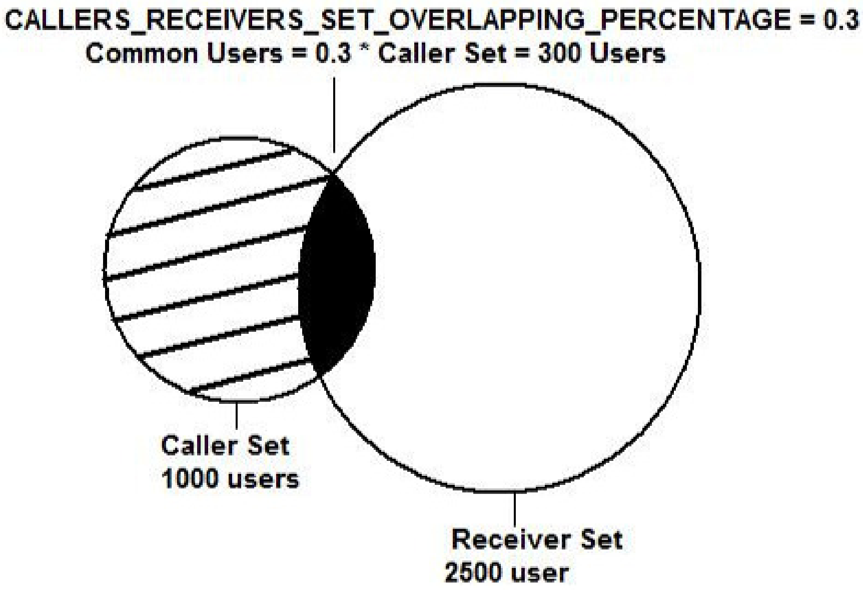

CALLERS_RECEIVERS_SET_OVERLAPPING_PERCENTAGE = 0.3
Common Users = 0.3 * Caller Set = 300 Users
Caller Set
1000 users
Receiver Set
2500 user

Figure 4.1: VoIPTG User Domain

sets can be considered as Internal Users for an Operator. User domain of VoIPTG can be explained with figure 4.1.

The configuration parameters to control the User Domain are as follows:

Parameter Name : NUMBER_CALLERS

- Description - Number of users in the Caller set.
- Data Type - Integer
- Range - 0 < NUMBER_CALLERS > 1.0^7 . The input for parameter 'NUMBER_CALLERS' should be correlated with the call generation rate of the Traffic Generator. For example with a call generation rate of 10 call/sec, Caller set containing only 100 callers may be unrealistic. This is because it indicates aroun 8640 calls per day by each user which is very high call generation rate.
- Default Value - 1000

- Example - NUMBER_CALLERS=1000. This indicates 1000 users in the Caller set.

Parameter Name : NUMBER_RECEIVER

- Description - Number of users in the Receiver set.
- Data Type - Integer
- Range - $0 <$ NUMBER_RECEIVER $> 1.0^7$. The input for parameter 'NUMBER_RECEIVER' should be correlated with the call generation rate of the Traffic Generator. For example with a call generation rate of 10 call/sec, Receiver set containing only 100 receiver may be unrealistic. This is because it indicates on average 8640 calls will be received by a receiver which is very high rate.
- Default Value - 1000
- Example - NUMBER_RECEIVER=1000 indicates 1000 users in the Receiver set.

Parameter Name : CALLERS_RECEIVERS_SET_OVERLAPPING_PERCENTAGE

- Description - Ratio of the overlapping users from Caller set to receiver set.
- Data Type - Double
- Range - Min : 0 - Max : 1
- Default Value - 0.2
- Example - If NUMBER_CALLERS=1000, NUMBER_RECEIVER=2500 and CALLERS_RECEIVERS_SET_OVERLAPPING_PERCENTAGE=0.3, then from the overlapping percentage we can evaluate that 300 users are common in Caller and Receiver set. This number of common users is calculated over the Caller Set. The graphical representation of this example is found in figure 4.1.

4.3.3 Parameter for Generating Users' Address

VoIPTG generates users following the standard internet mailing address naming convention:

username@subdomain.domain.top-level-domain

The parameters to generte the users' address are as follows:

Parameter Name : NUMBER_DOMAINS

- Description - Number of Domains.
- Data Type - Integer
- Default Value - 1000
- Example - NUMBER_DOMAINS = 1000

Parameter Name : SUBDOMAINS_ENABLED

- Description - Boolean parameter to choose whether subdomain is used in forming users' address.
- Data Type - Boolean
- Range - true, false
- Default Value - true
- Example - SUBDOMAINS_ENABLED=true indicates that sub-domain name will be used in forming the users' address.

Parameter Name : MAX_CHARS_USERS

- Description - Maximum number of character used to generate user name part of the user's address.
- Data Type - Char
- Range - Each character of the username is chosen randomly from a set of characters [a-z. A-Z, 0-9].
- Default Value - 10
- Example - MAX_CHARS_USERS=8 indicates that at most 8 characters will be used to generate user name in the user address.

Parameter Name : MAX_CHARS_DOMAINS

- Description - Maximum number of character used to generate domain name of the user's address.

- Data Type - Char
- Range - Each character for the sub-domain is chosen randomly from a set of characters [a-z, 0-9].
- Default Value - 10
- Example - MAX_CHARS_DOMAINS=9

Parameter Name : MAX_CHARS_SUBDOMAINS

- Description - Maximum number of character used to generate subdomain name of the user's address.
- Data Type - Char
- Range - Each character for the sub-domain is chosen randomly from a set of characters [a-z, 0-9]
- Default Value - 4
- Example - MAX_CHARS_SUBDOMAINS =3

4.3.4 Parameter for Configuring Observation Window

VoIPTG users can control the observation window of traffic generation. It is possible to start traffic generation from any specific time in a week. The parameters responsible for controlling the observation window are following:

Parameter Name : OBSERVATION_TIME

- Description - Time window / Duration (in week) of Traffic generation.
- Data Type - Double
- Range - It should be non-negative and greater than zero. OBSERVATION_TIME > 0
- Default Value - 1
- Measure Unit - Week.
- Example - OBSERVATION_TIME = 2 indicates traffic generation will continue for 2 weeks.

Parameter Name : START_OBSERVATION_DAY

- Description - Starting day of traffic generation.
- Data Type - Integer
- Range - Min : 0 - Max : 6 , This parameter can hold a number from 0 to 6 indicating 7 days in a week. Here, 0 - Monday, 1-Tuesday, 2-Wednesday, 3 - Thursday, 4-Friday, 5- Saturday and 6- Sunday.
- Default Value - 0 (Monday)
- Example - START_OBSERVATION_DAY =1 indicates that traffic generation will start from Tuesday.

Parameter Name : START_OBSERVATION_HOUR

- Description - Starting hour of traffic generation.
- Data Type - Integer
- Range - Min : 0 - Max : 23 . This parameter can hold a number from 0 to 23 indicating 24 hours in a day.
- Default Value - 9
- Example - START_OBSERVATION_HOUR=9 indicates that traffic generation will start from 9 AM in the morning on spcified starting day.

4.3.5 Parameter for configuring Traffic Variance and Intensity

VoIPTG allows traffic variance over weeks and days and considers that traffic intensity degrades during the weekends and also during the off-peak hour in a day. Configuration file allows setting parameters for defining the traffic degradation in weekends by defining the weekly traffic reduction percentage. And the daily traffic variance is considered by setting the value of peak and off-peak hour in a day. VoIPTG users set the parameter for the starting and ending time of the peak hour where a day is divided in 24 (0 to 23) hour slots. So a value of peak hour start time as 9 and peak hour end time as 21 means the peak hour starts from 9 am and ends at 9 pm. Parameters for getting traffic variance over weeks and also during the day are follows:

Parameter Name : HIGH_DAILY_RATE

- Description : Traffic Intensity/Traffic generation rate at peak hours in a day.
- Data Type : Double
- Range : HIGH_DAILY_RATE > 0, It should be non-negative and greater than zero.
- Default Value : 0 2
- Measure Unit : Call per second.
- Example : HIGH_DAILY_RATE =0.2 indicates that during peak hours in a day VoIPTG will generate 0.2 calls per second.

Parameter Name : LOW_DAILY_RATE

- Description - Traffic Intensity/Traffic generation rate at off-peak hours in a day.
- Data Type - Double
- Range - LOW_DAILY_RATE > 0, It should be non-negative and greater than zero.
- Default Value - 0.1
- Measure Unit - Call per second.
- Example - LOW_DAILY_RATE =0.1 indicates that during off-peak hours in a day the call generation rate will degrades and then VoIPTG will generate 0.1 calls per second.

Parameter Name : TRAFFIC_VOLUME_REDUCTION_PERCENTAGE_IN_WEEKEND

- Description - This parameter holds the percentage of decrease in traffic volume in the weekends. Generally in weekend the traffic volume degrades. This parameter allows us to define the traffic degradation percentages based on traffic volume in the week days.
- Data Type - Double
- Range - Min : 0 - Max : 1.
- Default Value - 0.5

- Example - TRAFFIC_VOLUME_REDUCTION_PERCENTAGE_IN_WEEKEND = 0.5 indicates that in weekend (Saturday and Sunday) traffic volume reduces to 50% of the traffic volume in the week days.

Parameter Name : HIGH_DAILY_RATE_START

- Description - Starting of Peak-Hour in a day. This parameter holds the starting time of peak hour where a day is divided in 24 (0 to 23) hour slots.
- Data Type - Integer
- Range - Min : 0 - Max : 23 . This parameter can hold a number from 0 to 23 indicates 24 hours in a day.
- Default Value - 9
- Example - HIGH_DAILY_RATE_START =9 indicates that peak hour statrs from 9 AM in the morning on each day. And during peak hour the traffic generation rate follows the HIGH_DAILY_RATE.

Parameter Name : HIGH_DAILY_RATE_END

- Description - Ending time of peak hour/ Starting time of off-peak hour. This parameter holds the ending hour of peak time where a day is divided in 24 (0 to 23) hour slots.
- Data Type - Integer
- Range - Min : 0 - Max : 23 . This parameter can hold a number from 0 to 23 indicates 24 hours in a day.
- Default Value - 16
- Example - HIGH_DAILY_RATE_END =16 indicates that peak hour ends at 4 PM in the afternoon everyday and the off-peak hour starts from 5 PM in the afternoon. During the off-peak hour the traffic generation rate follows the LOW_DAILY_RATE.

Figure 4.2 shows traffic distribution over the a week following the configuration parameter for Traffic Variance and Intensity. The parameter configuration for this example is found in table 4.1.

Parameter	Value
HIGH_DAILY_RATE	1 call/sec
LOW_DAILY_RATE	0.5 call/sec
TRAFFIC_VOLUME_REDUCTION_PERCENTAGE_IN_WEEKEND	0.5%
HIGH_DAILY_RATE_START	0 (12 AM)
HIGH_DAILY_RATE_END	12 (12 PM)

Table 4.1: Configuring parameters for Traffic Variance and Intensity

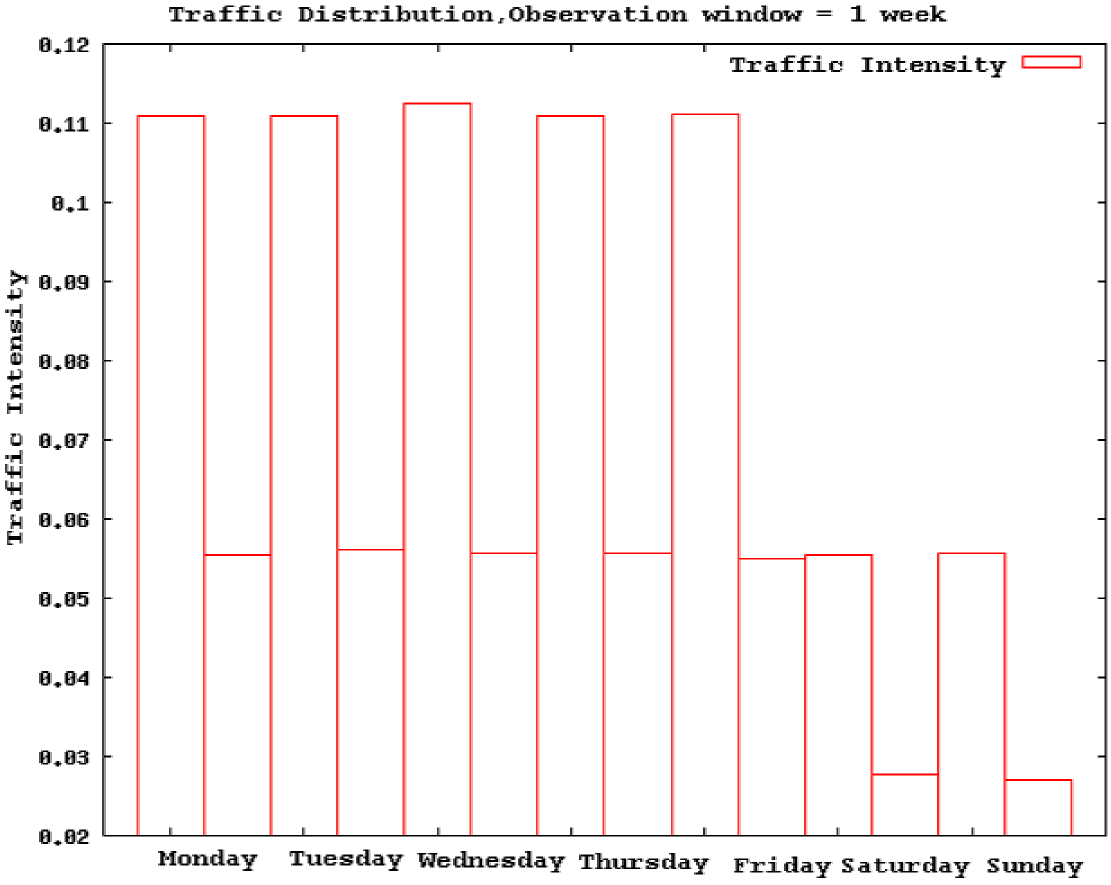


Figure 4.2: Traffic Distribution over a week.

4.3.6 Parameter for Interarrival Time Distribution

In present implementation of VoIPTG, the interarrival time of the calls are exponentially distributed.

Parameter Name : INTERARRIVAL_DISTRIBUTION_FUNCTION_TYPE

- Description - This paramete holds the choice for distribution type for the interarrival time of the calls.
- Data Type - Integer
- Range - 1 : Exponentical Distribution
- Default Value - 1
- Example - INTERARRIVAL_DISTRIBUTION_FUNCTION_TYPE= 1.

4.3.7 Parameter for Call Duration Distribution

VoIPTG is able to generate call following different distributions for the call duration. Present implementation supports exponential and Log Normal distribution for call duration. Users first need to define the choice for distribution for call duration. The parameter for choosing distributions is following:

Parameter Name : CALL_DURATION_DISTRIBUTION_FUNCTION_TYPE

- Description - This paramete holds the choice of distribution type for the call duration.
- Data Type - Integer
- Range - 1 : Exponentical Distribution, 2 : Log Normal Distribution.
- Default Value - 1
- Example - CALL_DURATION_DISTRIBUTION_FUNCTION_TYPE = 1 indicates that the duration of the generated calls are exponentially distributed.

According to the choice of parameter ‘CALL_DURATION_DISTRIBUTION_FUNCTION_TYPE users need to define the configuration parameters for specific distribution. The configuration parameters for other distributions can be ignored.

Parameter for Exponential distribution is follows:

Parameter Name: AVERAGE_CALL_DURATION

- Description - Average Duration of the calls in the observation time. It is used as Mean value for exponential distribution.

- Data Type - Double
- Range - 0 < AVERAGE_CALL_DURATION < 3600 sec (1 hour).
- Measure Unit - second
- Example - AVERAGE_CALL_DURATION = 180 indicates that the average duration of the generated calls is 180 seconds.

Parameters for Log Normal distribution are follows:

Parameter Name: LOGNORMAL_TRUNCATION_POINT_FOR_DURATION

- Description - This parameter defines the truncation point for Log Normal Distribution as the infinite support of Log Normal distribution is not needed (Due to finite duration of calls). Here the truncation point indicates the maximum duration of the generated call in the observation time.
- Data Type - Double
- Range - This parameter is used to restrict the heavy tail of the log normal distribution. We are not interested of calls with large duration as long duration calls are not so frequent, we should set the truncation point less that 60 minutes.

  0 < LOGNORMAL_TRUNCATION_POINT_FOR_DURATION < 3600 sec (1 hour)
- Measure Unit - second
- Example - LOGNORMAL_TRUNCATION_POINT_FOR_DURATION = 1800 indicates that the maximum duration of the calls can be 1800 seconds (30 min).

Parameter Name: LOGNORMAL_MEDIAN_FOR_DURATION

- Description - This parameter defines the median for Log Normal Distribution.
- Data Type - Double
- Range - 0 < LOGNORMAL_MEDIAN_FOR_DURATION < 3600 sec (1 hour)

- Measure Unit - second
- Example - LOGNORMAL_MEDIAN_FOR_DURATION = 87 indicates that the median of duration of the generated calls is 87 seconds.

4.3.8 Parameters of call distribution over the caller

VoIPTG provides the opportunity to switch between four distributions for call distribution over the user. These are call generation with delta probability, Uniform (Max-Min), Exponential and Zipf distribution. Users first need to choose the distribution for volume per user. The parameter for choosing distributions is following:

Parameter Name : CALLER_CALL_DISTRIBUTION_TYPE

- Description - This paramete holds the choice for distribution type for volume per caller.
- Data Type - Integer
- Range - 1 - Delta Probability, 2 - Uniform (Max-Min) Distribution ,3 - Exponential Distribution, 4 - Zipf Distribution.
- Default Value - 1
- Example - CALLER_CALL_DISTRIBUTION_TYPE = 2 indicates that the call generation by the users follows Uniform distribution bounded by a range from minimum to maximum calls.

According to 'CALLER_CALL_DISTRIBUTION_TYPE", users need to define the configuration parameters for specific distribution. The configuration parameters for other distribution can be ignored. The configuration parameters for specific distributions are defined in following subsections.

Parameters of Delta Probability

To generate call with delta probability, it requires only the total number of caller in the caller domain to evaluate the delta probability for all the call generating calls. Here the call generation probability for all the caller are same.

Parameters of Uniform (Min-Max) Distribution

In Uniform(Min-Max) distribution, the users generate call within a range with equal probability. So it requires two parameters defining this range. Configuration parameter for Uniform (Min-Max) distribution are follows:

Parameter Name: UNIFORM_MAX

- Description - Maximum number of call generated by user in a week.
- Data Type - Double
- Range - This parameter indicates the upper range of Uniform distribution. It should be correlated with the global call generation rate of the generator. HIGH_DAILY_RATE contains the global call generation rate per second. From HIGH_DAILY_RATE we can evaluate the call generation rate per week and this value should be the average of UNIFORM_MAX and UNIFORM_MIN.
- Default Value - 90
- Measure Unit - Call per Week
- Example - UNIFORM_MAX = 60 indicates that the users generates maximum 60 calls in a week.

Parameter Name: UNIFORM_MIN

- Description - Minimum number of call generated by user in a week.
- Data Type - Double
- Range - This parameter indicates the lower range of Uniform distribution. It should be correlated with the global call generation rate defined in parameter HIGH_DAILY_RATE. From HIGH_DAILY_RATE we can evaluate the call generation rate per week and this value should be the average of UNIFORM_MAX and UNIFORM_MIN.
- Default Value - 30
- Measure Unit - Call per Week
- Example - UNIFORM_MAX=90 and UNIFORM_MIN=30 indicate that 90 is the maximum number of call can be generated by the user in a

week while the minimum number of generated call is 30 in a week. Again users can generate call in this rage from 30 to 90 with identical probability. Here users can generate on average 60 calls per week. So to get this desired output the generator must be able to generate on average 60 calls per week. This introduces the necessity to define the global call generation rate very carefully at the time of defining UNIFORM_MAX and UNIFORM_MIN. In this example the parameter HIGH_DAILY_RATE which indicates the global call generation per second would be 1. This means on average the generator is able to generate 60 calls per week. But if the user choose global call generation rate HIGH_DAILY_RATE=2 then it will surely contradit with the defined UNIFORM_MIN and UNIFORM_MAX value with average 120 calls per week and fails to produce desired output.

We should be carefull to define the global call generation rate because this should be equal to the average call generation rate by the users. Sometimes due to daily and weekly traffic degradation users do not achieve desired result. For example, UNIFORM_MAX=90 and UNIFORM_MIN=30 indicates that the users will generate 60 calls per week. So the global call generation rate must be 60 calls/week to get the desired distribution. HIGH_DAILY_RATE = 1 call/sec indicates that on average VoIPTG will generate 60 calls/sec. But if we also introduce traffic degradation in the off-peak hours and in weekends then with call generation rate of 1 call/Sec will fails to give perfect result. This situation is found in figure 4.3. To avoid this situation we need to increase global call generation rate to maintain the generation rate as 60 calls/week.

Parameters of Exponential Distribution

In Exponential Distribution, the call generation by the users is exponentially distributed. So it requires one parameters defining mean value of the generated call by the user. The average call generated by the user need not be defined externally, the generator can calculate it through the call genration rate and the total number of callers.

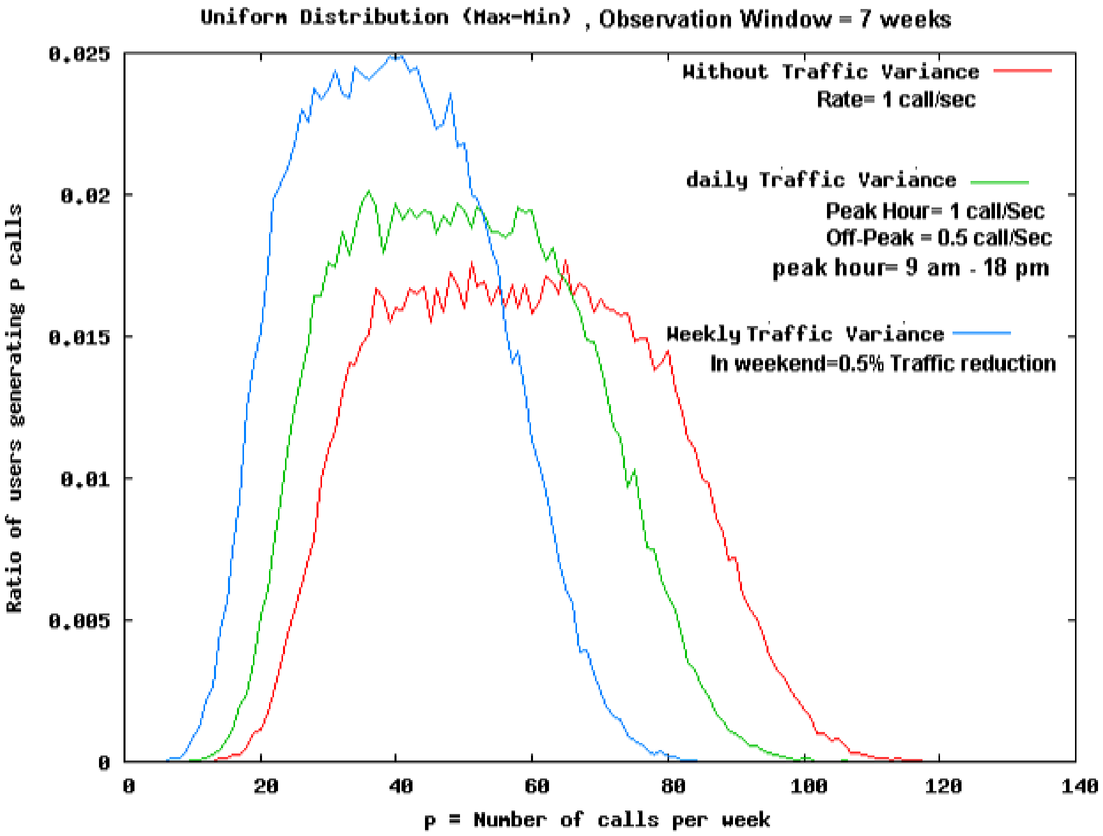


Figure 4.3: Uniform Distribution for volume per caller distribution.

Parameters of Zipf Distribution

For Zipf Distribution, we need to define the truncation point which is the maximum number of generated call by the user and the scale parameter.

Parameter Name: ZIPF_TRUNCATION_FOR_CALLERS_DIST

- Description - Maximum number of calls generated by user in a week.
- Data Type - Integer
- Range - 0< ZIPF_TRUNCATION_FOR_CALLERS_DIST < ((Call generation rate per week)* 10).
- Measure Unit - Maximum Number of generated calls in a week.

- Example - ZIPF_TRUNCATION_FOR_CALLERS_DIST = 500 indicates that one users can generate maximum 500 calls in 1 weeks observation time.

Parameter Name: ZIPF_SCALE_FOR_CALLERS_DIS

- Description - Scale parameter for Zipf Distribution. This parameter defines the spread of the zipf distribution.
- Data Type - Double
- Range - 0 < ZIPF_SCALE_FOR_CALLERS_DIS < 5
- Default Value - 0.9
- Example - ZIPF_SCALE_FOR_CALLERS_DIST = 0.9

4.3.9 Parameters of call distribution over the Receiver

Present VoIPTG implementation only supports simplest distribution for call distribution over the receiver. The generated calls are distributed over the receiver with a fixed probability defined as delta. As in future we will implement other distributions for call distribution over the receiver, we introduce the following parameter to choose distributions.

Parameter Name : RECEIVER_CALL_DISTRIBUTION_TYPE

- Description - This paramete holds the choice for distribution type for volume per receiver.
- Data Type - Integer
- Range - 1 - Delta Probability
- Example - RECEIVER_CALL_DISTRIBUTION_TYPE = 1 .

According to the choice of parameter 'RECEIVER_CALL_DISTRIBUTION_TYPE ", users need to define the configuration parameters for specific distribution.

Parameters of Delta Probability

To distribute call over the receiver with delta probability, it requires only the total number of receiver in the receiver domain to evaluate the call receive probability. Here the call receiver probability for all the receiver are same which is defined as delta.