J-OWAMP

Java Implementation of OWAMP

User's Manual
for versions 1.2 and 2.1

Authors: Hélder Veiga, Teresa Pinho, José L. Oliveira, Rui Valadas, Paulo Salvador, António Nogueira

Affiliation: University of Aveiro / Institute of Telecommunications Aveiro
hveiga@av.it.pt; {jlo, rv}@det.ua.pt; {salvador, nogueira}@av.it.pt

Last updated: 12-12-2005

Contents

1 Introduction and requirements

In order to create an innovator platform for active measurements, that can also represent a basis for the development and test of new algorithms and models, we built a system designated by J-OWAMP (that stands for Java implementation of OWAMP) that corresponds to the analogous of the One-Way Active Measurement Protocol (OWAMP) model. The previous versions of J-OWAMP (version 1.2) implement the May 2004 draft proposal of OWAMP (http://www.internet2.edu/~shalunov/ippm/draft-ietf-ippm-owdp-08.txt). The current version of J-OWAMP (version 2.1) implements the December 2004 draft proposal of OWAMP (http://www.internet2.edu/~shalunov/ippm/draft-ietf-ippm-owdp-14.txt).

The J-OWAMP system can be used both in Windows and Linux platforms, requiring only the installation of the J2SE Java Runtime Environment (JRE), available for both Windows and Linux. The installation of this module is mandatory, in order to allow the execution of Java applications. The J2SE Java Runtime Environment (JRE) 5.0 is needed for having IPv6 support on Microsoft Windows. With the J2SDK/JRE 1.4 release, there is IPv6 support on Linux.

Note: For IPv6 support, addresses with scope link-local (start with FE8) or site-local (start with FEC) shouldn’t be used.

You must also save a MySQL Connector/J file (http://dev.mysql.com/downloads/) in the *"<jre_diretory>\lib\ext"* directory to allow the interaction between J-OWAMP and a MySQL database. MySQL Connector/J is used for connecting to MySQL from Java.

OWAMP requires a synchronized clock in order to provide meaningful measurements. But, more importantly, the clock needs to be stable. If the system clock is stepped during an OWAMP session, the results can be misleading. OWAMP requires the use of GPS or NTP to synchronize the system clock.

The time synchronization, accuracy and resolution are some of the most important requirements of OWAMP. Better accuracy and resolution can be obtained by using performance counters in conjunction with the system time, in order to calculate accurate and smaller time increments. To do so, a time reference should be defined. In any instant the current time can be determined as follows: *currentTime = timeReference + (currentCounterValue - referenceCounterValue)/counterFrequency*. The time reference can be defined by using the system time, in which case the system should be synchronized, or directly from a NTP server. J-OWAMP allows these two approaches. For the second approach, J-OWAMP includes a NTP client which can be used to get the time reference directly from a NTP server, without the requirement of a synchronized clock on the machine hosting J-OWAMP. This approach can lead to even better results than the first one. On both approaches, time reference should be frequently synchronized. Therefore, an update interval should be defined.

With J-OWAMP it is possible to accomplish two kinds of test sessions: single test sessions and confidence interval test sessions. The single test session corresponds to the configuration of one single test between two machines. The user can define the number of packets, the packet interarrivals, the packet length and the start time of the session. The results obtained with this type of session are the estimation performance metrics such as mean delay, losses, duplication and throughput. The confidence interval test session allows splitting the complete session period in a number of smaller test intervals and to perform a set of tests in each interval to enable the construction of 90% confidence intervals. For example, if the session period is set at 24 hours, the user can define 24 intervals (of one hour duration), and define also that in each interval a total of 10 tests should be performed in order to calculate the 90% confidence interval. In this case, the interval between the start of consecutive session can also be defined (say 2 minutes).

In this manual, we will refer to the Windows version of J-OWAMP for explaining how to use this application to perform active measurements. The manual is structured in the following way: section 2 presents the system architecture; section 3 shows how to run J-OWAMP; section 4 describes in detail all the necessary steps to conveniently configure the different system modules and section 5 describes in detail all the necessary steps to conveniently configure the CalculateCI module.

2 System architecture

The J-OWAMP system implements the OWAMP architecture shown in **Figure 1**:

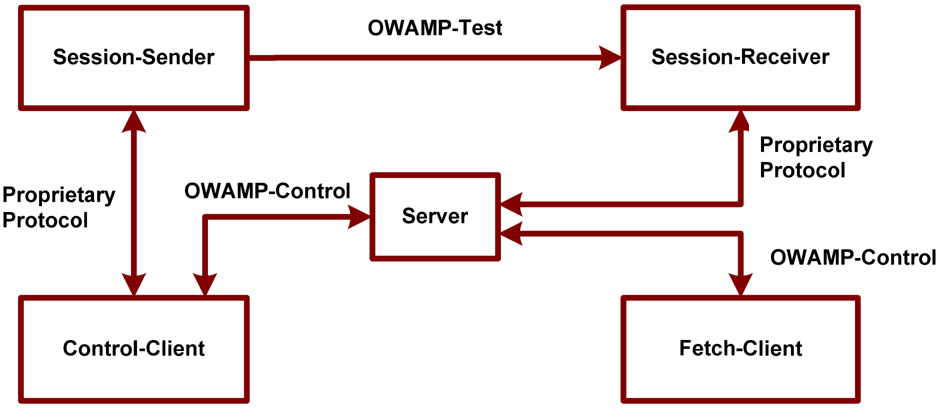


Figure 1: OWAMP architecture

The OWAMP architecture is based on two inter-dependent protocols, the OWAMP-Control and the OWAMP-Test, which can guarantee a complete isolation between client entities and server entities. The OWAMP-Control is used to begin and end test sessions as well as receive the results of those tests, whereas the OWAMP-Test protocol is used to allow the exchange of test packets between any two points that belong to the monitored network.

The proposed architecture includes the following elements (**Figure 1**):

- ***Session-Sender:*** the sender of the test packets. It is implemented by the ***OWAMP_SessionSender.exe*** application.
- ***Session-Receiver:*** the receiver of the test packets. It is implemented by the ***OWAMP_SessionReceiver.exe*** application.
- ***Server:*** the entity that is responsible for the global management of the system. It can configure both network terminal elements that are being tested and receive the results of a test session. It is implemented by the ***OWAMP_Server.exe*** application.
- ***Control-Client:*** a terminal system that programs the requests for test sessions, triggers the beginning of a session set and can also finish one or all ongoing sessions. It is implemented by the ***OWAMP_ControlClient.exe*** application.
- ***Fetch-Client:*** a terminal system that triggers the requests for results of test sessions that have already ended or are still running. It is implemented by the ***OWAMP_FetchClient.exe*** application.

A network element can carry out several logical functions at the same time (simplified scenario). For example, we can have only two network elements: one is carrying out the functions corresponding to a Control-Client, a Fetch-Client and a Session-Sender and the other one is carrying out the functions corresponding to a Server and a Session-Receiver (**Figure 2**).

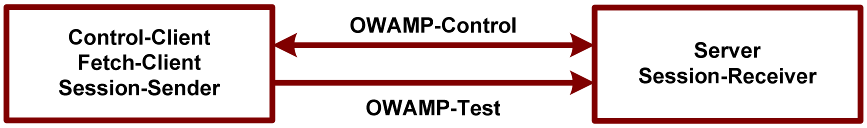


Figure 2: OWAMP simplified architecture

The architecture of **Figure 1** allows the definition of only one client and one server in the network (possibly installed in machines with the highest processing capacity) and allows the installation of senders and receivers in any machine of the network, which leads to a lower processing impact. In this way, the network manager can perform tests all over the network from a single machine, which is not possible in the simplified scenario.

Note: All machines that are in charge of executing the different applications that compose this measurement system should be precisely synchronized in order to guarantee good measurements results.

3 Running J-OWAMP

To run J-OWAMP the user has to execute each one of the elements (applications) that compose its architecture: it can make a double click with the mouse button in the respective icon or use the command line (DOS window in Windows OS). Using double click the default port numbers that will be used in the TCP connections between the different elements of the J-OWAMP architecture are the following:

J-OWAMP element	**Port number**
OWAMP-Server	22368
OWAMP Session-Receiver	28181
OWAMP Session-Sender	4181

Table 1: Port numbers used in each element for TCP connections

In order to use other port numbers, each one of these modules must be executed in the command line using the following format:

OWAMP-Server:

Usage: *OWAMP_Server [-options]*

where options include:

-a | -address	*The local IP address to bind to. This address can be used on a multi-homed host for a Server that will only accept connection requests to one of its addresses. If not defined, it will by default accept connections on any/all local addresses.*
-p | -port port	*The TCP port number to be used in OWAMP-Control.*
-r | -receiverTCPport port	*TCP port number to be used in OWAMP-Control with all OWAMP Session-Receivers.*
-n | -ntpServer	*The address of the NTP server used to get the time reference. If not defined, the system time will be used to get the time reference.*
-u | -updateInterval	*The value (in minutes) to be used for the time interval between two consecutive updates of the time reference.*
-? -help	*Print this help message.*

The *receiverTCPport* is the TCP port number to be used in the OWAMP-Control between the Server and all OWAMP Session-Receivers. It is the TCP port where the OWAMP Session-Receivers will be waiting for connection requests. If not defined, the default receiver TCP port 28181is used for OWAMP-Control.

Note: The configuration (Request-Session packet and the schedule slot description packets) of each test session processed is saved to a file whose name is composed by session *SID* string with the extension '.rs'. This file is saved in the *serverRSdir* directory located on the directory from where the OWAMP-Server is running. The *serverRSdir* directory is made every time the OWAMP-Server is started. If it exists, it will be deleted and created again.

OWAMP Session-Sender:

Usage: *OWAMP_SessionSender [-options]*

where options include:

-a | -address	*The local IP address to bind to. This address can be used on a multi-homed host for a Session-Sender that will only accept connection requests to one of its addresses. If not defined, it will by default accept connections on any/all local addresses.*
-p | -port port	*The TCP port number to be used in OWAMP-Control.*
-r | -portRange [lowport,highport]	*The UDP port range to be used in OWAMP-Test sessions.*
-n | -ntpServer	*The address of the NTP server to be used to get the time reference. If not defined, the system time will be used to get the time reference.*
-u | -updateInterval	*The value (in minutes) to be used for the time interval between two consecutive updates of the time reference.*
-? | -help	*Prints this help message.*

OWAMP Session-Receiver:

Usage: *OWAMP_SessionReceiver [-options]*

where options include:

-a | -address	*The local IP address to bind to. This address can be used on a multi-homed host for a Session-Receiver that will only accept connection requests to one of its addresses. If not defined, it will by default accept connections on any/all local addresses.*
-p | -port port	*The TCP port number to be used in OWAMP-Control.*
-r | -portRange [lowport,highport]	*The UDP port range to be used in OWAMP-Test sessions.*
-s | -sleepVal sleepVal	*The number of milliseconds to wait between the sending of two consecutive packet records, when in fetch results operation. This value is used to prevent overloading the Session Receiver's machine CPU during Fetch Sessions.*
-k | -keepResults	*If defined, the configuration and the results of the test session will be kept after the processing of a Fetch-Session requesting all the results of that test session. Otherwise the results are deleted (Used by default).*
-n | -ntpServer	*The address of the NTP server to be used to get the time reference. If not defined, the system time will be used to get the time reference.*
-u | -updateInterval	*The value (in minutes) to be used for the time interval between two consecutive updates of the time reference.*
-? | -help	*Prints this help message.*

The *sleepVal* value is used to configure the time interval between the sending of two consecutive test packet records, when the system is in fetch results operation. This value is used to prevent the overload of the machine processor (where the Session-Receiver is installed) when fetching the results.

If *keepResults* is defined the configuration and the results of the test session will be kept after the processing of a Fetch-Session requesting all the results of that test session. The configuration of the test session (Request-Session packet and the schedule slot description packets) is saved to a file whose name is composed by session *SID* string with the extension ‘.rs’. This file is saved in the *receiverDir\receiverRSdir* directory located on the directory from where the Session-Receiver is running. The results of the test session (results for each individual packet received) are saved to a file whose name is composed by the session *SID* string with the extension ‘.owp’. This file is saved in the *receiverDir\receiverTestResults* directory also located on the directory from where the Session-Receiver is running. The *receiverDir* directory is made every time the OWAMP Session-Receiver is started. If it exists, it will be deleted and created again.

If *keepResults* is not defined, the configuration and the results of the test session are deleted after the processing of a Fetch-Session that requested all the results of that test session (option used by default).

These command lines should be executed in the directory that contains the executable files.

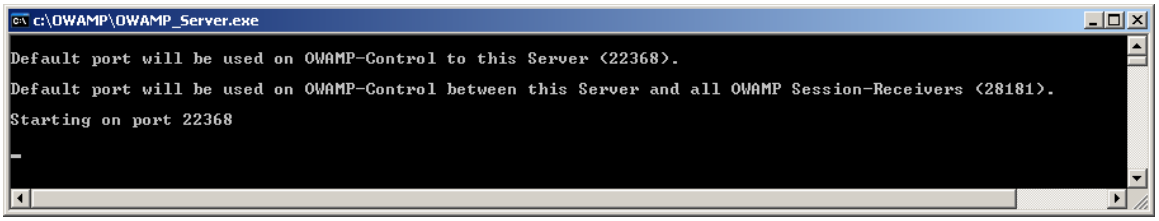
4 Configuring J-OWAMP modules

The following sub-sections show how the user can configure each module of the J-OWAMP measurement system.

4.1 OWAMP Server, OWAMP Session-Sender and OWAMP Session-Receiver

The execution of each one of these modules can be made as explained in section 3.

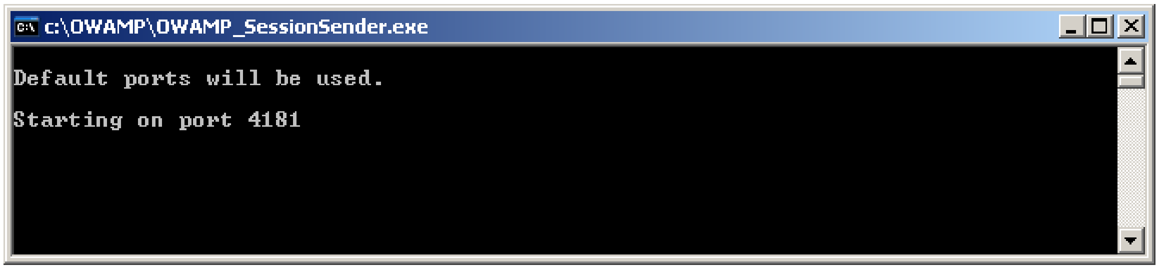
The execution of OWAMP Server executable presents the following window:



```
Default port will be used on OWAMP-Control to this Server (22368).
Default port will be used on OWAMP-Control between this Server and all OWAMP Session-Receivers (28181).
Starting on port 22368
_
```

Figure 3: Window illustrating the execution of OWAMP_Server.exe

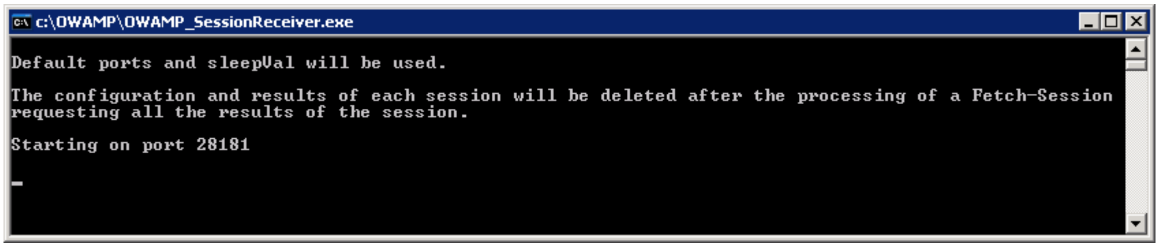
The execution of OWAMP Session-Sender executable presents the following window:



```
Default ports will be used.
Starting on port 4181
```

Figure 4: Window illustrating the execution of OWAMP_SessionSender.exe

The execution of OWAMP Session-Receiver executable presents the following window:



```
Default ports and sleepVal will be used.
The configuration and results of each session will be deleted after the processing of a Fetch-Session
requesting all the results of the session.
Starting on port 28181
_
```

Figure 5: Window illustrating the execution of OWAMP_SessionReceiver.exe

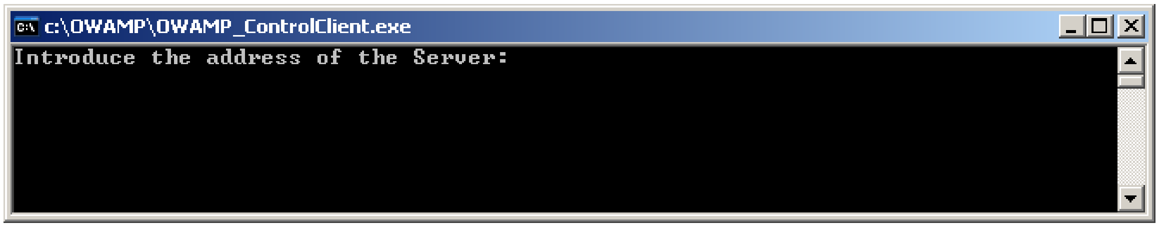
4.2 OWAMP Control-Client

This module has two modes of operation:

- Configure test sessions, show results to user and/or save them to a file;
- Configure test sessions and save test configurations and results to a MySQL database.

4.2.1 Configure test sessions, show results to user and/or save them to a file

In this mode of operation the OWAMP Control-Client is executed using the Windows executable or from the command line without any arguments. The execution of this module presents the following window:



```
Introduce the address of the Server:
```

Figure 6: Window illustrating the execution of OWAMP_ControlClient.exe

At this prompt, the user should introduce the IP address of the machine where the OWAMP Server is installed. The validation of the introduced address is made by pressing the ENTER key. The server IP address should be in the following format:

IPv4_address:TCPportNumber // for IPv4 address

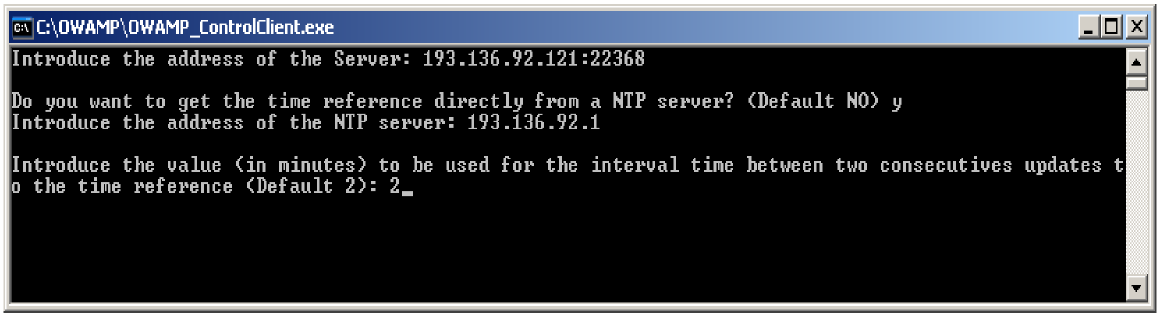
[IPv6_address]:TCPportNumber // for IPv6 address

The *TCPportNumber* is the TCP port to be used in OWAMP-Control communication between the Control-Client and the respective OWAMP-Server. The Server’s TCP port number corresponds to the port where the Server is waiting for connection requests. If the user wants to use the default *TCPportNumber* (22368), the server IP address can be in the following format:

IPv4_address // for IPv4 address

[IPv6_address] // for IPv6 address

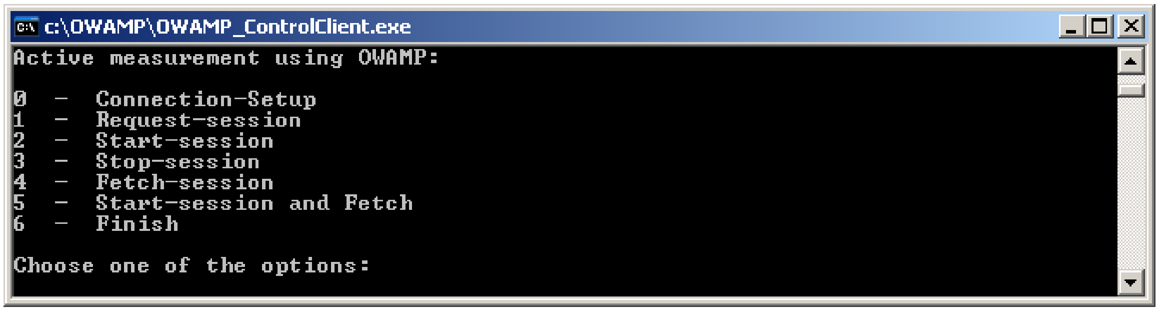
Figure 7 illustrates a possible example:



```
Introduce the address of the Server: 193.136.92.121:22368

Do you want to get the time reference directly from a NTP server? (Default NO) y
Introduce the address of the NTP server: 193.136.92.1

Introduce the value (in minutes) to be used for the interval time between two consecutives updates t
o the time reference (Default 2): 2_
```

Figure 7: Window illustrating the execution of OWAMP_ ControlClient.exe

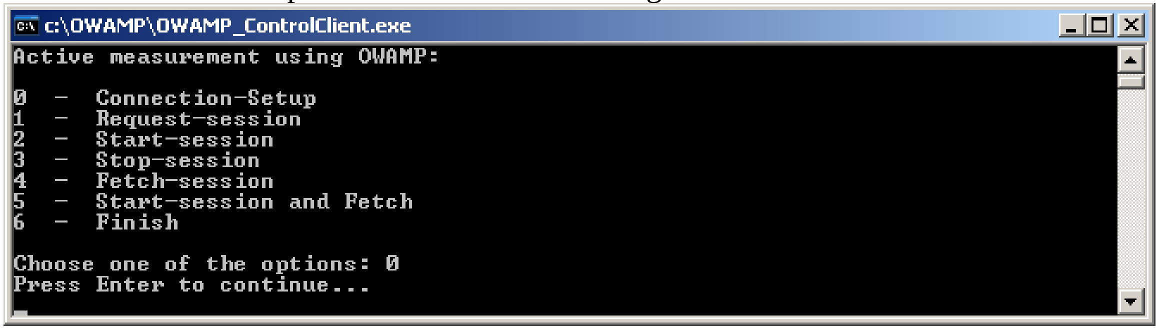
The address considered by default (pressing the ENTER key without introducing any value) is 127.0.0.1:22368 (loopback address). Then, the user has to choose whether he wants to get the time reference directly from a NTP server or not. If yes, the address of the NTP server should be introduced. If no, the system time is used to define the time reference. The time interval between two consecutive updates of the time reference is the next parameter to configure. Then, the following Menu is presented:



```
Active measurement using OWAMP:

0  -  Connection-Setup
1  -  Request-session
2  -  Start-session
3  -  Stop-session
4  -  Fetch-session
5  -  Start-session and Fetch
6  -  Finish

Choose one of the options:
```

Figure 8: Main Menu

Before requesting any test session, it is necessary to establish a connection with OWAMP Server, even if the user has already requested previous tests sessions. To request this connection, the first option to be selected is option ‘0’. Pressing the ENTER key will validate the selected option and continue the configuration.

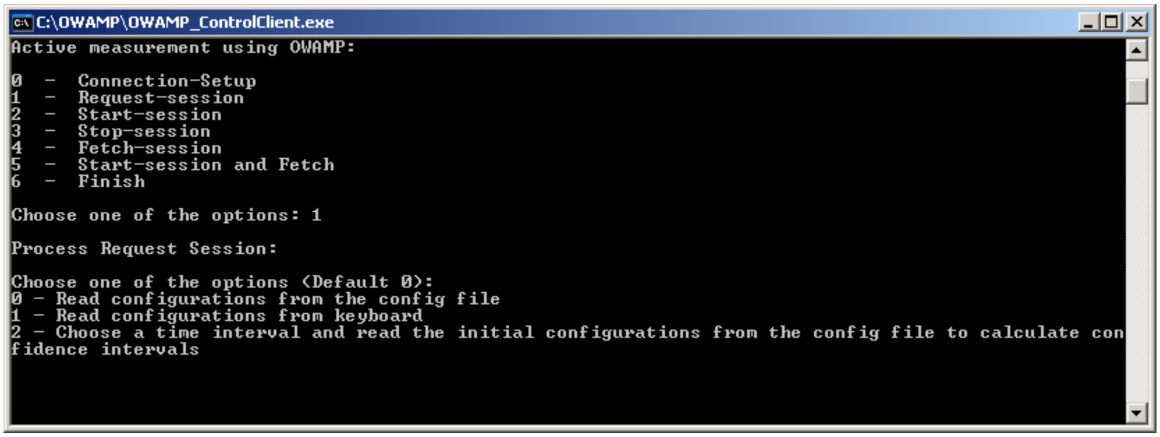


```
Active measurement using OWAMP:

0  -  Connection-Setup
1  -  Request-session
2  -  Start-session
3  -  Stop-session
4  -  Fetch-session
5  -  Start-session and Fetch
6  -  Finish

Choose one of the options: 0
Press Enter to continue...
_
```

Figure 9: Connection Setup

With J-OWAMP it is possible to accomplish two kinds of test sessions: single test sessions and confidence interval test sessions. The single test session corresponds to the configuration of one single test between two machines. The user can define the number of packets, the packet interarrivals, the packet length and the start time of the session. The results obtained with this type of session are the estimation performance metrics such as mean delay, losses, duplication and throughput. The confidence interval test session allows splitting the complete session period in a number of smaller test intervals and to perform a set of tests in each interval to enable the construction of 90% confidence intervals. For example, if the session period is set at 24 hours, the user can define 24 intervals (of one hour duration), and define also that in each interval a total of 10 tests should be performed in order to calculate the 90% confidence interval. In this case, the interval between the start of consecutive session can also be defined (say 2 minutes).

To request a test session the user should now choose option 1, resulting in the display of the following menu:



```
Active measurement using OWAMP:

0  -  Connection-Setup
1  -  Request-session
2  -  Start-session
3  -  Stop-session
4  -  Fetch-session
5  -  Start-session and Fetch
6  -  Finish

Choose one of the options: 1

Process Request Session:

Choose one of the options (Default 0):
0 - Read configurations from the config file
1 - Read configurations from keyboard
2 - Choose a time interval and read the initial configurations from the config file to calculate con
fidence intervals
```

Figure 10: Request of a test session

This menu presents three options:

- **Option 0** – to read the test configurations in the configuration file. This is the RsconfigFile.txt file that must be located in the same directory as the OWAMP-Client executable (single test sessions);
- **Option 1 –** to read the test configurations from the keyboard (single test sessions);
- **Option 2 –** to choose the time interval during which test sessions should be made. These tests will be used on the measurement of some network behaviour parameters and their respective confidence intervals (confidence interval test sessions). In this case, the initial configurations should be read from the configuration file.

The option considered by default is option ‘0’. Now, a more detailed description of each option is provided:

i) Option 0: all configuration parameters should be configured using the format specified in the configuration file, RsconfigFile.txt;

ii) Option 1: this option requires the configuration of the following parameters (a description of each parameter is also included):

Introduce the number of test packet to be sent (Default 100): number of test packets to be sent

Introduce the address of the Sender: IP address of the machine where the OWAMP Session-Sender is installed

Introduce the address of the Receiver: IP address of the machine where the OWAMP Session-Receiver is installed

The sender IP address should be in the following format:

IPv4_address:TCPportNumber // for IPv4 address
[IPv6_address]:TCPportNumber // for IPv6 address

The *TCPportNumber* is the TCP port to be used in OWAMP-Control communication between the Control-Client and the respective OWAMP-SessionSender. The sender's TCP port number corresponds to the port where the sender is waiting for connection requests. If the user wants to use the default *TCPportNumber* (4181), the sender IP address can be in the following format:

IPv4_address // for IPv4 address
[IPv6_address] // for IPv6 address

The receiver address should be in the following format:

IPv4_address // for IPv4 address
[IPv6_address] // for IPv6 address

The TCP port to use in the OWAMP-Control communication between the Server and receiver should be defined when running OWAMP_Server; otherwise the default TCP port will be used.

Note: Sender and Receiver address should have the same IP version (both IPv4 or IPv6). If one of these addresses, sender or receiver, is the loopback address the other should be loopback address too. The address considered by default (pressing the ENTER key without introducing any value) is 127.0.0.1 (loopback address). The loopback address can only be used if the Control-Client and the Server elements are running on the same host.

Introduce the padding length of test packet to be sent (Default 0): additional size, in bytes, that can be introduced in the test packets

Introduce the start time of this session: Introduction of the start time of the session.
Introduce the year (Default x): *x is the default year. Its value is the current year when configuring the session.*
Introduce the month (Default x): *x is the default month. Its value is the number representation of the current month when configuring the session (1 for January).*
Introduce the date (Default x): *x is the default day of the month. Its value is the current day of the month when configuring the session.*
Introduce the hour (Default x): *x is the default hour. Its value is the current hour when configuring the session.*
Introduce the minutes (Default x): *x is the default minute. Its value is the current minute when configuring the session.*
Introduce the seconds (Default 0): *is the seconds value to use. Its default value is zero.*

Introduce the timeout value to use for test packets (Default 10): maximum time interval during which a packet should be received. Packets that arrive to Receiver after timeout seconds are considered as lost

Introduce the number of packets Schedule slot descriptions (Default 1): This parameter defines a group of packets to be sent with the request session. Each packet represents a 'slot'. So, we have a schedule with a given number of 'slots'. Each slot has a type and a parameter. Two types are supported: exponentially distributed pseudo-random quantity (denoted by a code of 0) and a fixed quantity (denoted by a code of 1). The parameter is expressed as a timestamp and specifies a time interval. For a type 0 slot this interval is the mean value (or $1/\lambda$, if the distribution density function is expressed as $\lambda e^{(-\lambda x)}$ for positive values of *x*). For a type 1 slot, the parameter is the delay itself. The sender starts with the beginning of the schedule and executes the instructions in the slots: for a type 0 slot, it waits for an exponentially distributed time interval, with mean equal to the specified parameter, and then it sends a test packet (and proceeds to the next slot); for a type 1 slot, it waits for the specified time interval and sends a test packet (and proceeds to the next slot). The schedule is circular: when there are no more slots, the sender returns to the first slot.

This circular schedule can be configured to follow one of the following intervals:

Choose witch type of schedule do you want (Default 0):

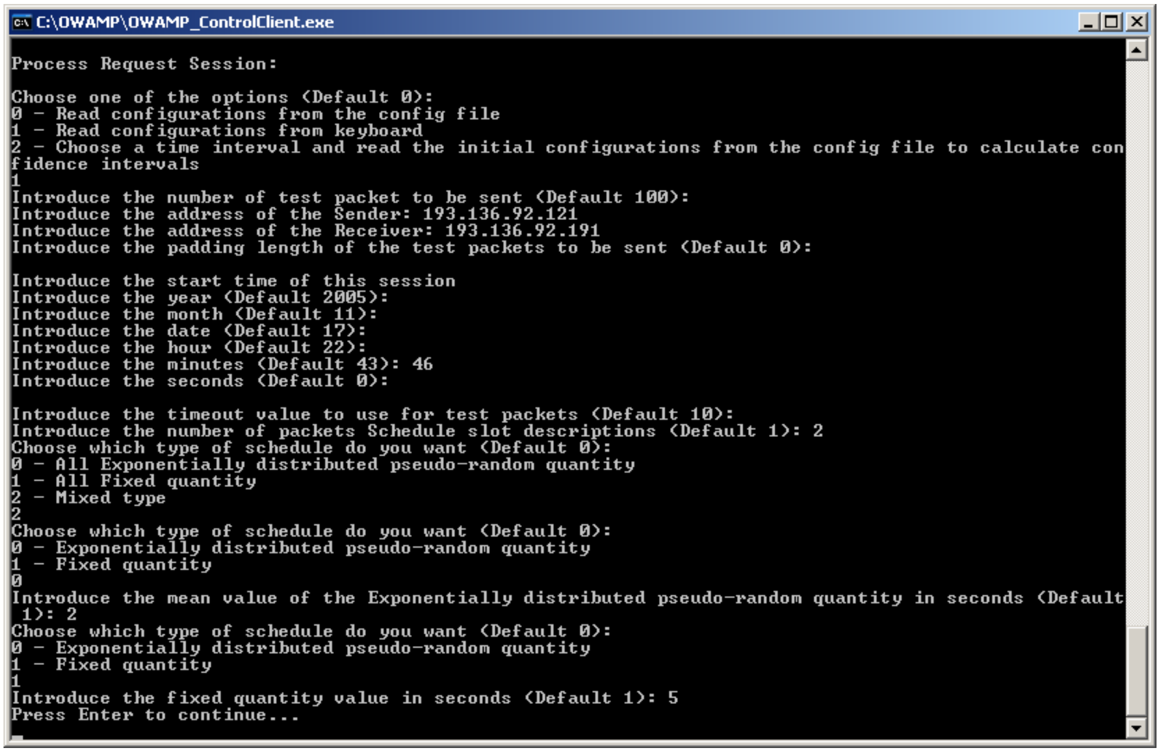
0 - All Exponentially distributed pseudo-random quantity

1 - All Fixed quantity

***2 - Mixed type* (to combine the two previous options)**

Choosing one of these options results in the following configuration steps:

- Option 0: the mean value of the exponentially distributed time interval to use should be introduced:
  Introduce the mean value of the exponentially distributed pseudo-random quantity in seconds (Default 1): to configure the mean value of the exponentially distributed time interval to use
- Option 1: the fixed time interval to use should be introduced:
  Introduce the fixed quantity value in seconds (Default 1): to configure the interval of fixed time to be used
- Option 2: proceeds to the configuration of the time intervals to use, that can be of types '0' or '1'. **Figure 11** illustrates an example corresponding to the case of an interval with exponential distribution and to the case of a fixed one:



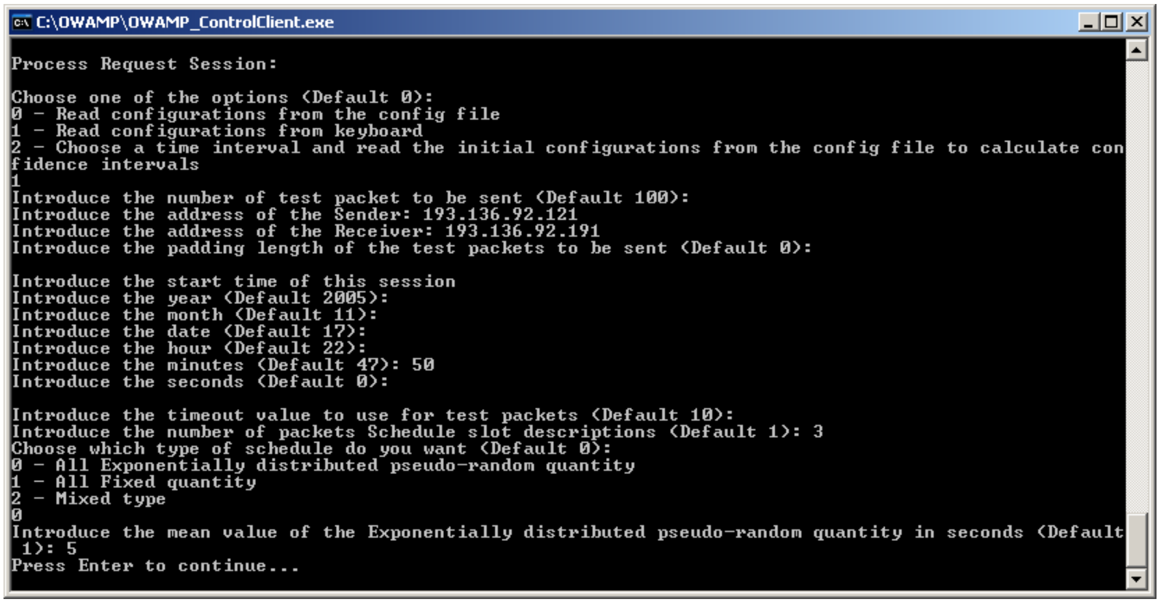
```
Process Request Session:

Choose one of the options (Default 0):
0 - Read configurations from the config file
1 - Read configurations from keyboard
2 - Choose a time interval and read the initial configurations from the config file to calculate con
fidence intervals
1
Introduce the number of test packet to be sent (Default 100):
Introduce the address of the Sender: 193.136.92.121
Introduce the address of the Receiver: 193.136.92.191
Introduce the padding length of the test packets to be sent (Default 0):

Introduce the start time of this session
Introduce the year (Default 2005):
Introduce the month (Default 11):
Introduce the date (Default 17):
Introduce the hour (Default 22):
Introduce the minutes (Default 43): 46
Introduce the seconds (Default 0):

Introduce the timeout value to use for test packets (Default 10):
Introduce the number of packets Schedule slot descriptions (Default 1): 2
Choose which type of schedule do you want (Default 0):
0 - All Exponentially distributed pseudo-random quantity
1 - All Fixed quantity
2 - Mixed type
2
Choose which type of schedule do you want (Default 0):
0 - Exponentially distributed pseudo-random quantity
1 - Fixed quantity
0
Introduce the mean value of the Exponentially distributed pseudo-random quantity in seconds (Default
 1): 2
Choose which type of schedule do you want (Default 0):
0 - Exponentially distributed pseudo-random quantity
1 - Fixed quantity
1
Introduce the fixed quantity value in seconds (Default 1): 5
Press Enter to continue...
```

Figure 11: Choosing the type of Schedule

From the group of schedule packets:

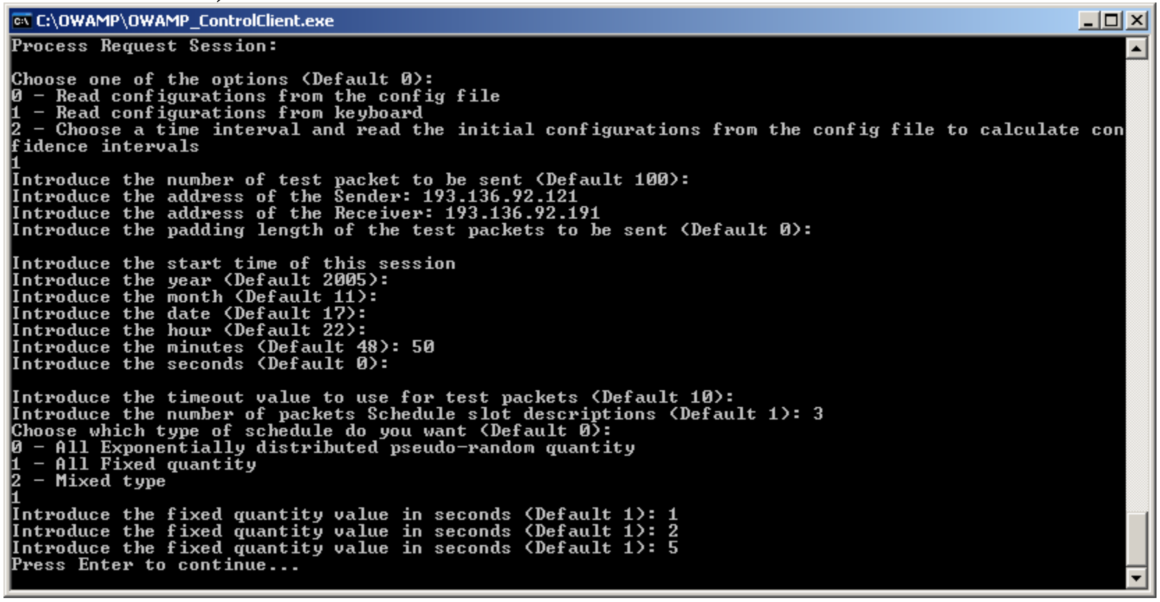
- for a fixed interval between the sending of packets, it is necessary to configure the fixed interval for each schedule packet;
- for the case of exponentially distributed intervals, it is only necessary to introduce the mean value of the time interval between sending of packets.

The following two figures illustrate two configuration examples, with a group of circular schedule with three interval values between sending of test packets, for the first two schedule types (exponential and fixed):



```
Process Request Session:

Choose one of the options (Default 0):
0 - Read configurations from the config file
1 - Read configurations from keyboard
2 - Choose a time interval and read the initial configurations from the config file to calculate con
fidence intervals
1
Introduce the number of test packet to be sent (Default 100):
Introduce the address of the Sender: 193.136.92.121
Introduce the address of the Receiver: 193.136.92.191
Introduce the padding length of the test packets to be sent (Default 0):

Introduce the start time of this session
Introduce the year (Default 2005):
Introduce the month (Default 11):
Introduce the date (Default 17):
Introduce the hour (Default 22):
Introduce the minutes (Default 47): 50
Introduce the seconds (Default 0):

Introduce the timeout value to use for test packets (Default 10):
Introduce the number of packets Schedule slot descriptions (Default 1): 3
Choose which type of schedule do you want (Default 0):
0 - All Exponentially distributed pseudo-random quantity
1 - All Fixed quantity
2 - Mixed type
0
Introduce the mean value of the Exponentially distributed pseudo-random quantity in seconds (Default
 1): 5
Press Enter to continue...
```

Figure 12: Introduction of the mean time value between the sending of packets (case of exponentially distributed intervals)



```
Process Request Session:

Choose one of the options (Default 0):
0 - Read configurations from the config file
1 - Read configurations from keyboard
2 - Choose a time interval and read the initial configurations from the config file to calculate con
fidence intervals
1
Introduce the number of test packet to be sent (Default 100):
Introduce the address of the Sender: 193.136.92.121
Introduce the address of the Receiver: 193.136.92.191
Introduce the padding length of the test packets to be sent (Default 0):

Introduce the start time of this session
Introduce the year (Default 2005):
Introduce the month (Default 11):
Introduce the date (Default 17):
Introduce the hour (Default 22):
Introduce the minutes (Default 48): 50
Introduce the seconds (Default 0):

Introduce the timeout value to use for test packets (Default 10):
Introduce the number of packets Schedule slot descriptions (Default 1): 3
Choose which type of schedule do you want (Default 0):
0 - All Exponentially distributed pseudo-random quantity
1 - All Fixed quantity
2 - Mixed type
1
Introduce the fixed quantity value in seconds (Default 1): 1
Introduce the fixed quantity value in seconds (Default 1): 2
Introduce the fixed quantity value in seconds (Default 1): 5
Press Enter to continue...
```

Figure 13: Introduction of the mean time value between sending of packets (case of fixed intervals)

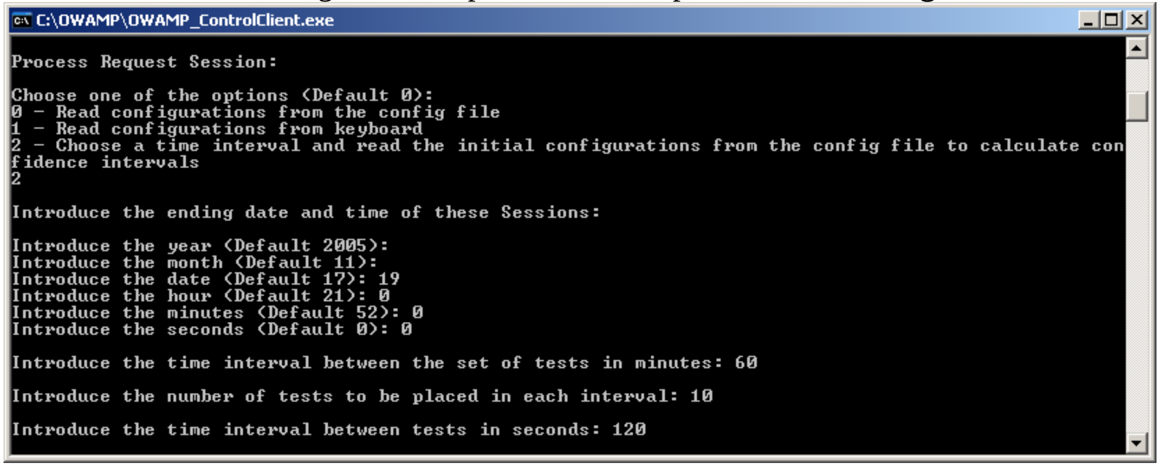
iii) Option 2: this option is used to allow the accomplishment of tests and the calculation of the 90% confidence intervals corresponding to the following parameters: mean delay, loss and throughput. In this way, in a certain time interval the user should perform *y* tests for each test session within a total of *n* OWAMP test sessions. At the end, the user will have *n* confidence intervals, each one based on *y* tests that were made. In order to obtain these statistics, it is necessary to create a configuration file with the initial configuration (mainly, the time instant for starting test sessions, that is taken as a reference to determine the beginning time instant of each one of the OWAMP test sessions to be made).

Configuration example

If we intend to study the behaviour of packet delays and losses between any two network machines, during a period of 24 hours, option 2 can be used to automatically configure all the necessary test session requests. Next, a possible test scenario will be described.

Test scenario: All tests are performed in a 24 hours period. In each hour, sets of 10 tests (including both packet delay and loss) are performed, making a total of 240 tests. In each group, the starting time instants of the tests are separated by 2 minutes. All tests lasted for 1 minute and consisted in sending 60 packets of 24 bytes each (minimum test packet size), at an average rate of 1 packet/second. In order to conveniently characterize the packet average delay and packet loss ratio, 90% confidence intervals were calculated based on the 10 average values obtained in each test belonging to a group of 10 tests.

The different configuration steps for this example are shown in **Figure 14**:



```
Process Request Session:

Choose one of the options (Default 0):
0 - Read configurations from the config file
1 - Read configurations from keyboard
2 - Choose a time interval and read the initial configurations from the config file to calculate con
fidence intervals
2

Introduce the ending date and time of these Sessions:

Introduce the year (Default 2005):
Introduce the month (Default 11):
Introduce the date (Default 17): 19
Introduce the hour (Default 21): 0
Introduce the minutes (Default 52): 0
Introduce the seconds (Default 0): 0

Introduce the time interval between the set of tests in minutes: 60

Introduce the number of tests to be placed in each interval: 10

Introduce the time interval between tests in seconds: 120
```

Figure 14: Configuration of sessions of tests for the calculation of intervals of confidence

The description of each parameter request shown in the above figure is as follows:

Process Request Session:
Choose one of the options (Default 0):
  0 - Read configurations from the config file
  1 - Read configurations from keyboard
  2 - Choose a time interval and read the initial configurations from the config file to calculate confidence intervals

2 (the user's choice)

Introduce the ending date and time of these Sessions: Introduction of the date for ending the sessions
Introduce the year (Default x): *x is the default year. Its value is the current year when configuring the session.*

Introduce the month (Default x): *x is the default month. Its value is the number representation of the current month when configuring the session (1 for January).*
Introduce the date (Default x): *x is the default day of the month. Its value is the current day of the month when configuring the session.*
Introduce the hour (Default x): *x is the default hour. Its value is the current hour when configuring the session.*
Introduce the minutes (Default x): *x is the default minutes. Its value is the current minute when configuring the session.*
Introduce the seconds (Default 0): *is the seconds value to use. Its default value is zero.*

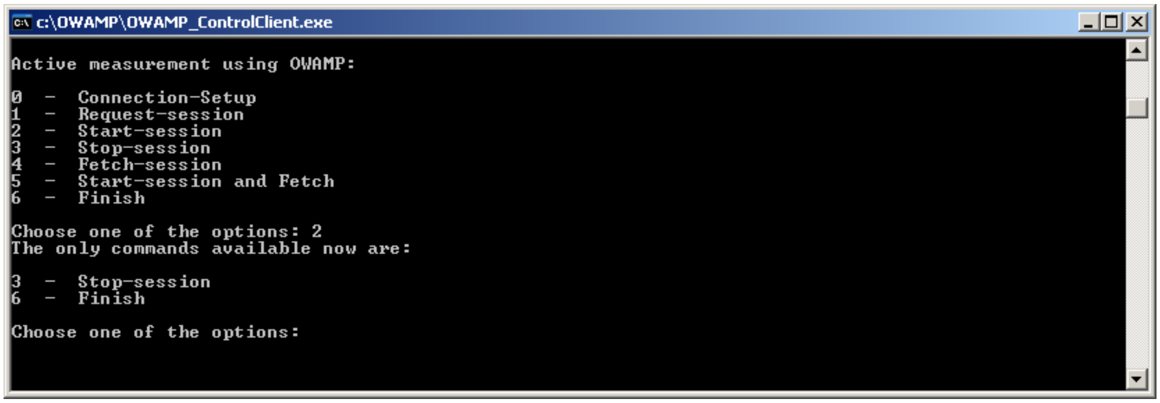
Introduce the time interval between the set of tests in minutes: *60* (time interval between groups of measurements)

Introduce the number of tests to be placed in each interval: *10* (number of test sessions to accomplish in each interval)

Introduce the time interval between tests in seconds: *120* (temporary separation between the beginning instants of each session of individual test)

After all these configurations, the number of test session requests is automatically calculated; these requests are then performed one by one.

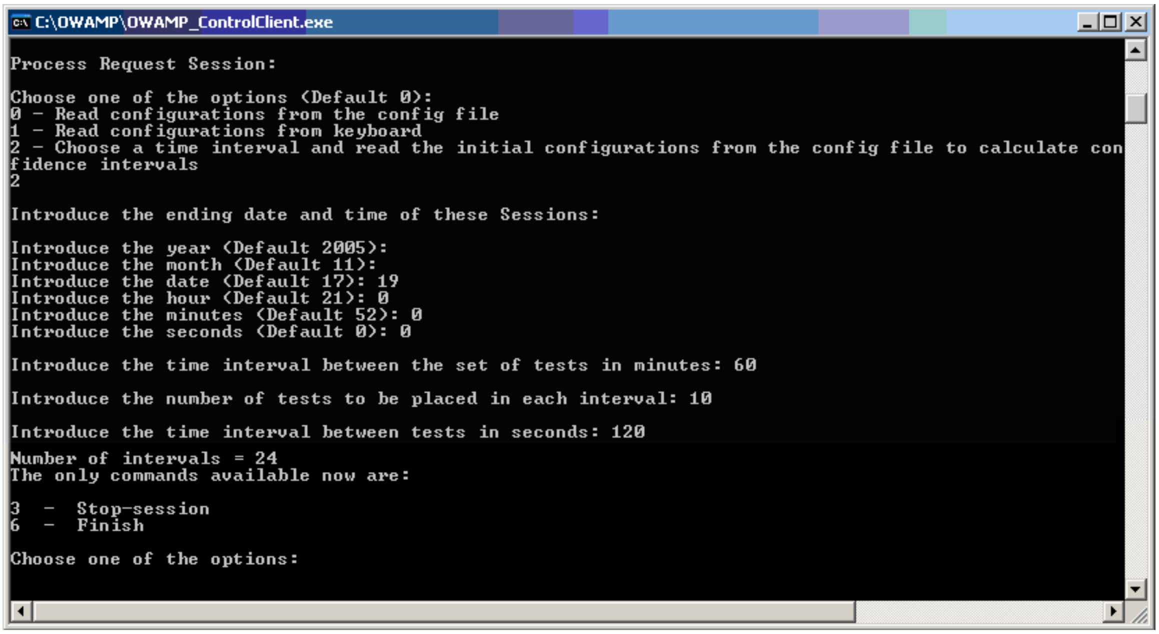
For the case of single test sessions (options 0 and 1), after configuring one or more test session requests, their execution can be started. In order to do this, option 2 should be selected from the Main Menu (**Figure 8**). All sessions are started with only one request of start sessions. After choosing this option, the following menu is presented:



```
Active measurement using OWAMP:

0  -  Connection-Setup
1  -  Request-session
2  -  Start-session
3  -  Stop-session
4  -  Fetch-session
5  -  Start-session and Fetch
6  -  Finish

Choose one of the options: 2
The only commands available now are:

3  -  Stop-session
6  -  Finish

Choose one of the options:
```

Figure 15: Menu presented during the execution of the test sessions

For the case of confidence interval test sessions (options 2), the test session is immediately started after its configuration (**Figure 16**).

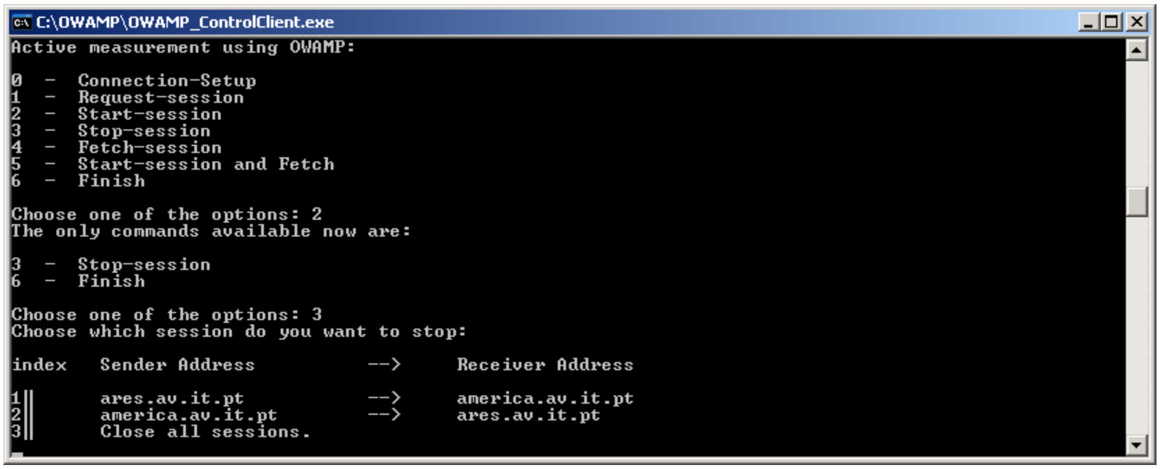


```
Process Request Session:

Choose one of the options (Default 0):
0 - Read configurations from the config file
1 - Read configurations from keyboard
2 - Choose a time interval and read the initial configurations from the config file to calculate con
fidence intervals
2

Introduce the ending date and time of these Sessions:

Introduce the year (Default 2005):
Introduce the month (Default 11):
Introduce the date (Default 17): 19
Introduce the hour (Default 21): 0
Introduce the minutes (Default 52): 0
Introduce the seconds (Default 0): 0

Introduce the time interval between the set of tests in minutes: 60

Introduce the number of tests to be placed in each interval: 10

Introduce the time interval between tests in seconds: 120
Number of intervals = 24
The only commands available now are:

3  -  Stop-session
6  -  Finish

Choose one of the options:
```

Figure 16: Configuration of sessions of tests for the calculation of intervals of confidence

A test session, either a single test or a confidence interval test session, can be finished even before its beginning or before sending all test packets. In order to do that, the Stop-Session (option 3) command is used. This is the only command that can be used after the beginning and before the end of test sessions, that is to say, during the session. To perform complete sessions, the user should wait until all Sessions are complete before sending the Stop-Sessions message. If the user intends to finish all the test sessions that are running, option 6 should be selected. In this situation, the selected option only appears on the screen after pressing the Enter key.

Detailed description of option 3: after having chosen option 3, the following menu is presented (this figure corresponds to an example of one test session conducted, in both directions, between two machines with IP addresses ares.av.it.pt/193.136.92.121 and america.av.it.pt/193.136.92.191):

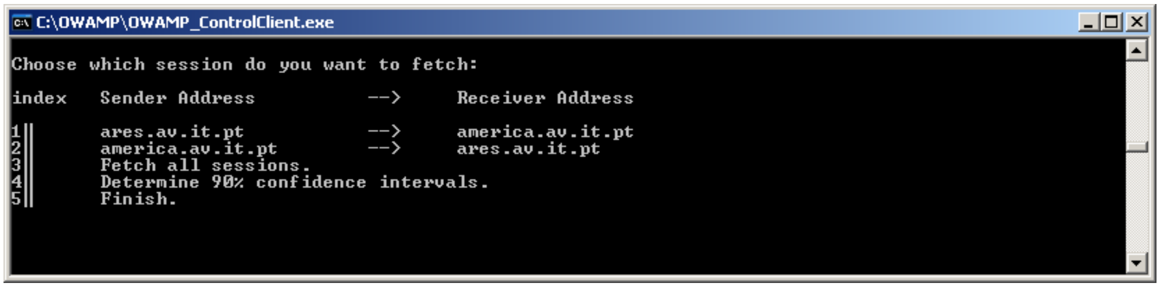


```
Active measurement using OWAMP:

0  -  Connection-Setup
1  -  Request-session
2  -  Start-session
3  -  Stop-session
4  -  Fetch-session
5  -  Start-session and Fetch
6  -  Finish

Choose one of the options: 2
The only commands available now are:

3  -  Stop-session
6  -  Finish

Choose one of the options: 3
Choose which session do you want to stop:

index   Sender Address          -->     Receiver Address

1║      ares.av.it.pt           -->     america.av.it.pt
2║      america.av.it.pt        -->     ares.av.it.pt
3║      Close all sessions.
```

Figure 17: Menu used to choose the session to be finished

Using this menu, the user can choose which test session to finish, or can finish all sessions.

After all test sessions have finished, either by user's option or because the time duration of the measurement session has expired, the following menu is presented (for the example under consideration):



```
Choose which session do you want to fetch:

index   Sender Address          -->     Receiver Address

1║      ares.av.it.pt           -->     america.av.it.pt
2║      america.av.it.pt        -->     ares.av.it.pt
3║      Fetch all sessions.
4║      Determine 90% confidence intervals.
5║      Finish.
```

Figure 18: Menu to request the results

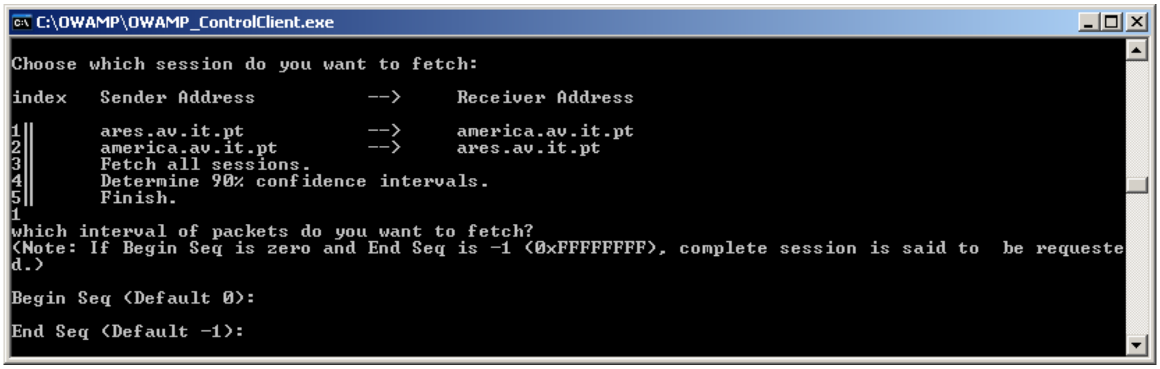
In this menu the user can select the session whose results he intends to obtain, determine the confidence intervals for a group of test sessions or he can just finish and return to the Main Menu (option Finish).

If the user only intends to obtain results corresponding to a specific session of tests, after choosing the option related to this session he has to introduce the interval of test packets whose results he intends to receive. This interval is defined in terms of the packet Sequence Number: the user has to introduce the sequence numbers of the first and the last packets that he intends to obtain.

Note: Each test packet is sent with a sequence number that begins at zero and finishes at the total number of test packets sent less 1.

If the user intends to obtain the results corresponding to all the test packets of the test session, the defined Sequence Number interval should be [0 -1].

Using the example under consideration, the following steps must be executed in order to obtain the complete results of session 1:



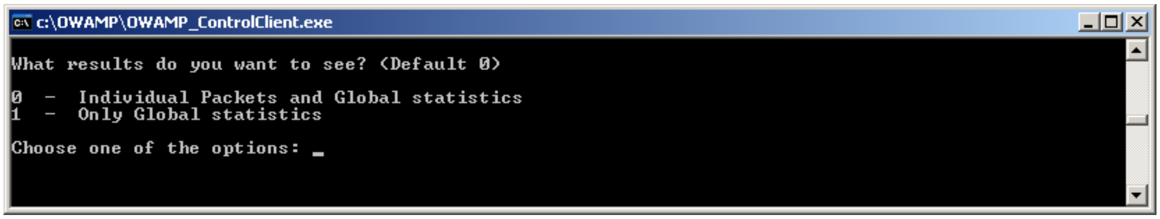
```
Choose which session do you want to fetch:

index   Sender Address          -->     Receiver Address

1║      ares.av.it.pt           -->     america.av.it.pt
2║      america.av.it.pt        -->     ares.av.it.pt
3║      Fetch all sessions.
4║      Determine 90% confidence intervals.
5║      Finish.
1
which interval of packets do you want to fetch?
(Note: If Begin Seq is zero and End Seq is -1 (0xFFFFFFFF), complete session is said to  be requeste
d.)

Begin Seq (Default 0):

End Seq (Default -1):
```

Figure 19: Menu to request the interval of results to be presented

After choosing the test packets interval, it is necessary to choose the type of results the user intends to obtain:

- Results referring to each packet and global statistics;
- Just global statistics.

Figure 20 illustrates this situation, for the example under consideration:



```
What results do you want to see? (Default 0)

0  -  Individual Packets and Global statistics
1  -  Only Global statistics

Choose one of the options: _
```

Figure 20: Menu to choice the type of results to be presented

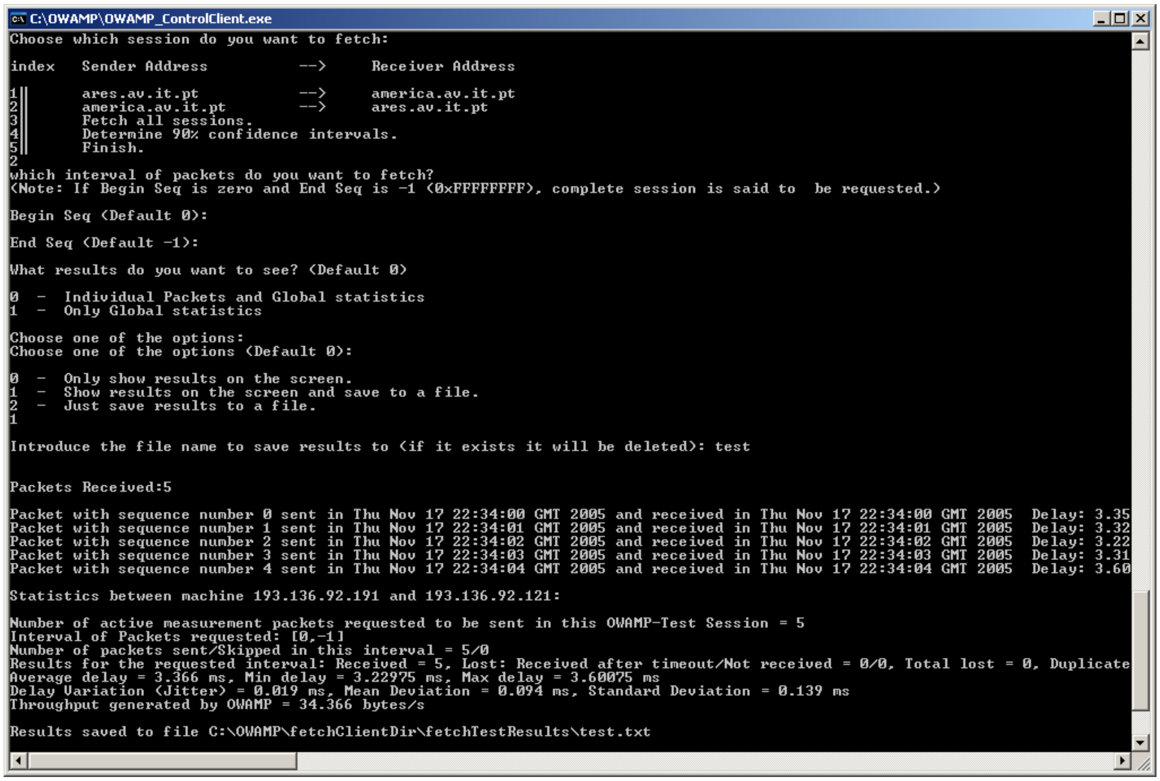
Then, it is necessary to choose one from these three options:

- To present the results on the screen;
- To present the results on the screen and save them to a file;
- Just save the results to a file.

For the last two options, the user has to introduce the name of the file where the statistics results will be saved. This file will be saved, with the *.txt* extension, in the *fetchClientDir\fetchTestResults* directory that is created in the same directory where OWAMP Control-Client is being executed.

The results of the test session (results for each individual packet record fetched) are saved to a file whose name is composed by the session *SID* string with the extension '.owp' (if it exists it will be overwrited). This file is saved in the *fetchClientDir\fetchPacketRecords* directory located on the directory from where the OWAMP Control-Client is running. If the *fetchClientDir* directory doesn't exist, it will be made.

Returning again to the example under consideration, suppose that the user asked for the results of each packet and for global statistics (presented on the screen) of the second session, also indicating a file to save the statistics results. Then, the results of executing a session of tests should look like the ones shown in **Figure 21**:



```
Choose which session do you want to fetch:

index   Sender Address          -->     Receiver Address

1║      ares.av.it.pt           -->     america.av.it.pt
2║      america.av.it.pt        -->     ares.av.it.pt
3║      Fetch all sessions.
4║      Determine 90% confidence intervals.
5║      Finish.
2
which interval of packets do you want to fetch?
(Note: If Begin Seq is zero and End Seq is -1 (0xFFFFFFFF), complete session is said to  be requested.)

Begin Seq (Default 0):

End Seq (Default -1):

What results do you want to see? (Default 0)

0  -  Individual Packets and Global statistics
1  -  Only Global statistics

Choose one of the options:
Choose one of the options (Default 0):

0  -  Only show results on the screen.
1  -  Show results on the screen and save to a file.
2  -  Just save results to a file.
1

Introduce the file name to save results to (if it exists it will be deleted): test


Packets Received:5

Packet with sequence number 0 sent in Thu Nov 17 22:34:00 GMT 2005 and received in Thu Nov 17 22:34:00 GMT 2005  Delay: 3.35
Packet with sequence number 1 sent in Thu Nov 17 22:34:01 GMT 2005 and received in Thu Nov 17 22:34:01 GMT 2005  Delay: 3.32
Packet with sequence number 2 sent in Thu Nov 17 22:34:02 GMT 2005 and received in Thu Nov 17 22:34:02 GMT 2005  Delay: 3.22
Packet with sequence number 3 sent in Thu Nov 17 22:34:03 GMT 2005 and received in Thu Nov 17 22:34:03 GMT 2005  Delay: 3.31
Packet with sequence number 4 sent in Thu Nov 17 22:34:04 GMT 2005 and received in Thu Nov 17 22:34:04 GMT 2005  Delay: 3.60

Statistics between machine 193.136.92.191 and 193.136.92.121:

Number of active measurement packets requested to be sent in this OWAMP-Test Session = 5
Interval of Packets requested: [0,-1]
Number of packets sent/Skipped in this interval = 5/0
Results for the requested interval: Received = 5, Lost: Received after timeout/Not received = 0/0, Total lost = 0, Duplicate
Average delay = 3.366 ms, Min delay = 3.22975 ms, Max delay = 3.60075 ms
Delay Variation (Jitter) = 0.019 ms, Mean Deviation = 0.094 ms, Standard Deviation = 0.139 ms
Throughput generated by OWAMP = 34.366 bytes/s

Results saved to file C:\OWAMP\fetchClientDir\fetchTestResults\test.txt
```

Figure 21: Results of a session of tests

If the user intends to obtain the 90% confidence intervals corresponding to the mean delay, loss and throughput of a group of test sessions, he has to choose the option ‘Determine 90% confidence interval’ in the request results Menu. Using this option, it is necessary to introduce the number of test sessions that will be used for the calculation of the confidence intervals. The total number of confidence intervals is calculated by dividing the total number of existent sessions by the introduced number of sessions to be used. If this division is smaller than one no interval is presented, otherwise the number of intervals corresponds to the result of the division.

In this case, there are also three possibilities for visualizing the results:

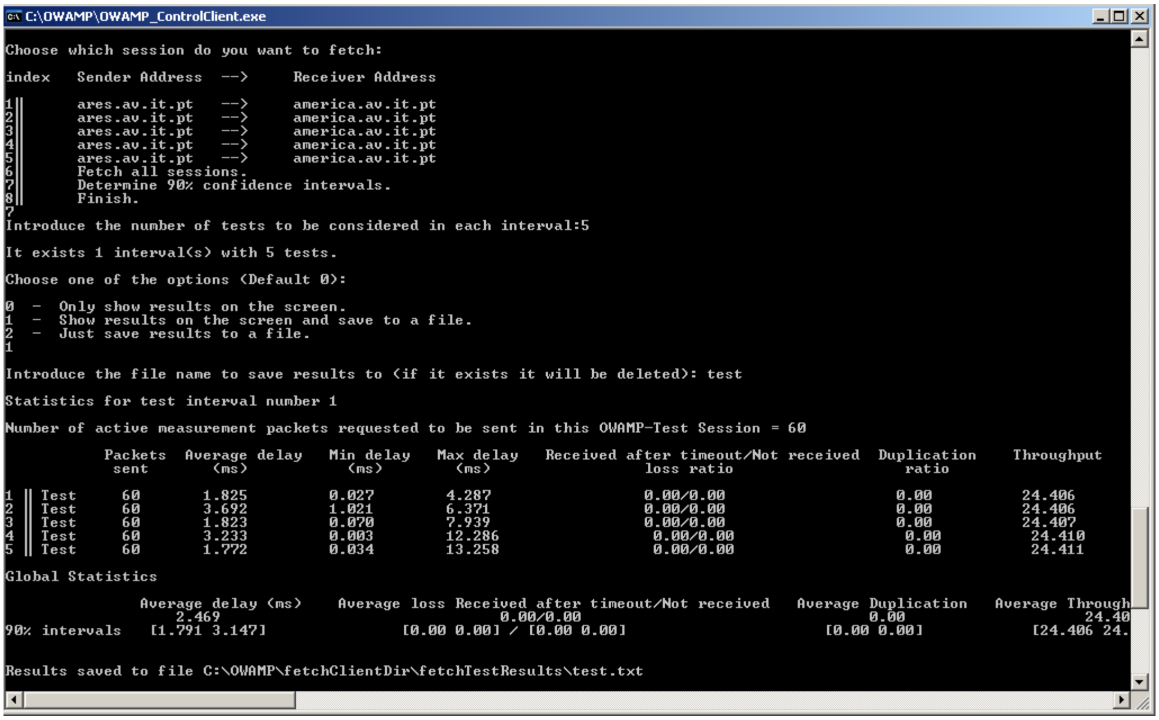
- Show the results on the screen ;
- Present the results on the screen and save them to a file;
- Just save the results to a file.

For the last two options it is necessary to introduce the name of the file where the statistics results will be saved. This file will be saved, with the *.txt* extension, in the directory *fetchClientDir\fetchTestResults* created in the same directory where OWAMP Control-Client is being executed.

The results of each test session (results for each session’s individual packet record fetched) are saved to a file whose name is composed by the session *SID* string with the extension ‘.owp’ (if it exists it will be overwrited). This file is also saved in the *fetchClientDir\fetchPacketRecords* directory located on the directory from where the

OWAMP Control-Client is running. If the *fetchClientDir* directory doesn’t exist, it will be made.

Figure 22 illustrates an example, for the case of 5 test sessions between a machine with IP address ares.av.it.pt/193.136.92.121 and a machine with IP address america.av.it.pt/193.136.92.191:



```
Choose which session do you want to fetch:

index   Sender Address  -->     Receiver Address

1║      ares.av.it.pt   -->     america.av.it.pt
2║      ares.av.it.pt   -->     america.av.it.pt
3║      ares.av.it.pt   -->     america.av.it.pt
4║      ares.av.it.pt   -->     america.av.it.pt
5║      ares.av.it.pt   -->     america.av.it.pt
6║      Fetch all sessions.
7║      Determine 90% confidence intervals.
8║      Finish.
7
Introduce the number of tests to be considered in each interval:5

It exists 1 interval(s) with 5 tests.

Choose one of the options (Default 0):

0  -  Only show results on the screen.
1  -  Show results on the screen and save to a file.
2  -  Just save results to a file.
1

Introduce the file name to save results to (if it exists it will be deleted): test

Statistics for test interval number 1

Number of active measurement packets requested to be sent in this OWAMP-Test Session = 60

           Packets  Average delay   Min delay   Max delay   Received after timeout/Not received  Duplication    Throughput
            sent       (ms)           (ms)        (ms)                 loss ratio                   ratio

1 ║ Test     60       1.825         0.027       4.287                 0.00/0.00                  0.00          24.406
2 ║ Test     60       3.692         1.021       6.371                 0.00/0.00                  0.00          24.406
3 ║ Test     60       1.823         0.070       7.939                 0.00/0.00                  0.00          24.407
4 ║ Test     60       3.233         0.003       12.286                 0.00/0.00                  0.00          24.410
5 ║ Test     60       1.772         0.034       13.258                 0.00/0.00                  0.00          24.411

Global Statistics

               Average delay (ms)    Average loss Received after timeout/Not received   Average Duplication   Average Through
                   2.469                               0.00/0.00                                0.00                    24.40
90% intervals   [1.791 3.147]               [0.00 0.00] / [0.00 0.00]                      [0.00 0.00]            [24.406 24.


Results saved to file C:\OWAMP\fetchClientDir\fetchTestResults\test.txt
```

Figure 22: Results of confidence intervals referring to 5 sessions of tests

Alternative starting of sessions:

As an alternative to option 2 (Start-Sessions) of the main Menu, the start of test sessions can be made through option 5 (Start-Sessions and Fetch). This option allows the user to begin a session and to obtain its results automatically, as soon as it ends, without having to go to the menu that is used to select the session for which the user wants to obtain the results. However, this option can only be chosen if the user has made only one request of sessions of tests.

After the end of a session of tests and before the request of another session, option 4 of the Main Menu can be used to return to the menu that is used to choose the test session for which the user intends to visualize the results.

Before requesting to start a test session (options 2 and 5 of the Main Menu), the user can conclude the execution of the OWAMP Control-Client application by using option 6 of the Main Menu. In this situation, all requests of test sessions made before this time instant are invalidated.

4.2.2 Configure test sessions and save test configurations and results to a MySQL database

In this mode of operation the OWAMP Control-Client module is used to configure a test session whose configurations and results must be saved to a MySQL database.

In order to configure test sessions and save its configurations and results to a MySQL database, the OWAMP Control-Client should be executed in the command line using the following syntax:

For single test sessions:

Usage: *OWAMP_ControlClient <userID> <serverAddress> <confSender> <confReceiver> <numberOfScheduleSlots> <numberOfPackets> <senderPort> <receiverPort> <senderAddress> <receiverAddress> <paddingLength> <startYear> <startMonth> <startDate> <startHour> <startMinute> <startSecond> <timeout> <typePDescriptor_Type> <typePDescriptor_Value> <slotType> <slotParameter> <DB_HOSTNAME> <DB_NAME> <DB_USERNAME> <DB_PASSWORD> < SSconfigTable > < ntpServerAddress> < updateInterval>*

For Confidence Interval test sessions (CI session):

Usage: *OWAMP_ControlClient <userID> <serverAddress> <confSender> <confReceiver> <numberOfScheduleSlots> <numberOfPackets> <senderPort> <receiverPort> <senderAddress> <receiverAddress> <paddingLength> <startYear> <startMonth> <startDate> <startHour> <startMinute> <startSecond> <timeout> <typePDescriptor_Type> <typePDescriptor_Value> <slotType> <slotParameter> <DB_HOSTNAME> <DB_NAME> <DB_USERNAME> <DB_PASSWORD> <CIconfigTable> <CIsidTable> <timeBetweenSetOfTests> <timeBetweenTests> <numIntervals> <numOfTestsInInterval> < ntpServerAddress> < updateInterval>*

Note: A single test session is used to calculate the average delay and loss of x packets sent from host A to host B. A CI session is used to calculate 90% confidence intervals based on the y average values obtained in each test belonging to a group of y tests.

The *userID* is the identification of the user who is configuring this test session. It is the IP address of the user's machine.

The *serverAddress* is the IP address of the OWAMP Server to be used in this test session. The server IP address should be in the following format:

IPv4_address:TCPportNumber // for IPv4 address

[IPv6_address]:TCPportNumber // for IPv6 address

The *TCPportNumber* is the TCP port to use in the OWAMP-Control communication between the Control-Client and the respective OWAMP-Server. The Server's TCP port number corresponds to the port where the Server is waiting for connection requests. If the user wants to use the default *TCPportNumber* (22368), the server IP address can be in the following format:

IPv4_address // for IPv4 address

[IPv6_address] // for IPv6 address

ConfSender and c*onfReceiver* MUST be set to 0 or 1 by the client. The server MUST interpret any non-zero value as 1. If the value is 1, the server is being asked to configure the corresponding agent (sender or receiver). In this case, the corresponding Port value should be disregarded by the server. At least one of *confSender* and *confReceiver* must be 1. (Both can be set, in which case the server is being asked to perform a session between two hosts it can configure.)

The *numberOfScheduleSlots* (Number of Schedule Slots) specifies the number of slot records. It is used by the sender to determine when to send test packets.

The *numberOfPackets* (Number of Packets) is the number of active measurement packets to be sent during this OWAMP-Test session.

If Conf-Sender is not set, *senderPort* (Sender Port) is the UDP port from which OWAMP-Test packets will be sent. If Conf-Receiver is not set, *receiverPort* (Receiver Port) is the UDP port of the OWAMP-Test to which packets are requested to be sent. But, if *senderPort* or *receiverPort* are zero a free port will be used to send or receive packets, respectively.

The *senderAddress* (Sender Address) and *receiverAddress* (Receiver Address) fields contain, respectively, the sender and receiver addresses of the end points of the Internet path over which an OWAMP test session is requested. The sender IP address should be in the following format:

```
IPv4_address:TCPportNumber       // for IPv4 address
[IPv6_address]:TCPportNumber     // for IPv6 address
```

The *TCPportNumber* is the TCP port to use in the OWAMP-Control communication between the Control-Client and the respective OWAMP-SessionSender. The sender's TCP port number corresponds to the port where the sender is waiting for connection requests. If the user wants to use the default *TCPportNumber* (4181), the sender IP address can be in the following format:

```
IPv4_address       // for IPv4 address
[IPv6_address]     // for IPv6 address
```

The receiver address should be in the following format:

```
IPv4_address       // for IPv4 address
[IPv6_address]     // for IPv6 address
```

The TCP port to use in the OWAMP-Control communication between the Server and receiver should be defined when running OWAMP_Server, or the default TCP port will be used.

Note: Sender and Receiver address should have the same IP version (both IPv4 or IPv6). If one of these addresses, sender or receiver, is the loopback address the other should be loopback address too. The address considered by default (pressing the ENTER key without introducing any value) is 127.0.0.1 (loopback address). The loopback address can only be used if the Server and the Control-Client elements are running on the same host.

The *paddingLength* (Padding length) is the number of octets to be appended to the normal OWAMP-Test packet.

The *startYear, startMonth, startDate, startHour, startMinute*, and *startSecond* fields represent the start time, which is the time instant when the session will be started. Sessions with a start time located in the past and separated from the present by more than *timeOut* are rejected.

The *timeOut* (Timeout or a loss threshold) is time interval (in seconds). A packet belonging to the current test session will be considered lost if it is not received during Timeout seconds after being sent.

The *typePDescriptor_Type* and *typePDescriptor_Value* fields represent the Type-P Descriptor which covers only a subset of (very large) Type-P space. If the first two bits of the Type-P Descriptor are 00, then subsequent six bits specify the requested Differentiated Services Codepoint (DSCP) value of the sent OWAMP-Test packets, as defined in RFC 2474. If the first two bits of Type-P descriptor are 01, then the subsequent 16 bits specify the requested PHB Identification Code (PHB ID), as defined in RFC 2836. Therefore, a value of all zeros specifies the default best-effort service.

The sender and the receiver both need to know the same sending schedule. In this way, when packets are lost the receiver knows when they were supposed to be sent. It is desirable to compress common schedules and still to be able to use an arbitrary one for the test sessions. In many cases, the schedule will consist of repeated sequences of packets: in this way, the sequence performs some test and the test is repeated a certain number of times in order to gather statistics. To implement this, we have a schedule with a given number of slots (*numberOfScheduleSlots*). Each slot has a type (*slotType*) and a parameter (*slotParameter*). Two types are supported: exponentially distributed pseudo-random quantity (denoted by a code of 0) and fixed quantity (denoted by a code of 1). The parameter is expressed in seconds and specifies a time interval. For a type 0 slot (exponentially distributed pseudo-random quantity) this interval is the mean value (or 1/lambda, if the distribution density function is expressed as lambda*exp(-lambda*x) for positive values of x). For a type 1 (fixed quantity) slot, the parameter is the delay itself. The sender starts with the beginning of the schedule and executes the instructions in the slots: for a slot of type 0, wait for an exponentially distributed time with mean equal to the specified parameter and then send a test packet (and proceed to the next slot); for a slot of type 1, wait for the specified time and send a test packet (and proceed to the next slot). The schedule is circular: when there are no more slots, the sender returns to the first slot.

The *DB_HOSTNAME* represents the IP address of the database server.

The *DB_NAME* represents the database where configurations and results of all sessions will be saved.

The *DB_USERNAME* and *DB_PASSWORD* are the username and password to be used in the connection to the database server.

The *SSconfigTable* is the name of the table to be used to save the Single Session's configurations. If this table doesn't exist in the database, it will be created.

The *CIconfigTable* is the name of the table to be used to save the Confidence Interval Session's configurations. If this table doesn't exist in the database, it will be created.

The *CIsidTable* is the name of the table to be used to save the *SID* (session identifier), *senderPort*, *receiverPort*, real start time and total test duration for each one of the individual single sessions of each interval of a Confidence Interval session. If this table doesn't exist in the database, it will be created.

Note: For the Confidence Interval session we have to use one table (*CIconfigTable*) to save the initial session's configuration but we also need another table to save the *SID* (session identifier), *senderPort*, *receiverPort*, real start time and total test duration for each individual single session of the CI session. This is so because each SS has a different *SID* and may have a different test duration if slot type zero is used and a different sender UDP test port if *senderPort* = 0 (a free UDP port will be used to send test packets). We always use *confReceiver* = 1 and *confSender* = 0, so *receiverPort* may be different for each SS test session.

The *timeBetweenSetOfTests* represents the time interval (in minutes) between a set of tests in CI sessions and the *timeBetweenTests* represents the time interval (in seconds) between each test session of a set of sessions belonging to an interval.

The *numIntervals* is the number of intervals of a Confidence Interval session and the *numOfTestsInInterval* is the number of tests in each interval.

The *ntpServerAddress* is the address of the NTP server used to get the time reference. The *updateInterval* is the value (in minutes) to use for the time interval between two consecutive updates of the time reference. The *ntpServerAddress* and *updateInteval* are optional but, when defined, they should be simultaneously defined. If they are not defined, the system time will be used to get the time reference and the update interval will have the default value of 2 minutes.

Using this mode of operation, the results of each individual test session are saved in a table which name is the *SID* (session identifier) string representation of the session. This table will be created in the *DB_NAME* database used for the respective test session. The Skip Range description information is saved in the table *SIDSRD*, where *SID* is the session identifier.

The table *SessionDescriptionRecords* is also created if it doesn't exist. This table is used to save the Session Description Record information [*SID*, *nextSeqno* (next sequence number that would have been sent from the send session), *numberOfSkipRanges* (number of skipped ranges) and *numberOfRecords* (number of packet records fetched from the Session-Receiver after the end of a test session)] of all fetched test sessions.

These last three tables are created using the Fetch-Session element. Please refer to section 4.3.2 for more information.

This mode of operation is used in the Web interface for J-OWAMP. This interface was developed to have a friendly user interface to our implementation of OWAMP.

4.3 OWAMP Fetch-Client

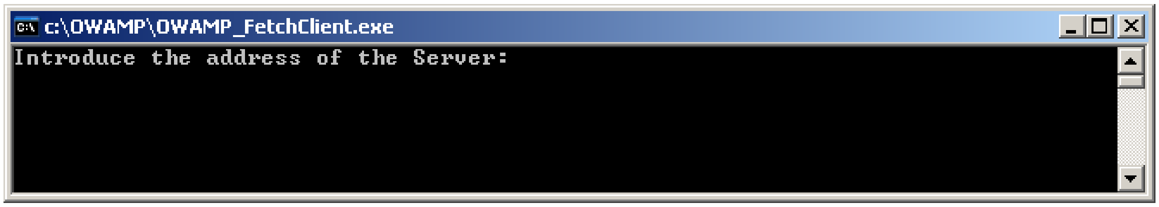
This module allows the user to request for the results of test sessions that are in progress or have successfully finished. For test sessions that are still in progress, the request of the complete results is not allowed. This request is rejected by the OWAMP Session-Receiver. The advantage of using the OWAMP Fetch-Client is to have the possibility of requesting results of test sessions that are still in progress, something that is not possible to do using OWAMP Control-Client (between a request for starting a session of tests and its end the only allowed command is a premature stop of the session - option 3, Stop sessions).

This module has two modes of operations:

- Fetch the results of a session and show them to user and/or save them to a file;
- Fetch the results of a session and save them to a MySQL database.

4.3.1 Fetch the results of a session and show them to user and/or save them to a file

In this mode the OWAMP Fetch-Client is executed using the Windows executable or from the command line without any arguments. The execution of the OWAMP Fetch-Client module presents the following window:



```
Introduce the address of the Server:
```

Figure 23: Window of execution of OWAMP_FetchClient.exe.

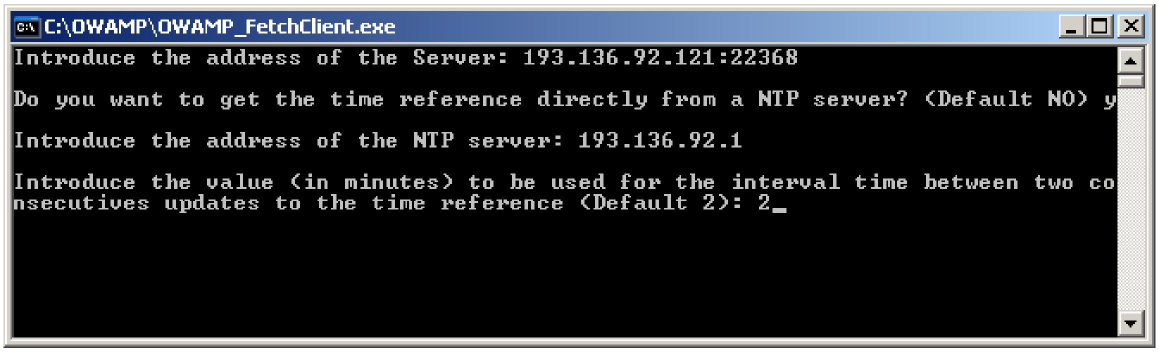
In this line the IP address of the same Server that was used to configure the sessions should be introduced, followed by the ENTER key. The server IP address should be in the following format:

IPv4_address:TCPportNumber // for IPv4 address
[IPv6_address]:TCPportNumber // for IPv6 address

The *TCPportNumber* is the TCP port to use in the OWAMP-Control communication between the Fetch-Client and the respective OWAMP-Server. The Server’s TCP port number corresponds to the port where the Server is waiting for connection requests. If the user wants to use the default *TCPportNumber* (22368), the server IP address can be in the following format:

IPv4_address // for IPv4 address
[IPv6_address] // for IPv6 address

Figure 24 illustrates a possible example:

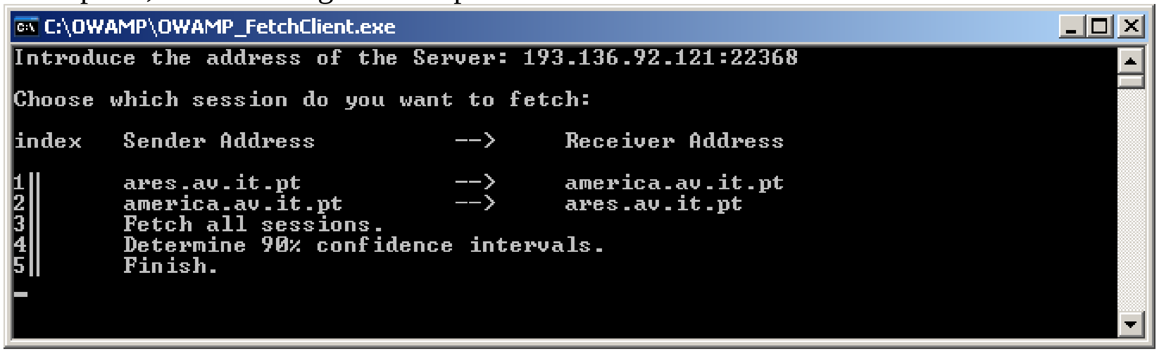


```
Introduce the address of the Server: 193.136.92.121:22368

Do you want to get the time reference directly from a NTP server? (Default NO) y

Introduce the address of the NTP server: 193.136.92.1

Introduce the value (in minutes) to be used for the interval time between two co
nsecutives updates to the time reference (Default 2): 2_
```

Figure 24: Window of execution of OWAMP_FetchClient.exe

The address considered by default (pressing the ENTER key without introducing any value) is 127.0.0.1:22368 (loopback address). Then, the user has to choose if he wants to get the time reference directly from a NTP server or not. If yes, the address of the NTP server should be introduced. If no, the system time is used to define the time reference. The time interval between two consecutives updates of the time reference is the next parameter to configure.

Then, considering the same example that was used in the OWAMP Control-Client description, the following Menu is presented:



```
Introduce the address of the Server: 193.136.92.121:22368

Choose which session do you want to fetch:

index   Sender Address          -->     Receiver Address

1║      ares.av.it.pt           -->     america.av.it.pt
2║      america.av.it.pt        -->     ares.av.it.pt
3║      Fetch all sessions.
4║      Determine 90% confidence intervals.
5║      Finish.
_
```

Figure 25: Menu used to request the results

In this way, it is possible to obtain at any instant the results of any session that is still in progress or that has already finished. The procedure, concerning the Menu that is used to request the results, is the same as the one presented in the OWAMP Control-Client's description (**Figure 18**).

4.3.2 Fetch the results of a session and save them to a MySQL database

In this mode of operation the OWAMP Fetch-Client fetches the results of a session and saves them to a database. If the session isn't finished yet, all available results are periodically fetched until the end of the test session. So a fetch session instance for a test session fetches all the results of that session. In this way, if a previous Fetch-Session was executed for a given test session, a new Fetch-Session will not be processed.

In order to fetch the results of a test session and save them to a MySQL database the OWAMP_FetchClient should be executed in the command line using the following syntax:

Usage: *OWAMP_FetchClient <serverAddress> <DB_HOSTNAME> <DB_NAME> <DB_USERNAME> <DB_PASSWORD> <SID> <numberOfPackets> < slotParameter> <startTime> <totalTimeToWait> < ntpServerAddress> < updateInterval>*

The *serverAddress* is the IP address of the OWAMP-Server to be used in the Fetch-Session. It should be in the same format specified in section 4.3.1.

The *DB_HOSTNAME* represents the IP address of the database server.

The *DB_NAME* represents the database where the results of the test session will be saved.

The *DB_USERNAME* and *DB_PASSWORD* are the username and password to be used in the connection to the database server.

The *SID* is the string representation of the session identifier of the test session whose results the user wants to fetch.

The *numberOfPackets* and the *slotParameter* are the number of test packets to be sent and the slot parameter configured for the corresponding test session.

The *startTime* is the time in milliseconds since 0h 1 Jan 1970, corresponding to the start time of the session to be fetched.

The *totalTimeToWait* is the total time (in milliseconds) duration of the corresponding test session.

The *ntpServerAddress* is the address of the NTP server used to get the time reference. The *updateInterval* is the value (in minutes) to use for the time interval between two consecutive updates of the time reference. The *ntpServerAddress* and *updateInteval* are optional but, when defined, they should be simultaneously defined. If they are not defined, the system time will be used to get the time reference and the update interval will have the default value of 2 minutes.

Using this mode of operation, the results of each individual test session are saved in a table which name is the *SID* (session identifier) string representation of the session. This table will be created in the *DB_NAME* database used for the respective test session. This table is created using the following SQL statement:

*"CREATE TABLE SID (id int NOT NULL AUTO_INCREMENT, sequenceNumber int, sendTimestamp double, sendError_s_z_scale int, sendError_multiplier int, receiveTimestamp double, receiveError_s_z_scale int, receiveError_multiplier int, sentTime text /*String representation of the packet send time.*/, receivedTime text /*String representation of the packet receive time.*/, delay double /*Packet delay in milliseconds*/, PRIMARY KEY (id))"*

The Skip Range description information is saved in the table *SIDSRD*, where *SID* is the session identifier. This table is created using the following SQL statement:

"CREATE TABLE SIDSRD (id int NOT NULL AUTO_INCREMENT, firstSeqnoSkipped int, lastSeqnoSkipped int, PRIMARY KEY (id))"

Also the table *SessionDescriptionRecords* is created if it doesn't exist. This table is used to save the Session Description Record information [*SID*, *nextSeqno* (next sequence number that would have been sent from the send session), *numberOfSkipRanges* (number of skipped ranges) and *numberOfRecords* (number of packet records fetched from the Session-Receiver after the end of a test session)] of all fetched test sessions. Each row of this table represents a fetched session and is identified by the session *SID*. The Session Description Record information of a session only should be saved in this table when the Fetch-Session is complete. So, using this table we can determine if the Fetch-Session of a test session is finished or not. This table is created using the following SQL statement:

"CREATE TABLE SessionDescriptionRecords (sid VARCHAR(32) NOT NULL, nextSeqno int, numberOfSkipRanges int, numberOfRecords int, PRIMARY KEY (sid))"

5 CalculateCI

This module is an auxiliary module used in the calculation of the 90% confidence intervals corresponding to the following parameters: mean delay, loss and throughput of a Confidence Interval test session. In this way, in a certain time interval the user should perform *y* tests for each test session within a total of *n* OWAMP test sessions. At the end, the user will have *n* confidence intervals, each one based on *y* tests that were made. These tests should be made using the configuration of section 4.2.2.

In order to calculate the 90% confidence intervals statistics of a CI session and save them to a MySQL database, the CalculateCI module should be executed in the command line using the following syntax:

Usage: *CalculateCI <serverAddress> <DB_HOSTNAME> <DB_NAME> <DB_USERNAME> <DB_PASSWORD> <CIconfigTable> <CIsidTable> <CI_id> <CI_ResultTable> <SS_ResultTable> < ntpServerAddress> < updateInterval>*

The *serverAddress* is the IP address of the OWAMP-Server to be used in the Fetch-Session. The server IP address should be in the following format:

IPv4_address:TCPportNumber // for IPv4 address

[IPv6_address]:TCPportNumber // for IPv6 address

The *TCPportNumber* is the TCP port to use in the OWAMP-Control communication between the Fetch-Client and the respective OWAMP-Server. The Server's TCP port number corresponds to the port where the Server is waiting for connection requests. If the user wants to use the default *TCPportNumber* (22368), the server IP address can be in the following format:

IPv4_address // for IPv4 address

[IPv6_address] // for IPv6 address

Note: If an individual test session of an interval isn't finished yet, this module waits until the end of the session and process the Fetch-Session (using mode 4.3.2 of operation) for the corresponding session. To do so, the OWAMP-Server address and port to use are needed.

The *DB_HOSTNAME* represents the IP address of the database server.

The *DB_NAME* represents the database where the results of the test session will be saved.

The *DB_USERNAME* and *DB_PASSWORD* are the username and password to be used in the connection to the database server.

The *CIconfigTable* is the name of the table where the Confidence Interval Session's configurations are saved (see section 4.2.2).

The *CIsidTable* is the name of the table where the *SID* (session identifier), *senderPort, receiverPort*, real start time and total test duration for each one of the individual single sessions belonging to each interval of a Confidence Interval session are saved (see section 4.2.2).

The *CI_id* is the row number of the entry of table *CIconfigTable* corresponding to this CI session.

The *CI_ResultTable* is the table to be used to save the statistic results of each interval of the CI session. This table is created using the following SQL statement:

" CREATE TABLE CI_ResultTable (id int not null AUTO_INCREMENT, averageDelay DOUBLE, varDelay DOUBLE, minIntervalDelay DOUBLE, maxIntervalDelay DOUBLE, averageTimeoutLosses DOUBLE, varTimeoutLosses DOUBLE, minIntervalTimeoutLosses DOUBLE, maxIntervalTimeoutLosses DOUBLE, averageNotReceivedLosses DOUBLE, varNotReceivedLosses DOUBLE, minIntervalNotReceivedLosses DOUBLE, maxIntervalNotReceivedLosses DOUBLE, averageNumDuplicates DOUBLE, varNumDuplicates DOUBLE, minIntervalNumDuplicates DOUBLE, maxIntervalNumDuplicates DOUBLE, averageThroughput DOUBLE, varThroughput DOUBLE, minIntervalThroughput DOUBLE, maxIntervalThroughput DOUBLE, PRIMARY KEY (id))"

The *SS_ResultTable* is the table where the statistic results of each individual Single Sessions will be saved. These statistic results are: number of packets sent, number of packets received, minimum, maximum and average delay, delay variance, packets loss by timeout ratio, packets not received ratio, duplication ratio and throughput generated by OWAMP. This table is created using the following SQL statement:

" CREATE TABLE SS_ResultTable (id INT NOT NULL AUTO_INCREMENT, SID VARCHAR(32), numOfPacketsSent INT, numPacketsReceived INT, minDelay DOUBLE, averageDelay DOUBLE, maxDelay DOUBLE, varDelay DOUBLE, numTimeoutLosses DOUBLE, numNotReceivedLosses DOUBLE, numDuplicates DOUBLE, throughput DOUBLE, PRIMARY KEY (id))"

The *ntpServerAddress* is the address of the NTP server used to get the time reference. The *updateInterval* is the value (in minutes) to use for the time interval between two consecutive updates of the time reference. The *ntpServerAddress* and *updateInteval* are optional but, when defined, they should be simultaneously defined. If they are not defined, the system time will be used to get the time reference and the update interval will have the default value of 2 minutes.