

# J-OWAMP

---

## Java Implementation of OWAMP

### User's Manual

**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

## Contents

1	Introduction and requirements .....	3
2	System architecture .....	4
3	Running J-OWAMP .....	6
4	Configuring J-OWAMP modules .....	7
4.1	OWAMP Server, OWAMP Session-Sender and OWAMP Session-Receiver...	7
4.2	OWAMP Control-Client .....	8
4.3	OWAMP Fetch-Client .....	19

## 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 J-OWAMP system can be used both in Windows and Linux platforms, requiring only the installation of J2SE Java Runtime Environment (JRE), available in both Windows and Linux versions. The installation of this module is mandatory, in order to allow the execution of Java applications.

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.

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 and section 4 describes in detail all the necessary steps to conveniently configure the different system modules.

## 2 System architecture

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

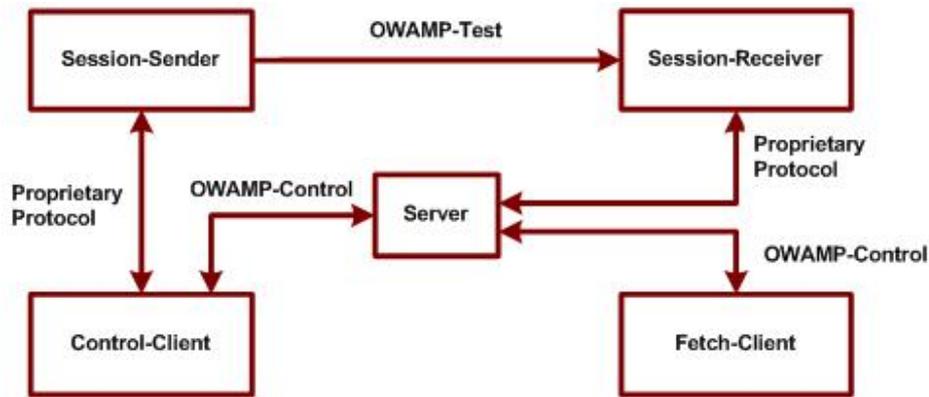


Figure 1: OWAMP architecture

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 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**).

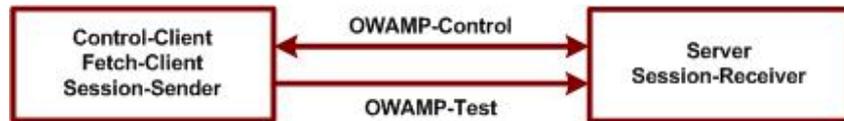


Figure 2: OWAMP simplified architecture

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 charged with 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:

<b>J-OWAMP element</b>	<b>Port number</b>
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:

*<name of the executable> <port to be used in the TCP communications >*

This command line should be executed in the directory that contains the system executable files.

The only exception of this syntax regards the OWAMP Session-Receiver, that can be executed in the command line using the following format:

*<name of the executable> <port to be used in the TCP communications > <sleep val in ms>*

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.

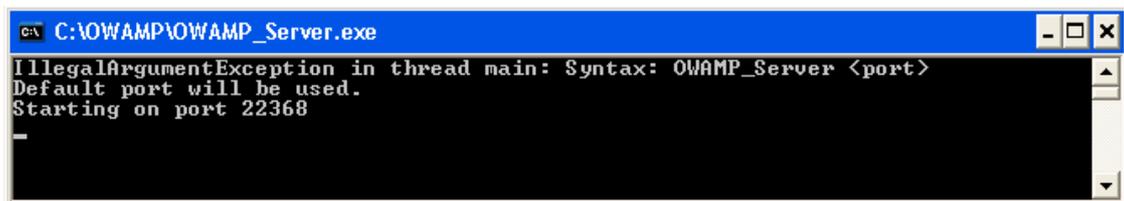
## 4 Configuring J-OWAMP modules

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. These modules don't request any other configuration besides that of the TCP port where the applications will be waiting for connection requests.

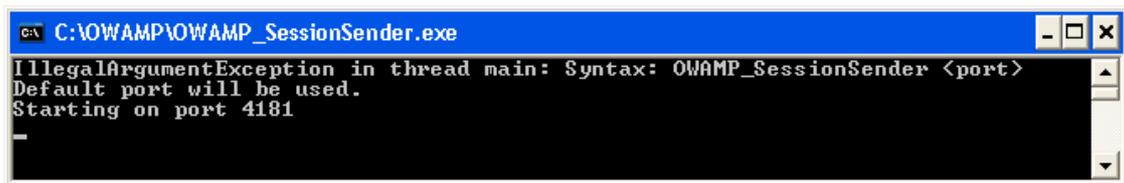
The execution of OWAMP Server presents the following window:

A screenshot of a Windows command prompt window titled "C:\OWAMP\OWAMP\_Server.exe". The text inside the window reads: "IllegalArgumentException in thread main: Syntax: OWAMP\_Server <port>  
Default port will be used.  
Starting on port 22368  
-".

```
C:\OWAMP\OWAMP_Server.exe
IllegalArgumentException in thread main: Syntax: OWAMP_Server <port>
Default port will be used.
Starting on port 22368
-
```

Figure 3: Window illustrating the execution of OWAMP\_Server.exe

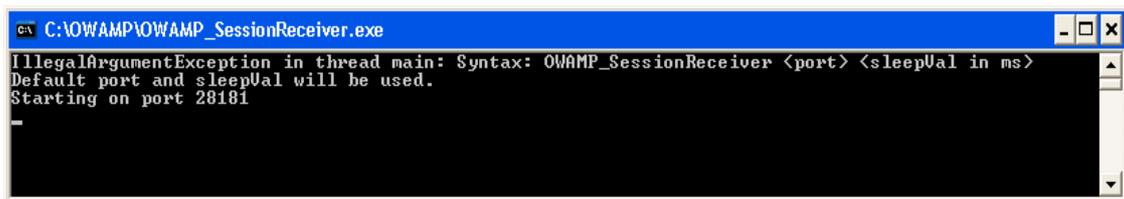
The execution of OWAMP Session-Sender presents the following window:

A screenshot of a Windows command prompt window titled "C:\OWAMP\OWAMP\_SessionSender.exe". The text inside the window reads: "IllegalArgumentException in thread main: Syntax: OWAMP\_SessionSender <port>  
Default port will be used.  
Starting on port 4181  
-".

```
C:\OWAMP\OWAMP_SessionSender.exe
IllegalArgumentException in thread main: Syntax: OWAMP_SessionSender <port>
Default port will be used.
Starting on port 4181
-
```

Figure 4: Window illustrating the execution of OWAMP\_SessionSender.exe

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

A screenshot of a Windows command prompt window titled "C:\OWAMP\OWAMP\_SessionReceiver.exe". The text inside the window reads: "IllegalArgumentException in thread main: Syntax: OWAMP\_SessionReceiver <port> <sleepVal in ms>  
Default port and sleepVal will be used.  
Starting on port 28181  
-".

```
C:\OWAMP\OWAMP_SessionReceiver.exe
IllegalArgumentException in thread main: Syntax: OWAMP_SessionReceiver <port> <sleepVal in ms>
Default port and sleepVal will be used.
Starting on port 28181
-
```

Figure 5: Window illustrating the execution of OWAMP\_SessionReceiver.exe

## 4.2 OWAMP Control-Client

The execution of this module presents the following window:



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 pressing the ENTER key. Then, the Server's TCP port number should be introduced (it corresponds to the port where the Server is waiting for connection requests). **Figure 7** illustrates a possible example:

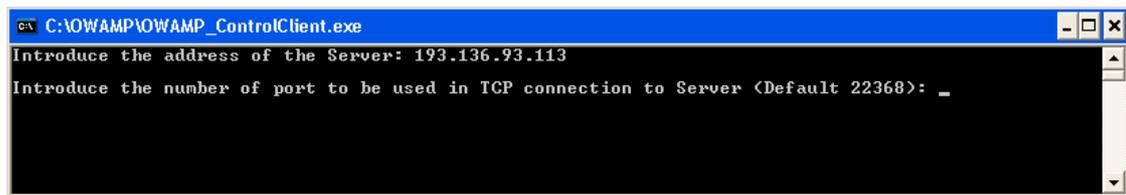


Figure 7: Window illustrating the execution of OWAMP\_ControlClient.exe

The port number considered by default (pressing the ENTER key without introducing any value) is 22368. Then, the following Menu is presented:

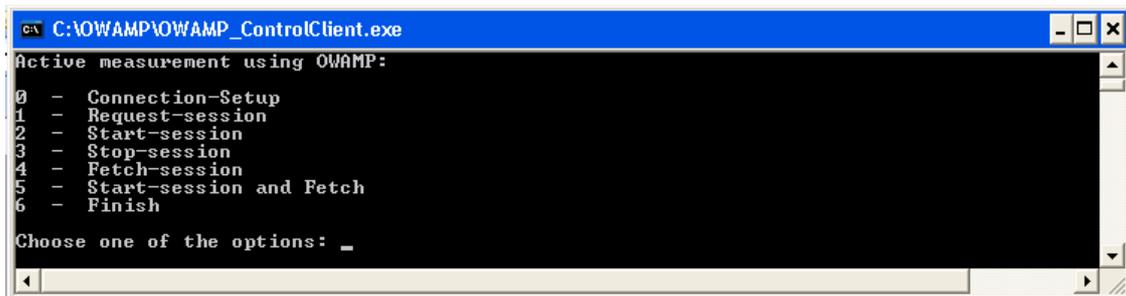
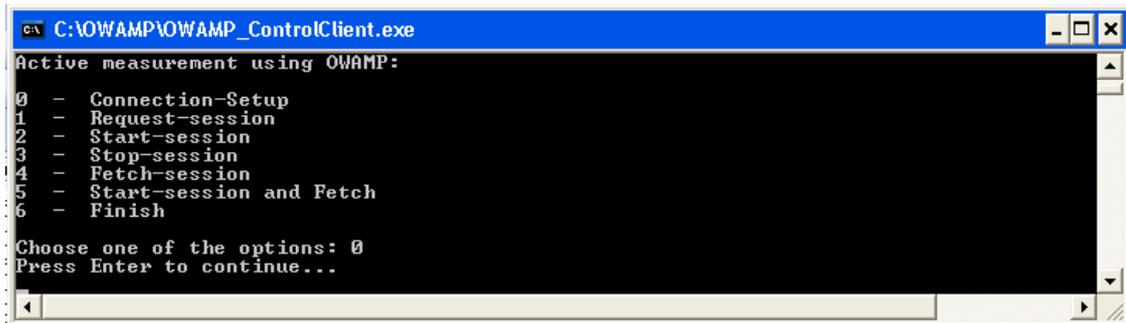


Figure 8: Main Menu

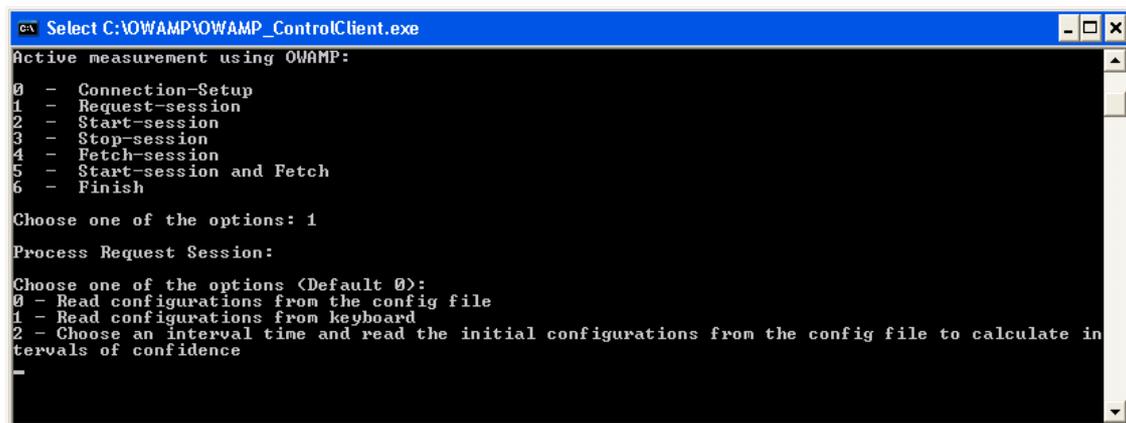
Before requesting any test session, it is necessary to establish a connection with OWAMP Server, even if the user has already requested previous tests sessions. To request this connection, the first option to be selected is option '0'. Pressing the ENTER key will validate the selected option and continue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



```
C:\OWAMP\OWAMP_ControlClient.exe
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

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



```
Select C:\OWAMP\OWAMP_ControlClient.exe
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 an interval time and read the initial configurations from the config file to calculate intervals of confidence
```

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;
- **Option 1** – to read the test configurations from the keyboard;
- **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. 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:

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

**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 timeout of test packet to be used (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 be used 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 be used 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 be used, 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:

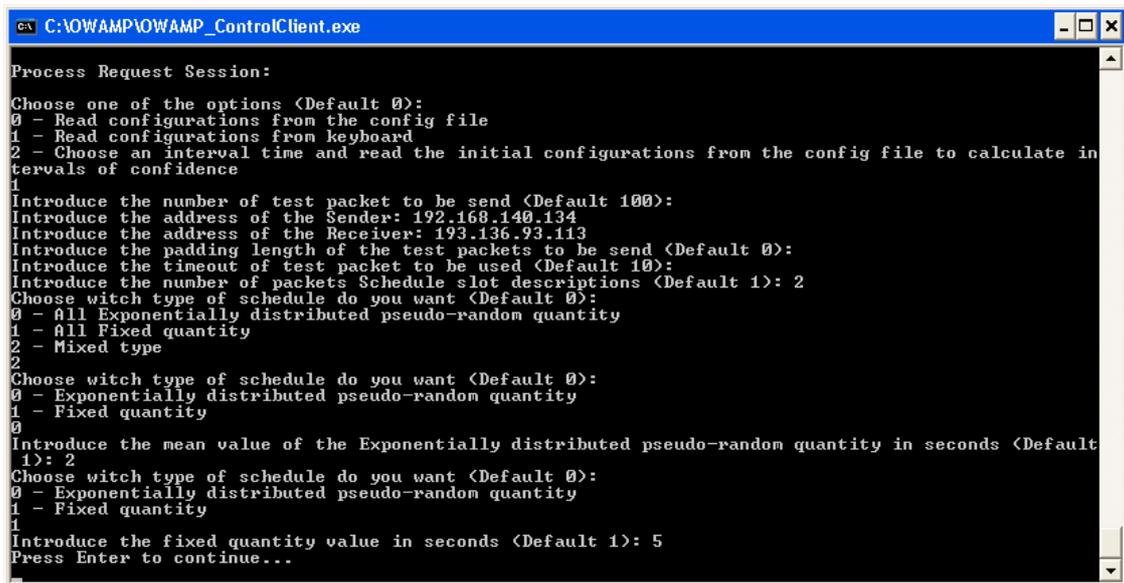


Figure 11: Choosing the type of Schedule

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):

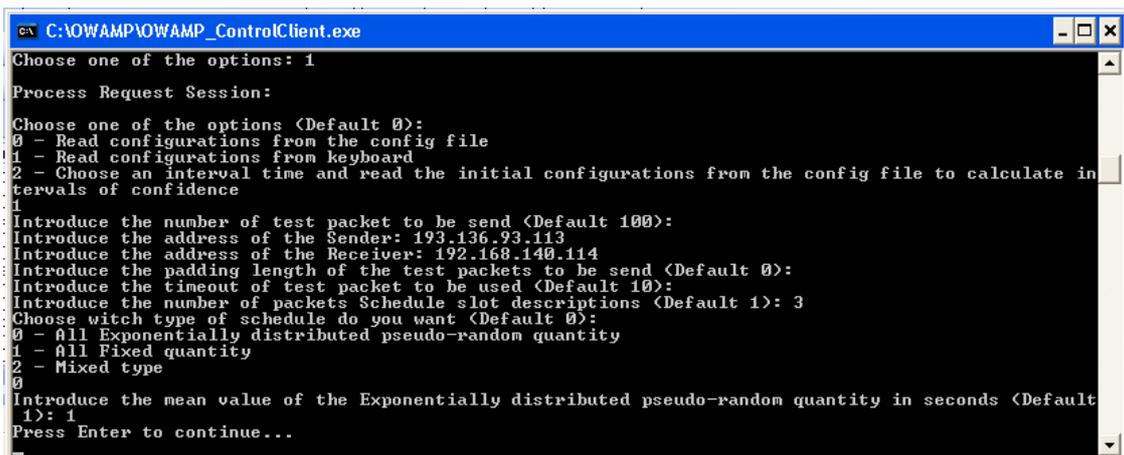


Figure 12: Introduction of the mean time value between the sending of packets (case of exponentially distributed intervals)

```

C:\VOWAMP\OWAMP_ControlClient.exe
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 an interval time and read the initial configurations from the config file to calculate in
tervals of confidence
1
Introduce the number of test packet to be send (Default 100):
Introduce the address of the Sender: 193.136.93.113
Introduce the address of the Receiver: 192.168.140.114
Introduce the padding length of the test packets to be send (Default 0):
Introduce the timeout of test packet to be used (Default 10):
Introduce the number of packets Schedule slot descriptions (Default 1): 3
Choose witch 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)

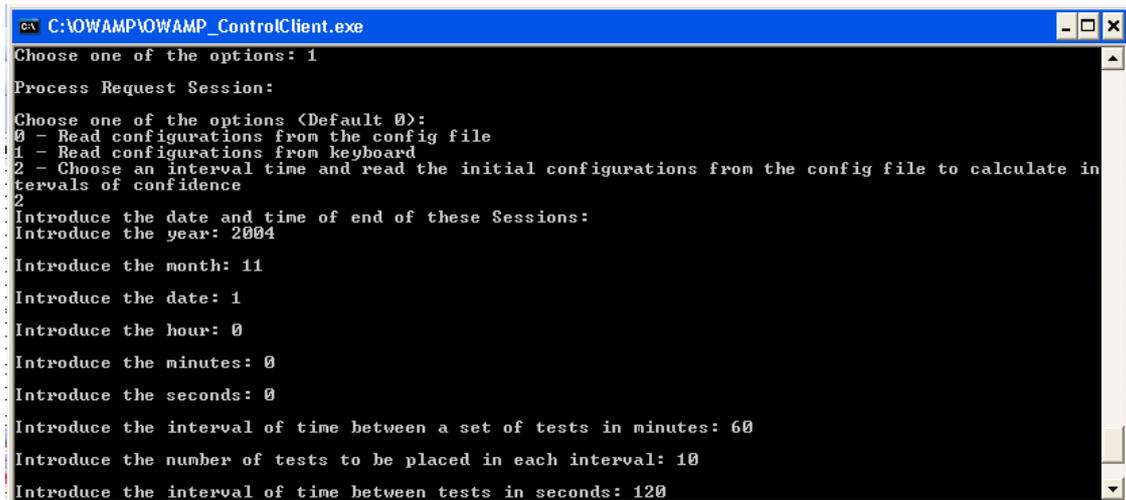
- iii) Option 2: this option is used to allow the accomplishment of tests and the calculation of the 90% confidence intervals corresponding to the following parameters: mean delay, losses and throughput. In this way, in a certain time interval the user should perform  $y$  tests for each test session 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 tests beginning time instants 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**:



```

C:\VOWAMP\OWAMP_ControlClient.exe
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 an interval time and read the initial configurations from the config file to calculate intervals of confidence
2
Introduce the date and time of end of these Sessions:
Introduce the year: 2004
Introduce the month: 11
Introduce the date: 1
Introduce the hour: 0
Introduce the minutes: 0
Introduce the seconds: 0
Introduce the interval of time between a set of tests in minutes: 60
Introduce the number of tests to be placed in each interval: 10
Introduce the interval of time 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 an interval time and read the initial configurations from the config file to calculate intervals of confidence**

2 (the user's choice)

*Introduce the dates and time of end of these Sessions:* Introduction of the date for ending the sessions

**Introduce the year: 2004** (year)

**Introduce the month: 7** (number of the month (July, in this case))

**Introduce the dates: 1** (day)

**Introduce the hour: 0** (hours)

**Introduce the minutes: 0** (minutes)

**Introduce the seconds: 0** (seconds)

**Introduce the interval of time between the set of tests in minutes: 60** (time interval between a group 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 interval of time 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.

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:

```

C:\OWAMP\OWAMP_ControlClient.exe
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

A 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 192.168.140.134 and 193.136.93.113):

```

C:\OWAMP\OWAMP_ControlClient.exe
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 witch session do you want to stop:
index  Sender Address  -->  Receiver Address
1 || 192.168.140.134    193.136.93.113
2 || 193.136.93.113    192.168.140.134
3 || - Close all sessions.
_
    
```

Figure 16: 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):

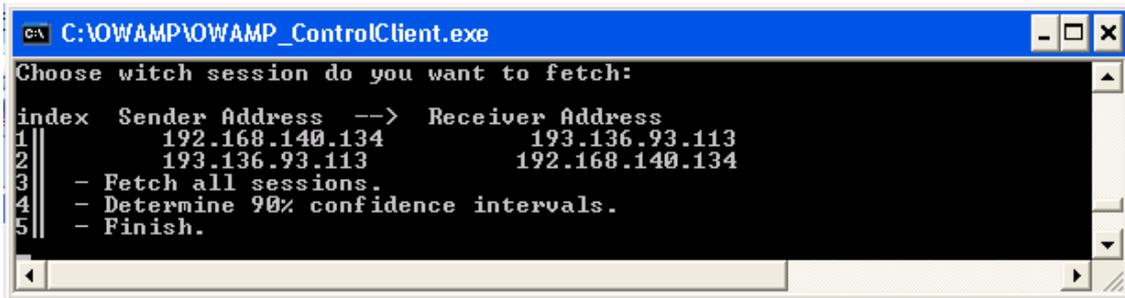


Figure 17: Menu to request the results

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 relative option 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:

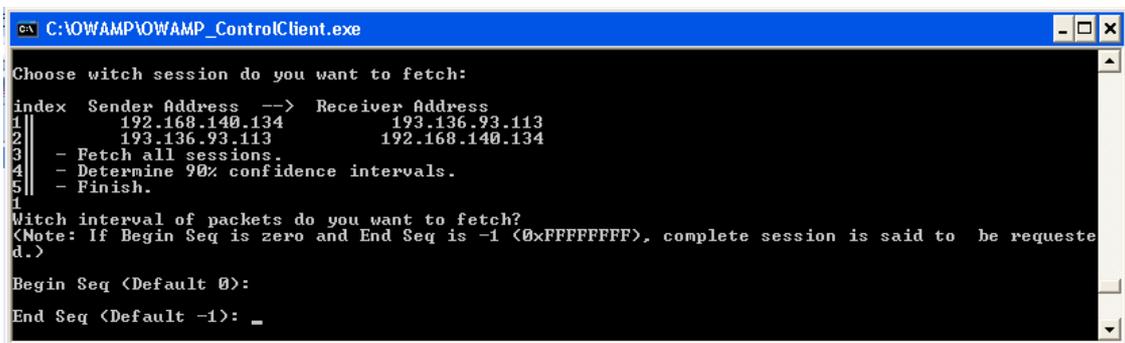


Figure 18: Menu to request the interval of results to be presented

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 19** illustrates this situation, for the example under consideration:

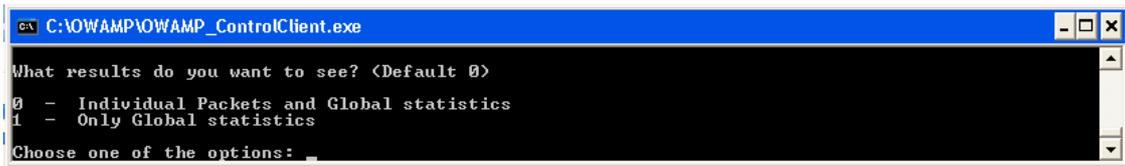


Figure 19: Menu to choice the type of results to be presented

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 results will be saved. This file will be saved, with the *.txt* extension, in the *fetchTestResults* directory, that is created in the same directory where OWAMP Control-Client is being executed.

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), also indicating a file to save the results. Then, the results of executing a session of tests should look like the ones shown in **Figure 20**:

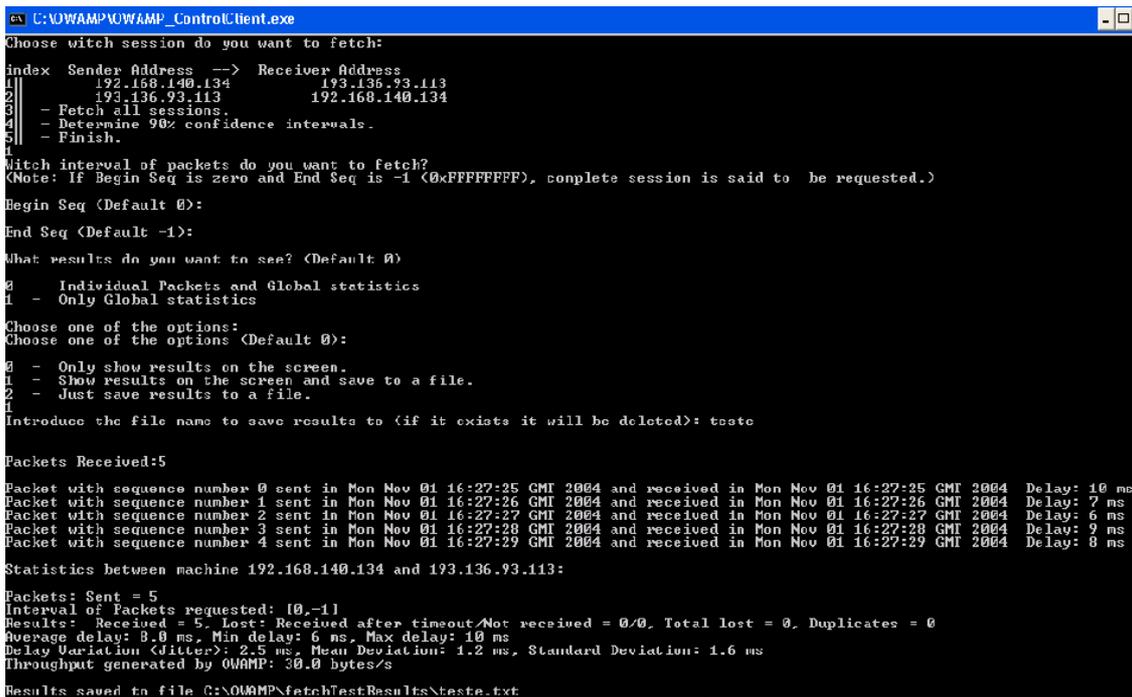


Figure 20: Results of a session of tests

If the user intends to obtain the 90% confidence intervals corresponding to the mean delay, losses 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 results will be saved. This file will be saved, with the *.txt* extension, in the directory *fetchTestResults*, created in the same directory where OWAMP Control-Client is being executed.

**Figure 21** illustrates an example, for the case of 5 test sessions between the machine with IP address 192.168.140.134 and the machine with IP address 193.136.93.113:

```

C:\VOWAMP\VOWAMP_ControlClient.exe
Choose witch session do you want to fetch:
index  Sender Address  -->  Receiver Address
1  192.168.140.134      193.136.93.113
2  192.168.140.134      193.136.93.113
3  192.168.140.134      193.136.93.113
4  192.168.140.134      193.136.93.113
5  192.168.140.134      193.136.93.113
6  - Fetch all sessions.
7  - Determine 90% confidence intervals.
8  - Finish.
?
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): teste
Statistics for test interval number 1
Packets sent      Average delay (ms)  Min delay (ms)  Max delay (ms)  loss Received after timeout/Not received  Throughput  File
1 Test 60 2.3 -1.0 53.0 0/7 22.7 Mon Nov 01 16:47:01 GMT
2 Test 60 14.5 -14.0 106.0 0/6 23.1 Mon Nov 01 16:49:01 GMT
3 Test 60 8.9 2.0 77.0 0/7 22.7 Mon Nov 01 16:51:01 GMT
4 Test 60 7.8 3.0 110.0 0/0 24.4 Mon Nov 01 16:53:01 GMT
5 Test 60 11.0 -13.0 -5.0 0/0 24.4 Mon Nov 01 16:55:01 GMT
Global Statistics
Average delay (ms)  Average loss Received after timeout/Not received  Average Throughput (bytes/s)
8.9 0.0/4.0 23.4
Interval 90% [5.617 12.195] [0.000 0.000] / [1.297 6.703] [22.793 24.076]
Results saved to file C:\VOWAMP\fetchTestResults\teste.txt
    
```

Figure 21: 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.

### 4.3 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).

The execution of the OWAMP Fetch-Client module presents the following window:

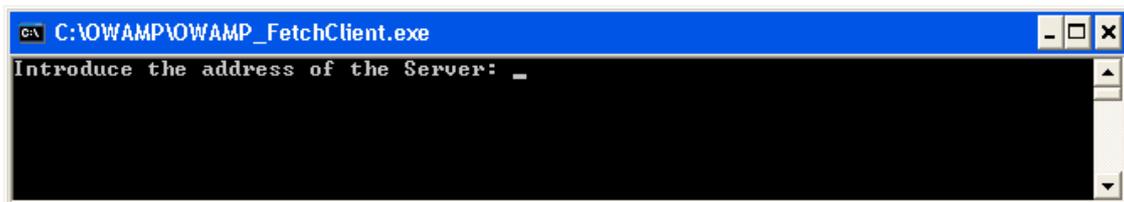


Figure 22: Window of execution of OWAMP\_FetchClient.exe.

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. Then, the number of the TCP port that is used in the communication with the Server should be introduced. **Figure 23** illustrates a possible example:

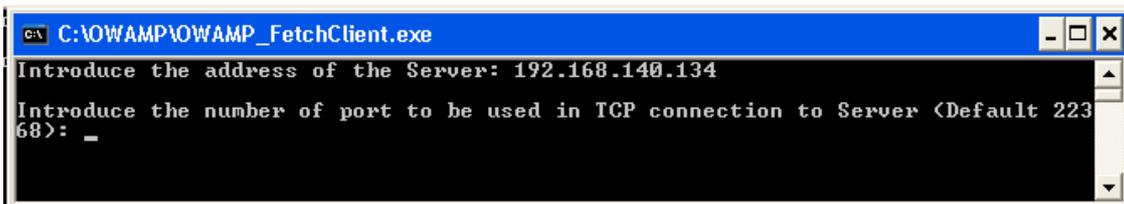


Figure 23: Window of execution of OWAMP\_FetchClient.exe

The port number considered by default is '22368'. Then, considering the same example that was used in the OWAMP Control-Client description, the following Menu is presented:

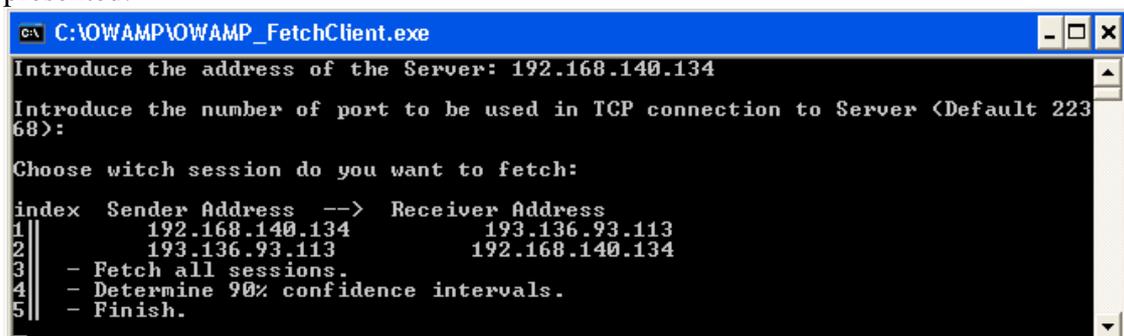


Figure 24: Menu used to request the results

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 (page 15).