Simulation Server for Project 25: Inter-RF Subsystem Interface (ISSI)

September 30, 2011
Simulation Server v1.0.0

1. Executive Overview

There are two parts to the Project 25: Inter-RF Subsystem Interface (ISSI) Network Simulator Tool:

- The simulator that models the behavior of the ISSI protocol, along with the server that allows to interact with the simulator, and
- The graphical user interface, which 1) permits the entering of the configuration information for input to the program, and 2) parses the output trace files for visualization and replay of the simulations.

The ISSI-NST server provides an interface between the ISSI model and the graphical user interface. This program receives user commands through the network, and runs the required simulations using the ISSI model. Similarly, the results of these simulations are sent to the user's graphical user interface when requested.

This document covers only the server program. This server is written in Java[1,2] and interacts with the network simulator NS-2 (for which the ISSI model has been developed.) to deploy the simulation models input by the user, run the simulations, collect the output, and send it back to the user upon request.

Section 2 contains a quick start to deploy and use the server. Section 3 provides a detailed description of the installation process. Section 4 describes the operation and configuration of the server. Section 5 describes the security of the communications and the authentication process.

As additional supporting documentation, Annex A describes the protocol used to communicate between the clients' user interfaces and the server, and Annex B shows how to install the server in Microsoft Windows[3] systems.

[1] Java is a registered trademark of Sun Microsystems, Inc. or its subsidiaries in the United States and other countries

[2] Disclaimer: Any mention of commercial products is for information only; it does not imply recommendation or endorsement by the NIST.

[3] Microsoft Windows is a registered trademark of Microsoft Corporation.

2. Quick Start

Install the server package by running the command

```
sh ISSI-NST_Server_setup_linux.bin
```

in the directory where the installation file has been saved.

Follow the installation script steps to provide an initial configuration. If the installation script finds a missing dependency, fix the problem and re-run the installation command.

Once the installation is successfully completed, start the server by running in the installation directory the following command:

```
java –jar NS2-Viz2_Server.jar
```

This will launch the server and log all the error events in the `log` directory. If we want to log more events, we can do so by adding the `log-level <new_log_level>` parameter to the previous command:

```
java –jar NS2-Viz2_Server.jar log-level <new_log_level>
```

where `<new_log_level>` is one of `0`, `1` or `2`:

`0`: (Debug log) This will log all the events in the server.

`1`: (Information log) This will log all the errors and significant events.

`2`: (Error log) This will only log the errors.

If no parameter is specified, the system will use a log level 2 by default (Error log)

In order to reconfigure the server (including adding or removing user accounts), start the server administration GUI by clicking the *Server Configuration* icon (if using Windows) or running

```
java –cp NS2-Viz2_Server.jar gov.nist.antd.hsntg.server.admin.AdminGui
```

if using linux.

2.1. Default Parameters

Users and passwords:

- **Simulation user:**
  username `user01` ; password `user01`
- **Administration user:**
  username `admin` ; password `admin`

Network addresses and ports:

- **Simulation:**
  Listening address `any`; port `8000`
- **Administration:**
  Listening address `any`; port `8080`

3. Operation

3.1. Starting the Server

In order to start the server we will run the following command from the installation directory:

```
java –jar NS2-Viz2_Server.jar
```

Note: It is important to make sure that the `java` command launches a compatible Java Virtual Machine (see 3.1). Otherwise, the server may behave unexpectedly.

This will launch the server and log all the error events in the `log` directory. If we want to log more events, we can do so by adding the `log-level <new_log_level>` parameter to the previous command:

```
java –jar NS2-Viz2_Server.jar log-level <new_log_level>
```

where `<new_log_level>` is one of `0`, `1` or `2`:

`0`: (Debug log) This will log all the events in the server.

`1`: (Information log) This will log all the errors and significant events.

`2`: (Error log) This will only log the errors.

If no parameter is specified, the system will use a log level 2 by default (Error log)

After the server starts, a file named `viznet_server_release.txt` is created in the log directory with the version of the Java server and the simulation model.

3.2. Configuring the Server

The tool to configure the server is currently included in the ISSI-NST installers (both the client and the server). This tool allows for the configuration of the ns installation directory, the network address, port and backlog for the simulation and configuration services, the log filename and the files with the list of users and passwords. It also allows for the creation, deletion and modification of users (users allowed to run simulations) and administrators (users that can change the configuration).

To launch the server configuration tool in a Windows system, use the `ISSI NST Server Configuration` icon (Figure 9):

Figure 1 ISSI NST Server Configuration Icon

In a Linux or OS X system, locate the server installation directory, and launch the configuration utility with the following command:

```
java -cp <server_installation_directory>/NS2-Viz2_Server.jar
gov.nist.antd.hsntg.server.admin.AdminGui
```

When the configuration tool is launched, it will request the server address and port to connect to, and the user and password to use to connect. At this point we need to provide the tool with the network address and port we chose during the installation to be used for the server **administration** (not the simulation ones). By default, this will be any network address that can be used to contact the server, and port 8080. The username and password to supply will be those of a valid administrator account (by default, username: `admin`; password: `admin`).

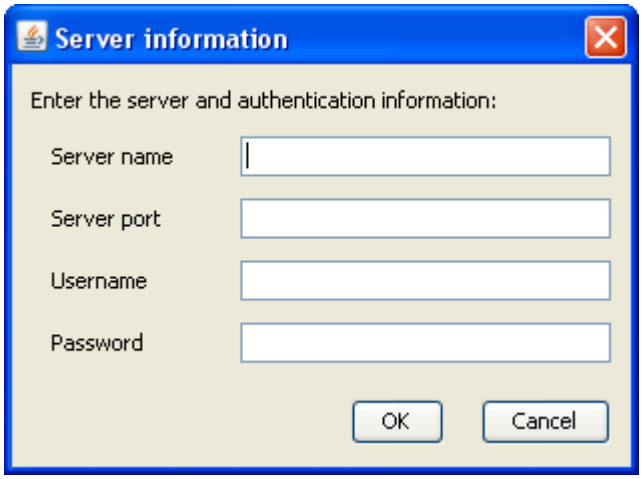
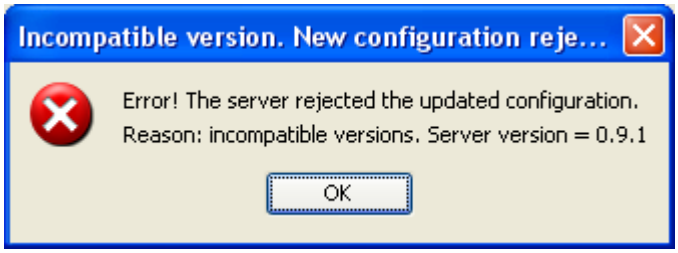


Figure 2 Configuration Tool Initial Window

Upon a successful authentication, the main window is loaded. This window contains three tabs that are used to configure the server parameters, the simulation user accounts and the administrator accounts. At the bottom of the window there are two buttons to save the updated configuration in the server (`Save in server`) or cancelling the configuration process (`Exit`).

Note: Clicking the '`Save in server`' button updates the configuration in the server with the changes made in all the tabs, not just the current active one.

In order to be able to save the configuration in the server, the configuration tool must have a version number compatible with the server. Otherwise, the server will reject the configuration update:



Upon loading, the tool will show the current configuration parameters and accounts used by the server (Figure 11). The server parameters available to configure are:

- **NS-2 base directory:** The full path to the directory in which the simulation model is installed in the server.
- **Simulation service address:** The network address used to listen to incoming simulation requests from clients.
- **Simulation service port:** The TCP port used to listen to incoming simulation requests from clients.
- **Simulation service concurrent connections:** The amount of users that can concurrently use the simulation service. If more users than this value try to use the service, they will be kept on a first come, first served queue until other users disconnect or their TCP connections time out.
- **Administration service address:** The network address used to listen to incoming configuration requests from configuration tools (like the one that is being run).
- **Administration service port:** The TCP port used to listen to incoming configuration requests from configuration tools (like the one that is being run).
- **Administration service concurrent connections:** The amount of users that can concurrently use the configuration service. If more users than this value try to use the service, they will be kept on a first come, first served queue until other users disconnect or their TCP connections time out.
- **Simulation log filename:** The name of the file in the `log` subdirectory in the server that stores the logged events.
- **Filename with the simulation users' accounts:** The name of the file in the `log` subdirectory in the server that stores the user accounts used to run simulations.
- **Filename with the administration users' accounts:** The name of the file in the `log` subdirectory in the server that stores the accounts used to configure the server.

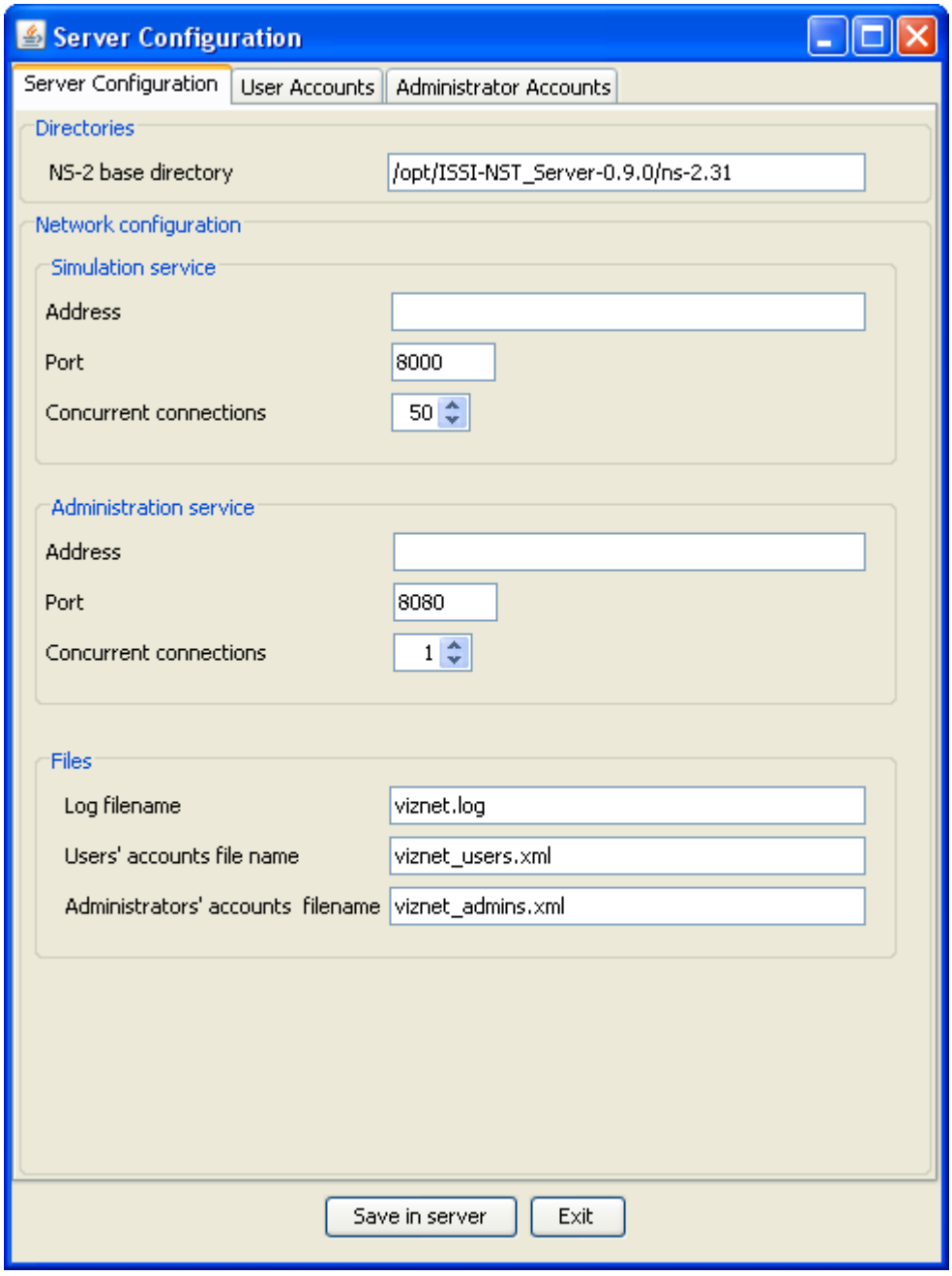


Figure 3 Server Parameters Tab

The tab for configuring the simulation user accounts (labeled '`User accounts`', Figure 12) presents the list of available accounts to be used when launching simulations from the client GUI. For each account, the following information is shown:

- **Username:** The name of the user.
- **Iterations:** The amount of password iterations available for the user (see Section 5 for details of the password management).
- **Change password button:** Button that allows changing the password for the user.
- **Delete account button:** Button that allows deleting the user account.

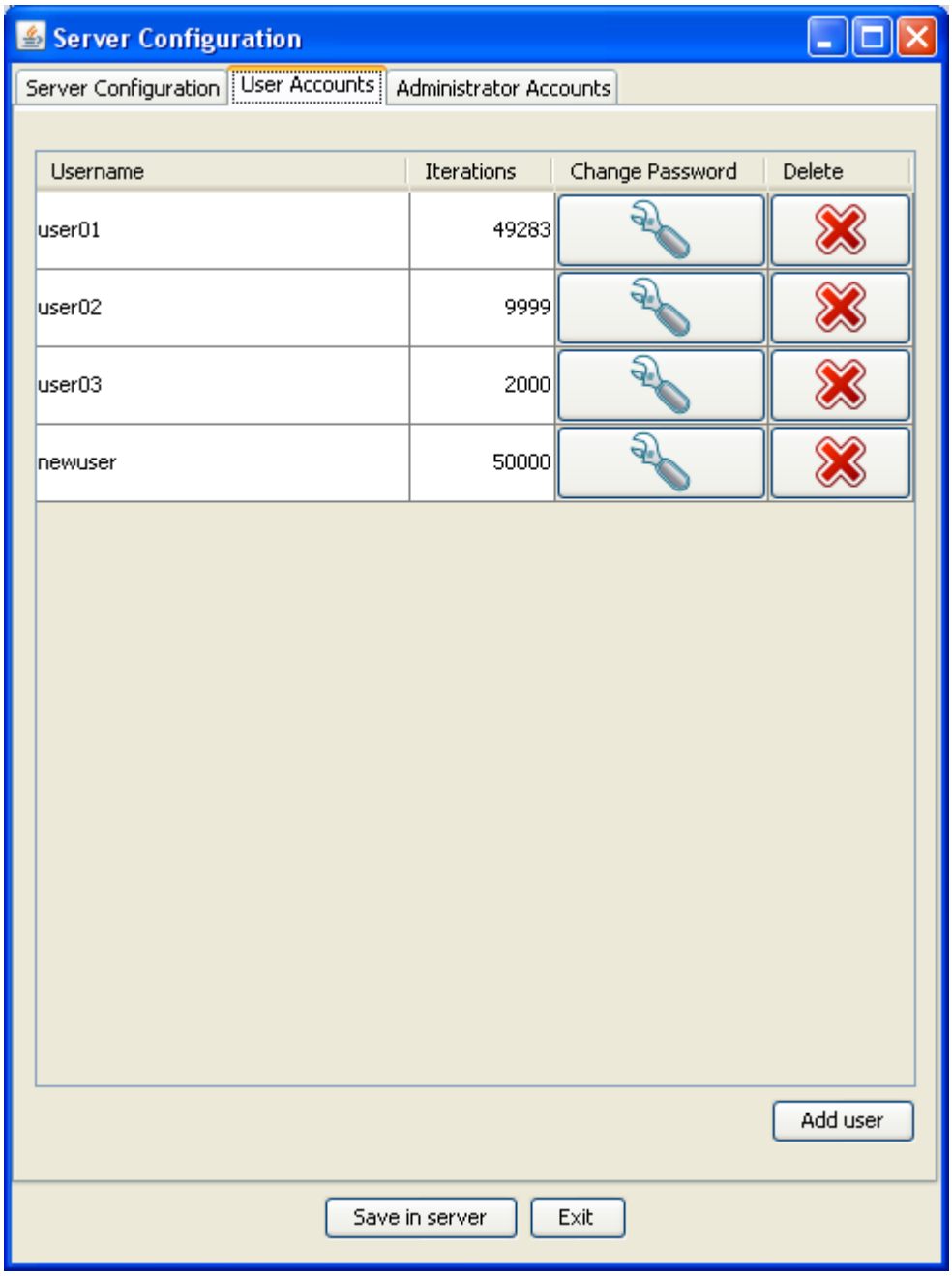


Figure 4 Simulation Users Accounts Management Tab

The ‘Add user’ button allows to create a new user account by showing the ‘new user account’ window (Figure 13):

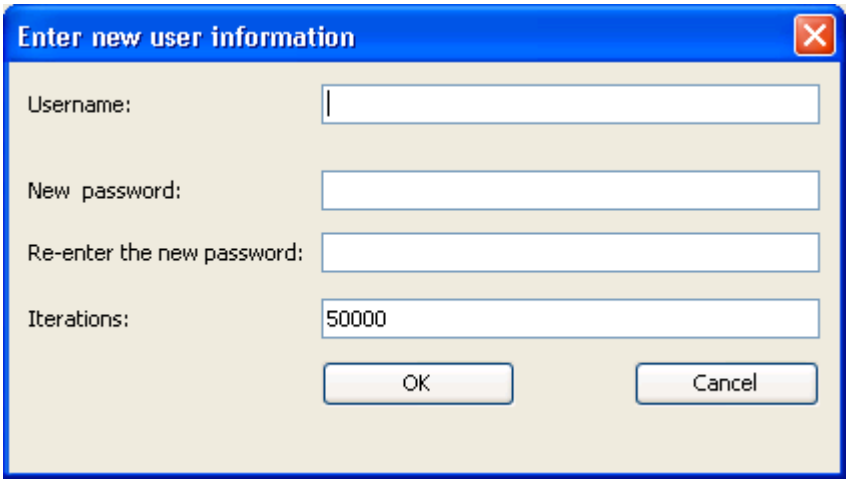


Figure 5 New User Account Window

In this window we are prompted for the account's username, and password (twice, to prevent typing errors). Also, we can modify the default starting number of password iterations.

The tab for configuring the administrator accounts (those to be used to configure the server, Figure 14) behaves in the same way as the tab for configuring the simulation accounts:

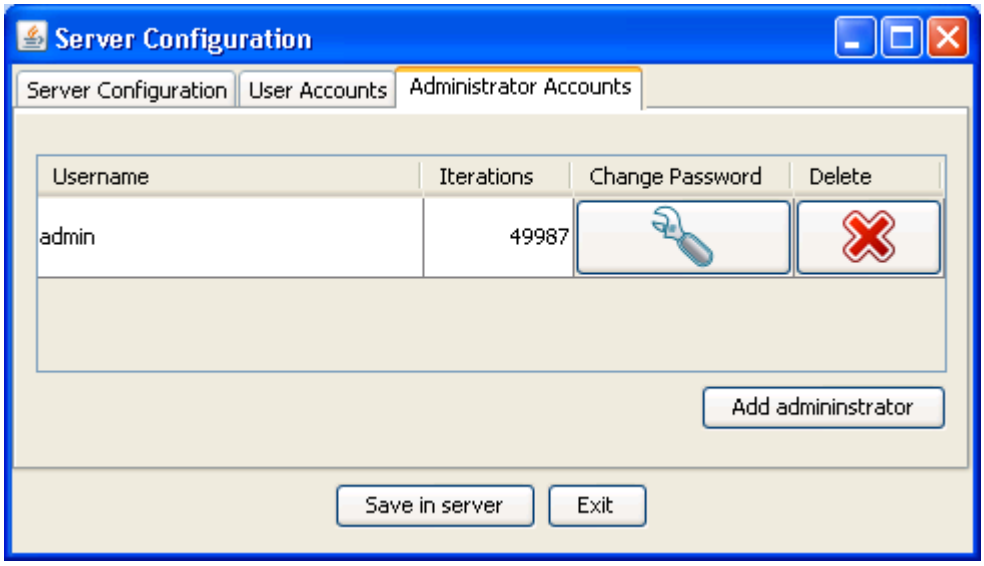


Figure 6 Administration Accounts Configuration Tab

3.3. Running Simulations

Simulations are run from the user GUI, which configures the topology, applications and duration of the simulation. When the client wants to run a simulation, it sends the topology (.TCL) and project configuration file (.XML) to the server, which assigns the simulation an identifier (Universally Unique Identifier, UUID). The server creates a separate directory for this simulation in the `simulations` directory. This separate directory's name is the UUID assigned to the simulation, and the files received from the client are copied in it. Next, the server launches an ns-2 process in a separate thread, and stores the object with the system process.

Periodically, the client will request from the server information about the progress of the simulation. The server will select the proper system process using the UUID, verify if the process is already done, and if not, locate the trace file and compute the progress so far. When the simulation is finished, the server sends the output files from the simulation (log, simulation trace and output audio traces) to the client. Once the transfer is over, and if the user selected so, the client GUI will instruct the server to delete the simulation directory with all the files for the project.

4. Security

The security of the communications is achieved using a two-fold approach:

- On the one hand, the communication itself is protected using TLS 1.0. The simulation service requires server authentication, while the configuration service requires server and client authentication.
- On the other hand, at the application level the users have to authenticate themselves against the server, using an S/KEY mechanism.

Currently some security options are hardcoded or fixed to reduce the user inconvenience, as follows:

- TLS certificates are currently shipped with the installer.
- The keys to the certificate stores are hardcoded in the code.
- When a password expires, it is automatically renewed using the old password.

4.1. Authentication Protocol

The authentication protocol is a challenge – response protocol based on S/KEY (RFC 1760 and RFC 2289), a mechanism to implement One Time Passwords (OTPs). S/KEY is based on executing a cryptographic hash function several times over the password, and replaying the intermediate hashes in reverse order to prevent message replays and password interception.

4.1.1. S/KEY

The authentication using S/KEY works as follows: The authentication server stores the username (*U*), the user password hashed *k* times (P_k), the value $k - 1$ and, optionally, the hashing algorithm (*A*). When it receives an authentication request for a given username (*U)*, the server returns the stored value $k - 1$ and optionally, the hashing algorithm (*A)*. When the user receives this message, he hashes his password $k - 1$ times, and sends the username (U) and the hashed password (P_{k-1}) to the server. Upon reception of this message, the server hashes the password once more and compares the result P_k' with the stored value P_k. If both passwords match, the authentication is successful, and the server replaces P_k with P_{k-1} and $k - 1$ with $k-2$ in the entry for user *U*,

To prevent the iteration counter *k* from becoming 0, a minimum threshold (min_T) is defined. When *k* becomes less than min_T, the server requests the user to reset his password, which may be done online or offline.

In this implementation, the minimum threshold is defined as 2000 iterations, and passwords are reset to a maximum of 50000 iterations.

5. References

[1] RFC 1760 “The S/KEY One-Time Password System”; Informational RFC; N. Haller; February 1995.
[2] RFC 2289 “A One-Time Password System”; Standard RFC; N. Haller, C. Metz, P. Nesser and M. Straw; February 1998.
[3] RFC 2246 “The TLS protocol. Version 1.0”; Proposed Standard; T. Dierks and C. Allen; January 1999.

A. Communications Protocol

The communication protocol used in VizNet is a request – response protocol that runs over Transport Layer Security (TLS) without authentication. Once the connection is established, the server authenticates the client with a protocol based on S/KEY. Upon successful authentication, the application message exchange between the client and server may take place.

All the available operations for the client are linked to a given unique simulation identification (SID). A client may request a new SID from the server, or use an existing one (e.g., when connecting at a later time to retrieve the output files from a simulation).

A.1. Protocol Messages

All the protocol messages are sent in plaintext. Message codes smaller than 099 are reserved for the authentication protocol, and they are processed by the authentication agent. For the server to process any message from the client that is not the `Version Ping`, the server must have previously authenticated the client. The protocol messages begin with a three digit code and are followed by either the client's version, or the server and simulation model version. After those fields, the message may contain more information depending on the type of message:

099 – Version Ping

Format: `100 <version> [<version>]\n`

Sent by the client to announce its version to the server. The server replies with a message with the same code, and its version and the model version as parameters.

100 – Request New Simulation ID

Format: `100 <client_version>\n`

Sent by the client to request a new unique simulation ID.

110 – Request Send Scenario

Format: `110 <client_version> <SID>\n`

Sent by the client to indicate the transfer of the simulation files to the server.

111 – Request Scenario File Transfer

Format: `111 <client_version> <SID> <data_size> \n<files>\n`

Sent by the client to transfer the scenario file to the server. The files are sent in a compressed stream.

120 – Request Simulation Start

Format: `120 <client_version> <SID>\n`

Sent by the client to request the initiation of the simulation in the server.

130 – Request Simulation Status

Format: `130 <client_version> <SID>\n`

Sent by the client to inquire about the status of the simulation.

140 – Request Simulation End

Format: `140 <client_version> <SID>\n`

Sent by the client to request the termination of the simulation process in the server.

150 – Request Get Simulation Output

Format: `150 <client_version> <SID>\n`

Sent by the client to request the output files of the simulation.

160 – Request Terminate SID

Format: `160 <client_version> <SID>\n`

Sent by the client to delete all the files related to the provided Simulation ID (SID) from the server.

200 – Response SID

Format: `200 <server_version> <model_version> <SID>\n`

Sent by the server to provide the client with a new SID.

210 – Response Proceed Send Scenario

Format: `210 <server_version> <model_version>\n`

Sent by the server to acknowledge the scenario transfer requested by the client. This message is sent after the server has created the required directories and files with no error.

211 – Response Scenario Transfer OK

Format: `211 <server_version> <model_version>\n`

Sent by the server to indicate to the client that the scenario file was successfully received and validated.

220 – Response Simulation Started

Format: `220 <server_version> <model_version>\n`

Sent by the server to indicate that the simulation has been launched. Note that the successful start of the simulation does not imply absence of errors in the simulation or the configuration of the scenario (apart from those detected with the basic validation performed upon reception of the file).

230 – Response Simulation Status

Format: `230 <server_version> <model_version> <latest_timestamp> [<finished> <cancelled>]\n`

Sent by the server to inform the client about the progress of a simulation. This message implies that the SID provided exists in the server and the TCL hash received matches that of the scenario stored in the server. If the simulation is no longer running, two additional fields are added to the message:

- finished: an 8-bit integer with the process exit code. An exit code of 0 indicates success, while any other value indicates an abnormal termination of the simulation.
- cancelled: value of ‘true’ if the simulation was cancelled by the user; ‘false’ otherwise.

240 – Response Simulation Ended

Format: `240 <server_version> <model_version>\n`

Sent by the server to indicate the successful termination of the simulation process in the server.

250 – Response Simulation Output

Format: `250 <SID> <server_version> <model_version> <trace_hash> <trace_hash_alg> <total_size> <trace_name> <trace_size> <log_name> <log size> \n<output_files> \n`

Sent by the server to transfer the output files from the simulation to the client. The output files are sent in a compressed stream.

260 – Response SID Terminated

Format: `260 <server_version> <model_version>\n`

Sent by the server to indicate the successful deletion of the files for the requested SID.

300 –Invalid SID

Format: `300 <server_version> <model_version>\n`

Sent by the server to indicate that the SID provided by the user in the request is invalid.

301 – Invalid Operation: Not Authenticated

Format: `301\n`

Sent by the server to indicate that the client has not performed a valid authentication prior to sending command messages.

302 – Invalid Scenario File

Format: `302 <server_version> <model_version>[\n<TCL error_message> \n]`

Sent by the server to indicate that the scenario file received was invalid. It may include an error message to help the client correct the problem.

304 – Unable to Terminate SID

Format: `304 <server_version> <model_version>\n`

Sent by the server to indicate that the requested SID could not be terminated.

305 – Protocol Error

Format: `305 [<server_version> <model_version>]\n`

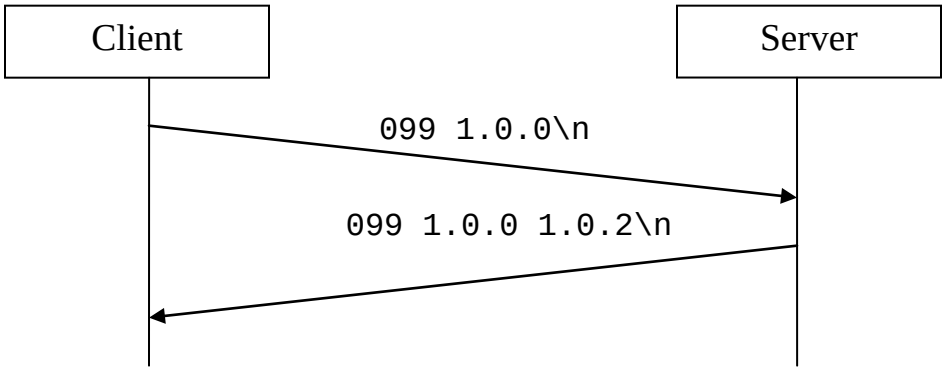
Sent by either peer to indicate the reception of an unexpected message (i.e., the current state in the state machine does not provide indications to handle the received message).

306 – Error: Incompatible version

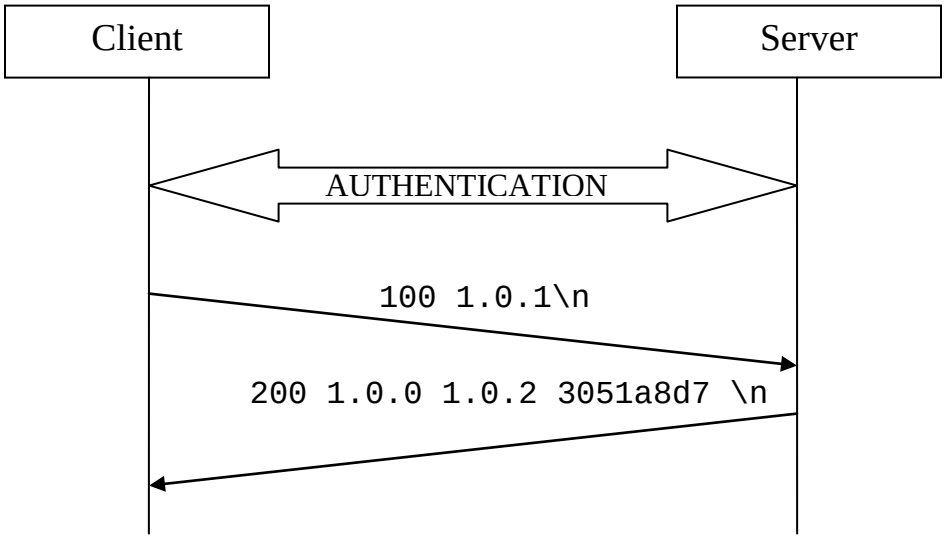
Format: `306 <server_version> <model_version>\n`

Sent by the server to indicate that the client is incompatible, and the last request will be ignored.

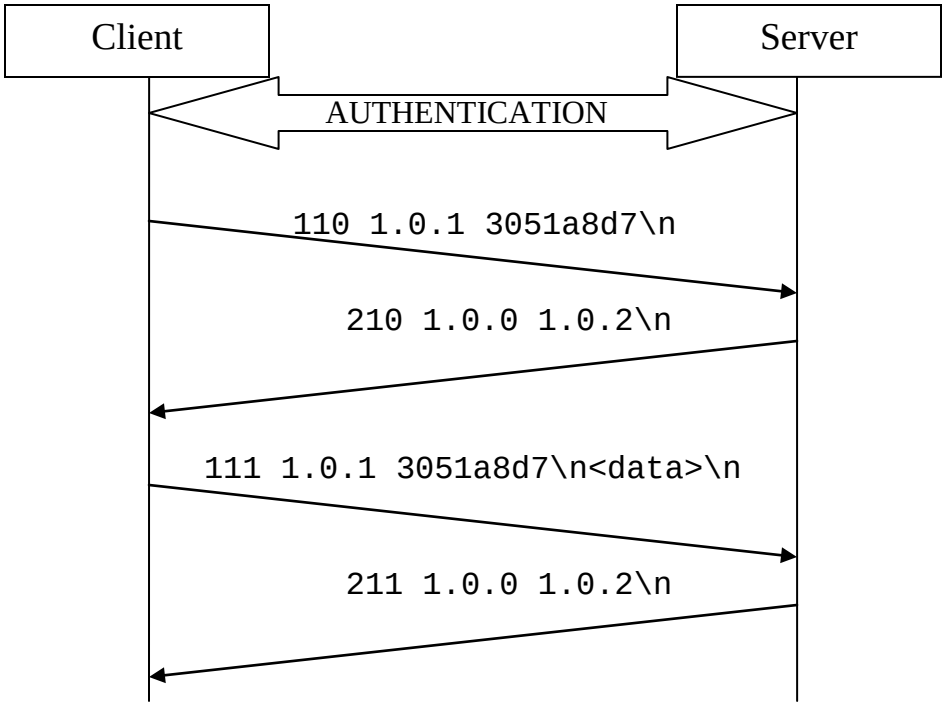
A.2. Message Flows



Version Ping

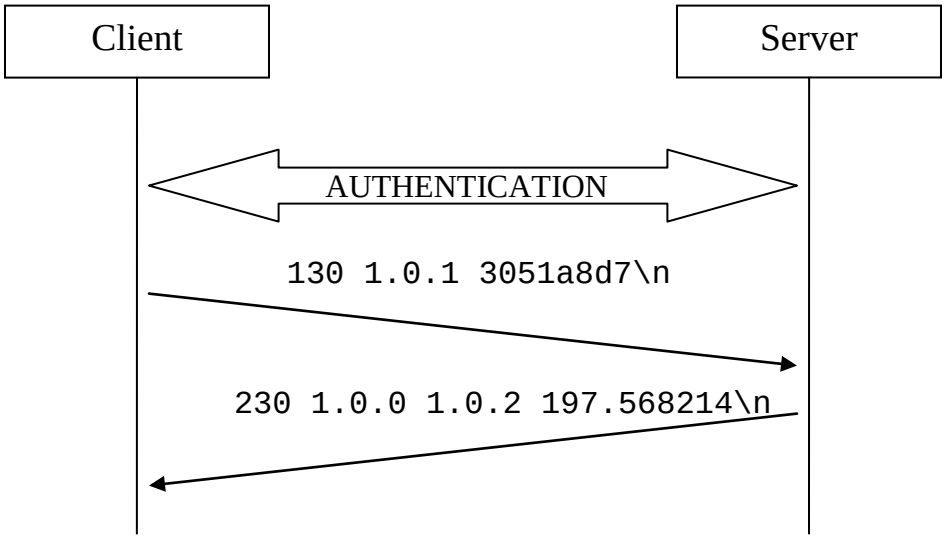


Successful SID request

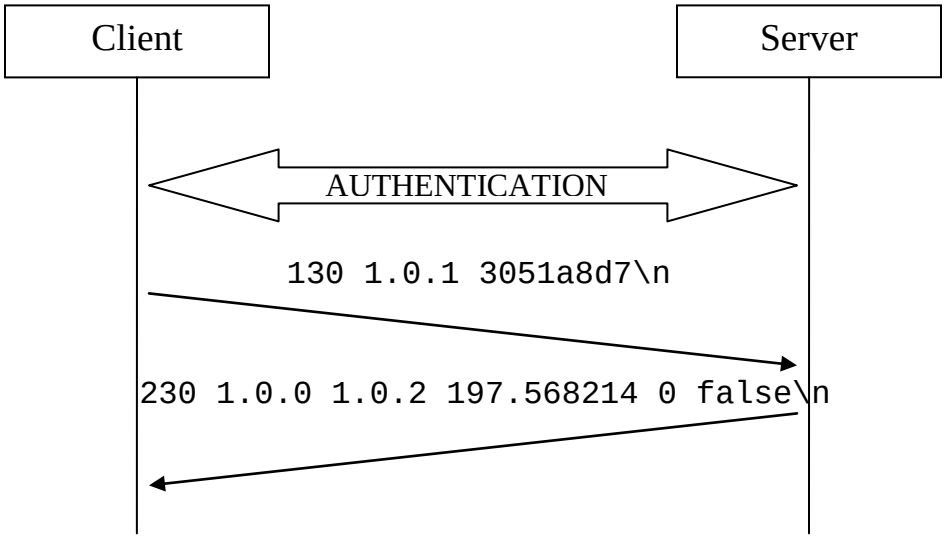


Successful scenario transfer

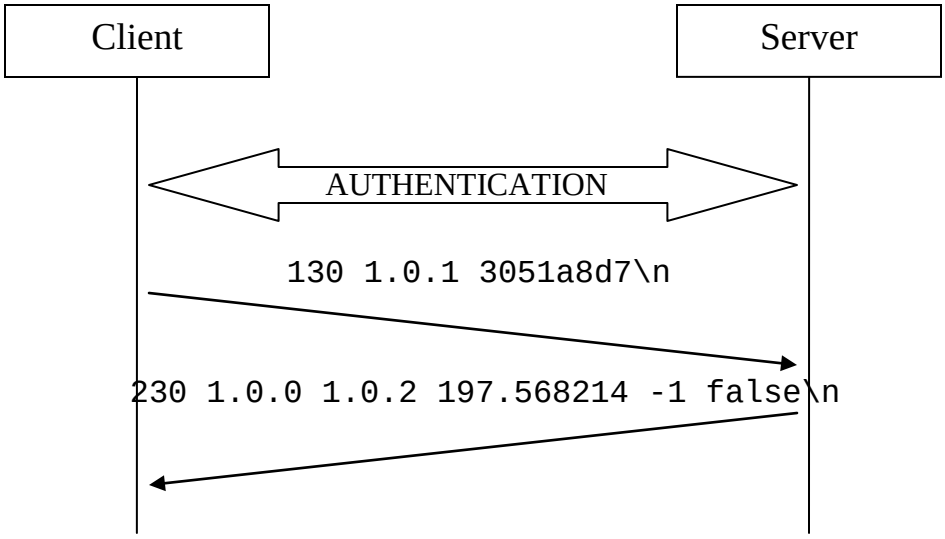
Successful simulation launch



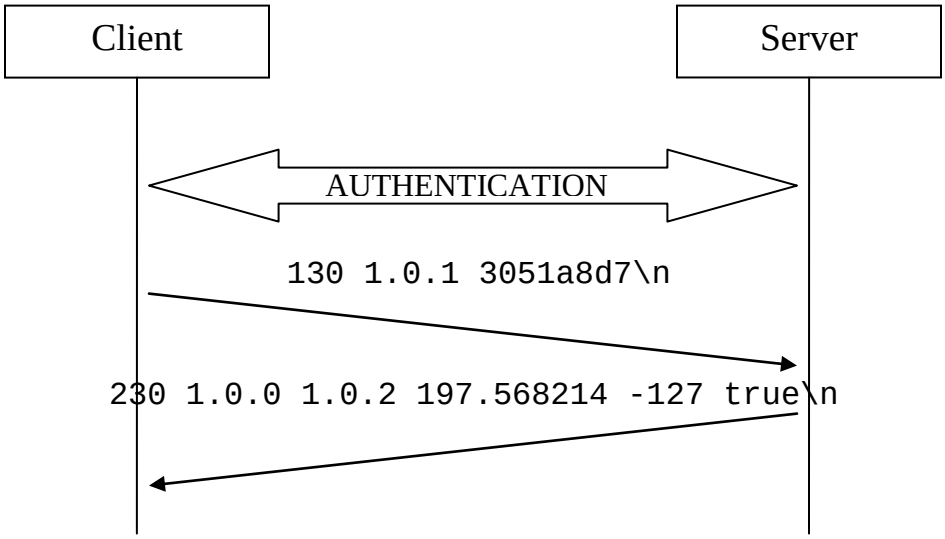
Successful simulation status check (simulation not finished)



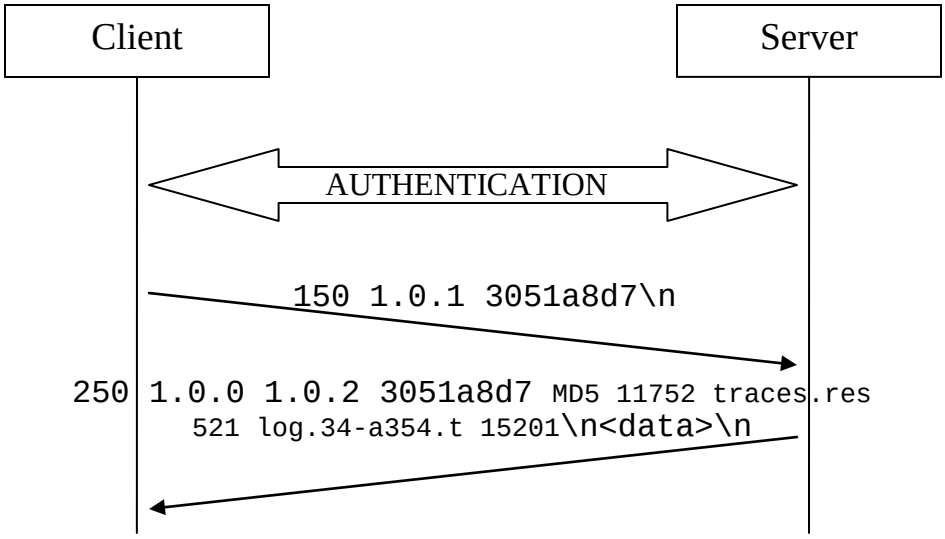
Successful simulation status check (simulation finished successfully, not cancelled)



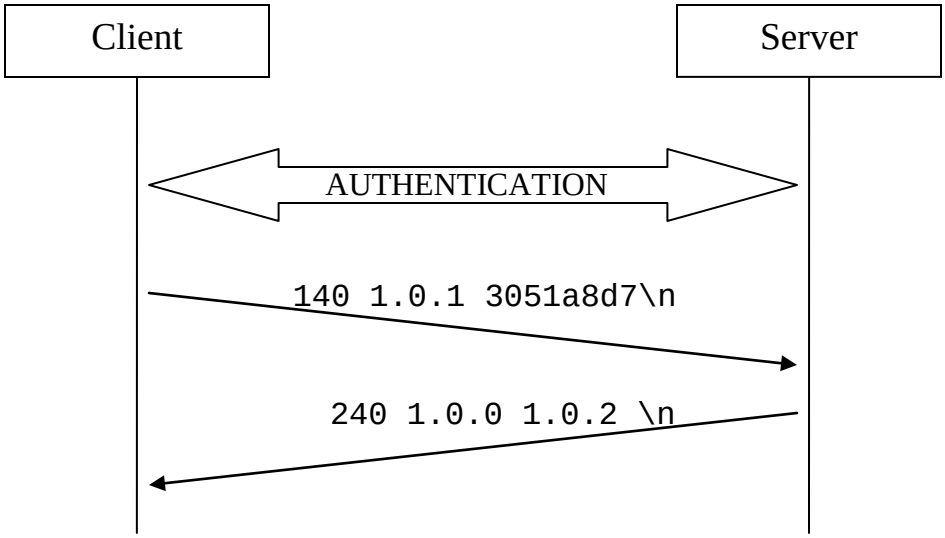
Successful simulation status check (simulation finished with an error, not cancelled)



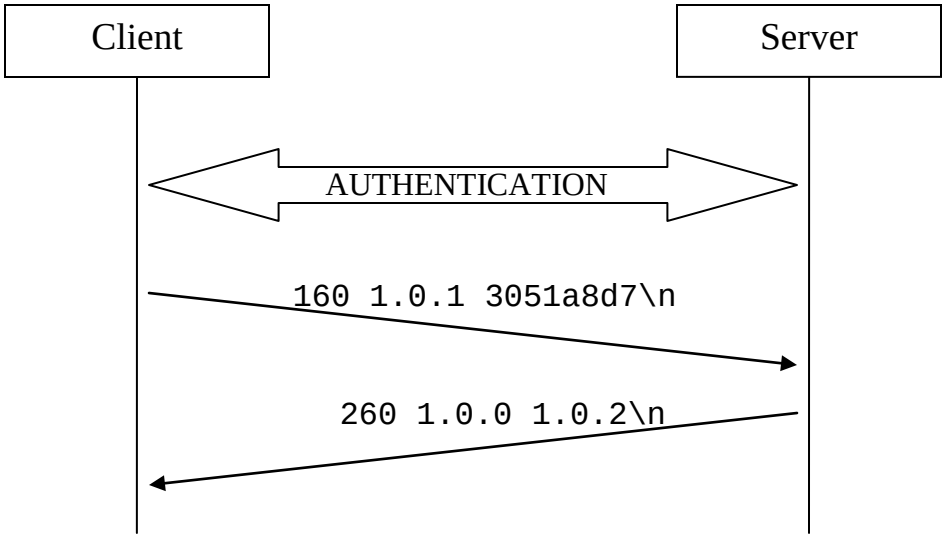
Successful simulation status check (simulation cancelled)



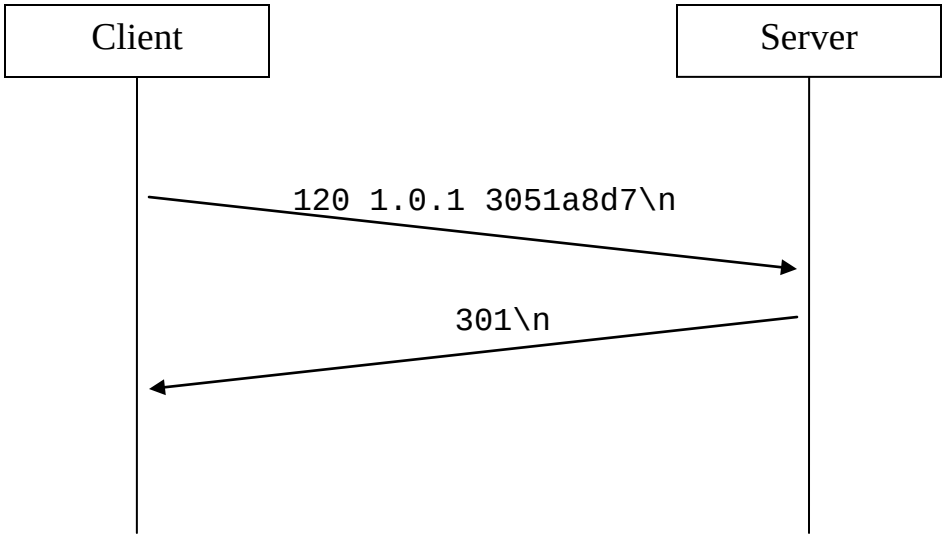
Successful output retrieval



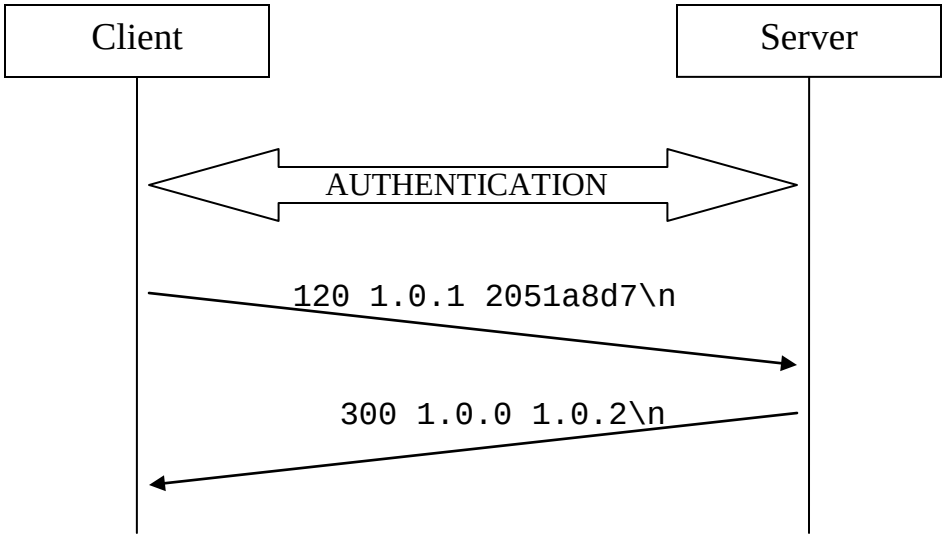
Successful simulation cancellation



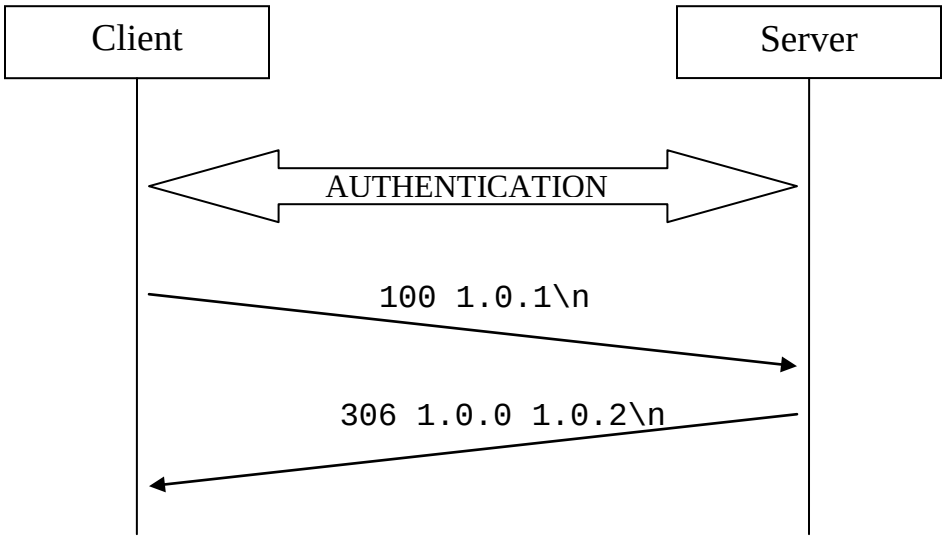
Successful SID termination



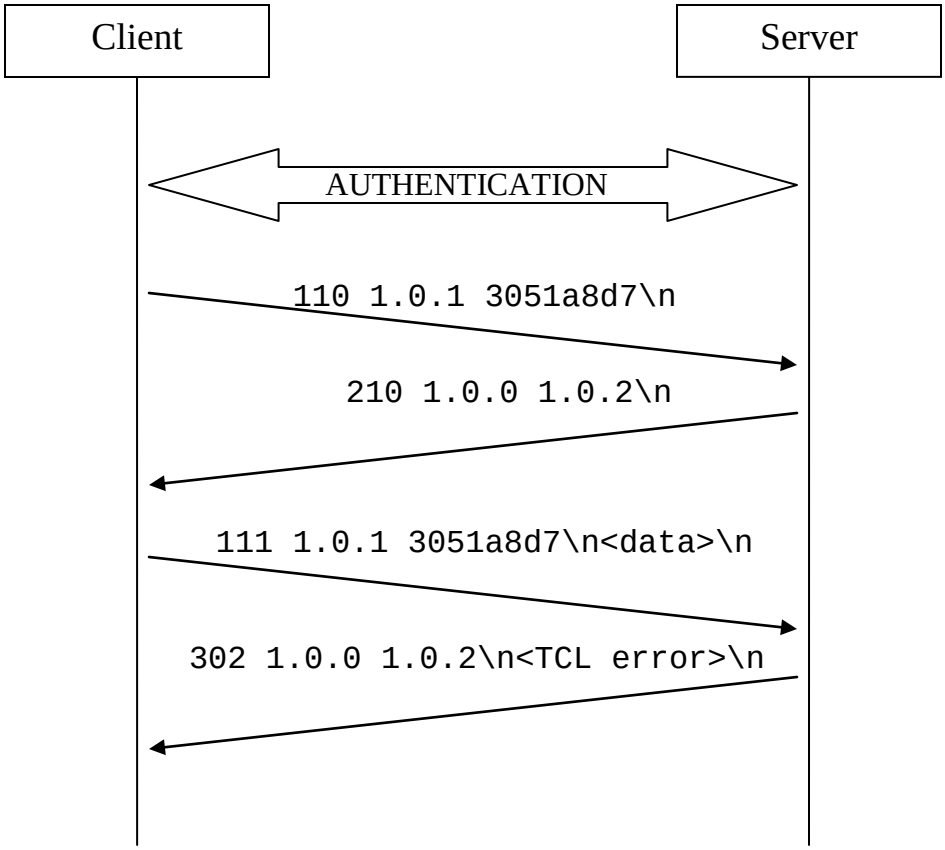
Aborted operation due to invalid authentication



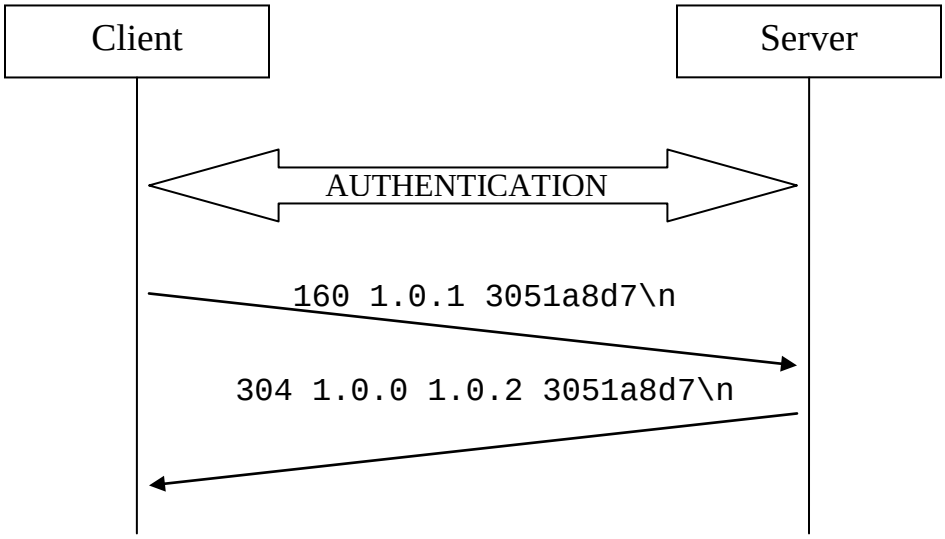
Aborted operation due to invalid SID



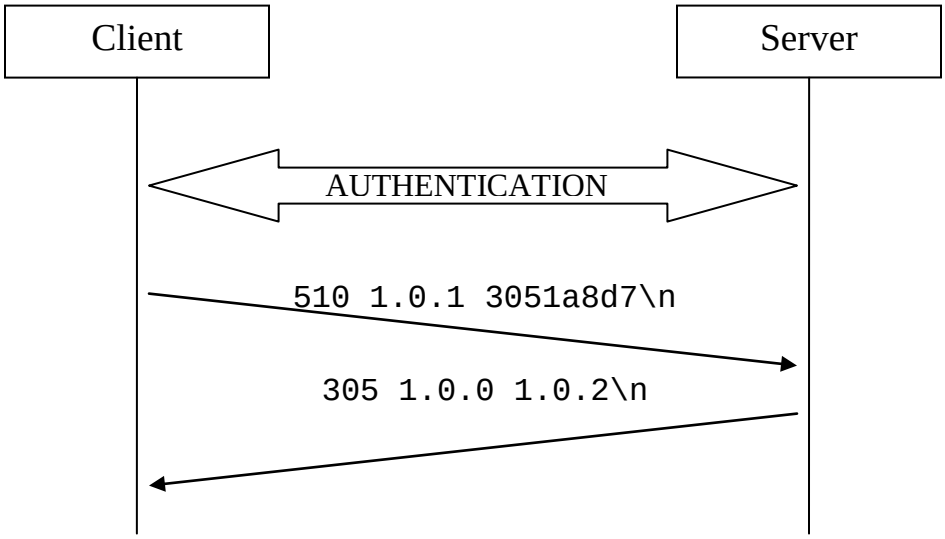
Aborted operation due to incompatible versions (server abort)



Invalid scenario received



Failed SID termination

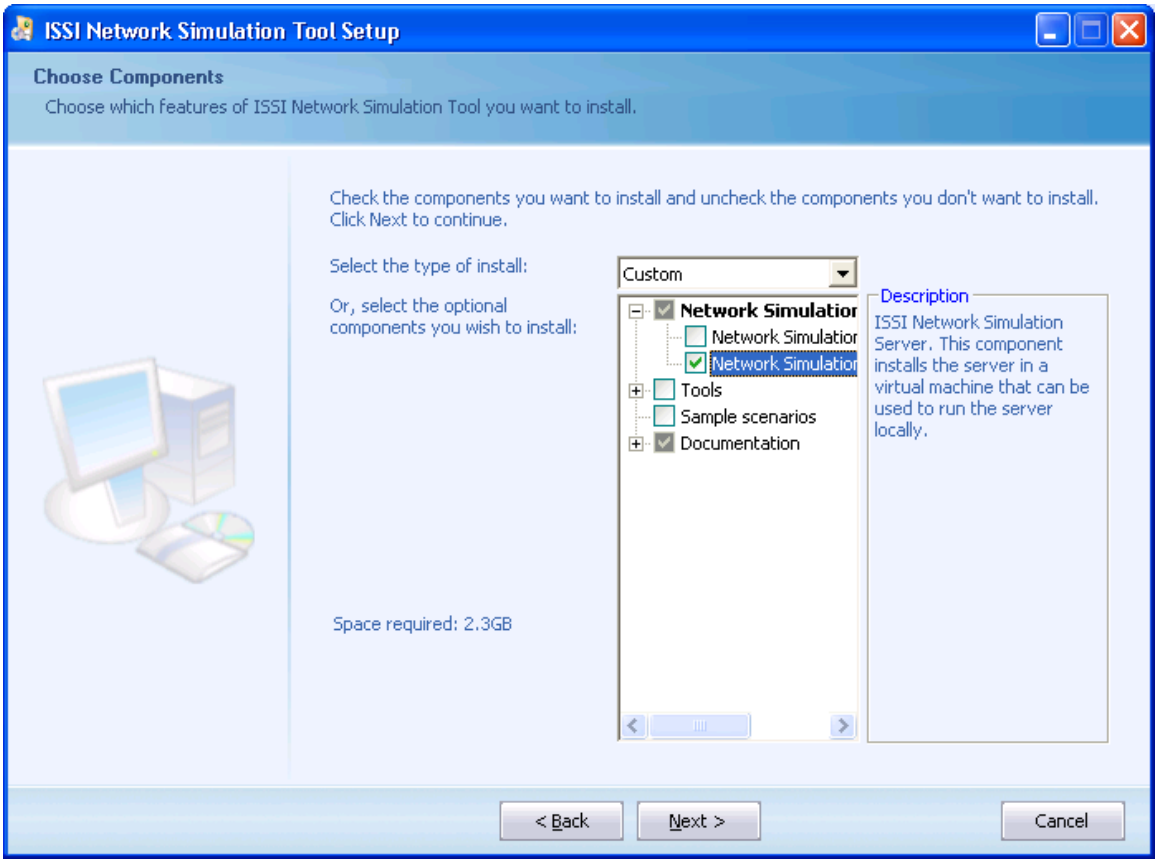


Protocol error

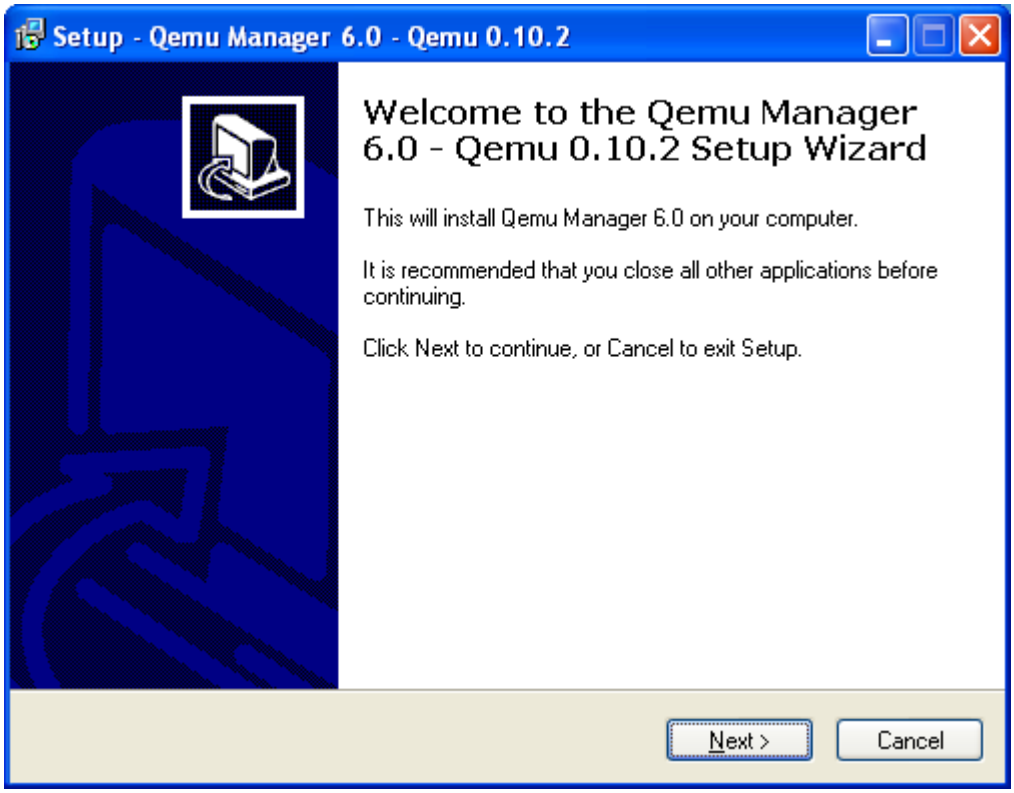
B. Running in Windows

The simulation model that the server uses to run the simulations is developed for Linux systems only. However, it is possible to install a server in a Windows system using a QEMU virtual machine. The All-in-one installer provides the QEMU software for Windows, the virtual machine disks ready to run the simulations, and a tool to configure the virtual machine parameters.

In order to install this server for windows, we need to select the 'Network Simulation server' component in the component page of the installer:

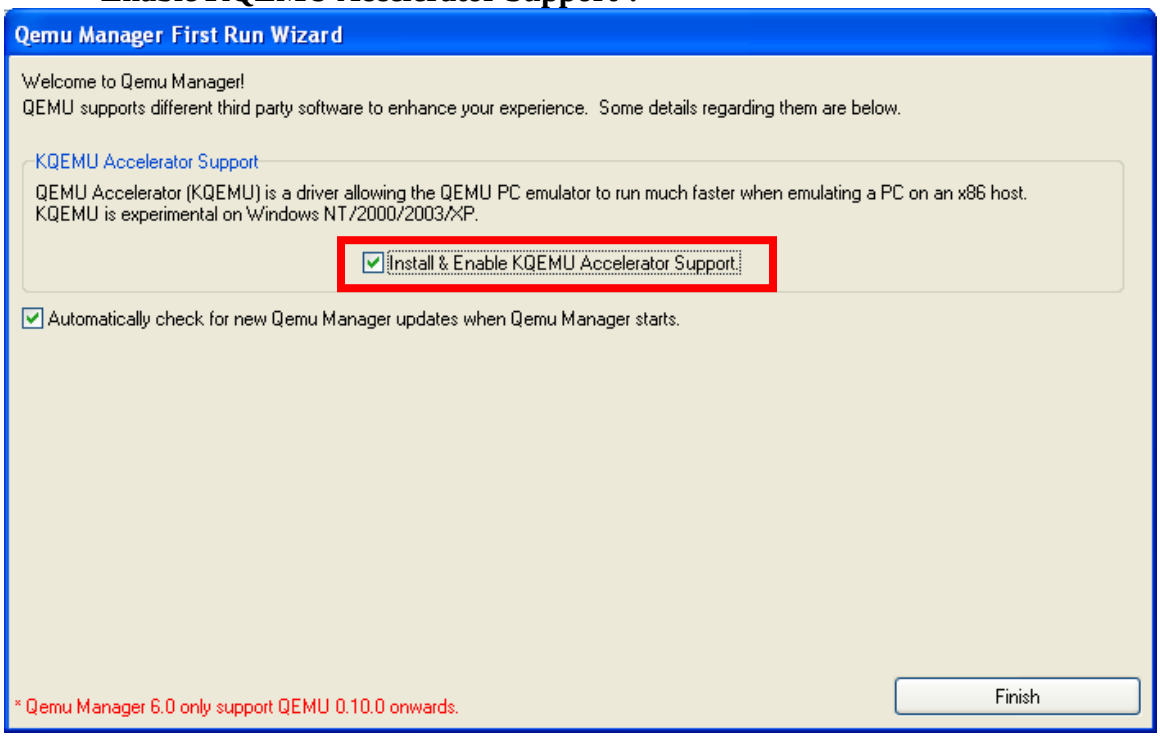


During the installation, if QemuManager is not found it will be installed to enable the management of the virtual machines (VMs):

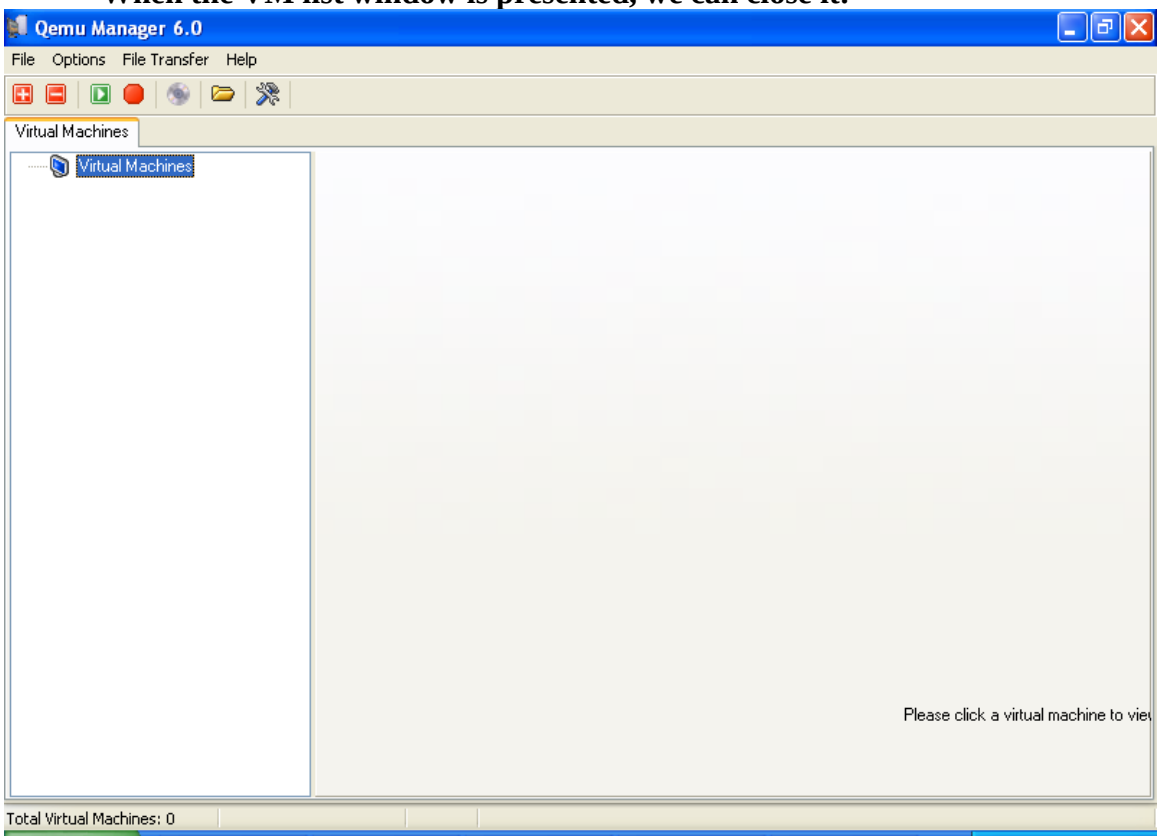


Important! When the Qemu Manager installation is over, we must launch the program:

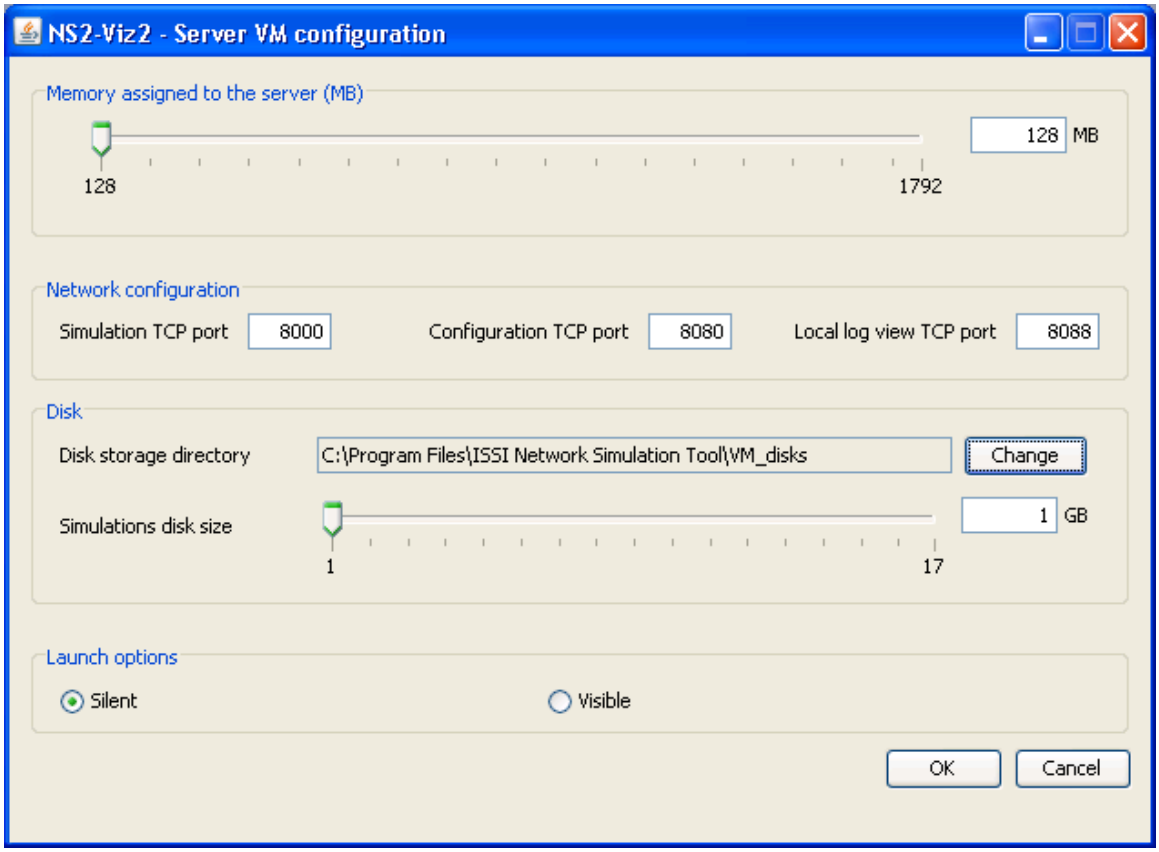
In the initial configuration screen we must check the box that says 'Install & Enable KQEMU Accelerator Support':



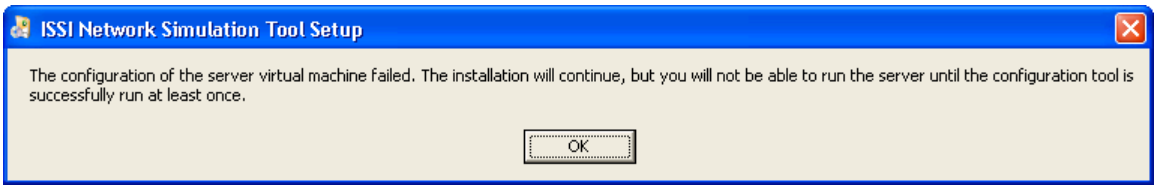
When the VM list window is presented, we can close it:



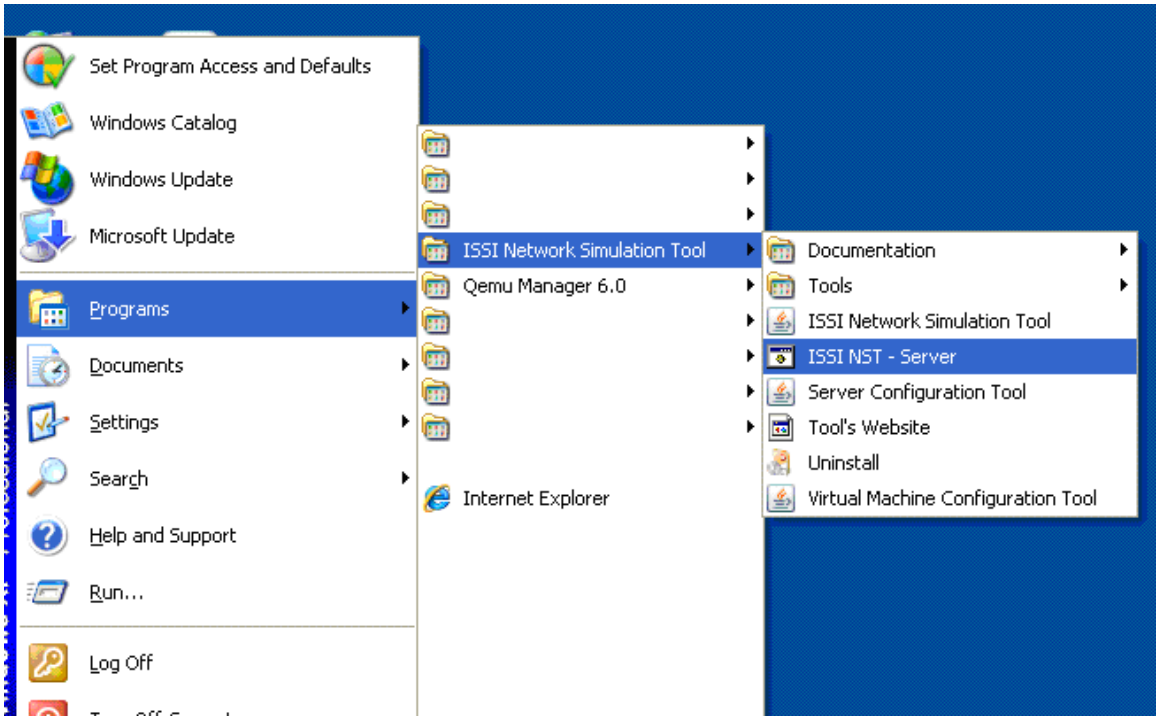
Also, the installer will launch the virtual machine configuration tool (see Section B.1):



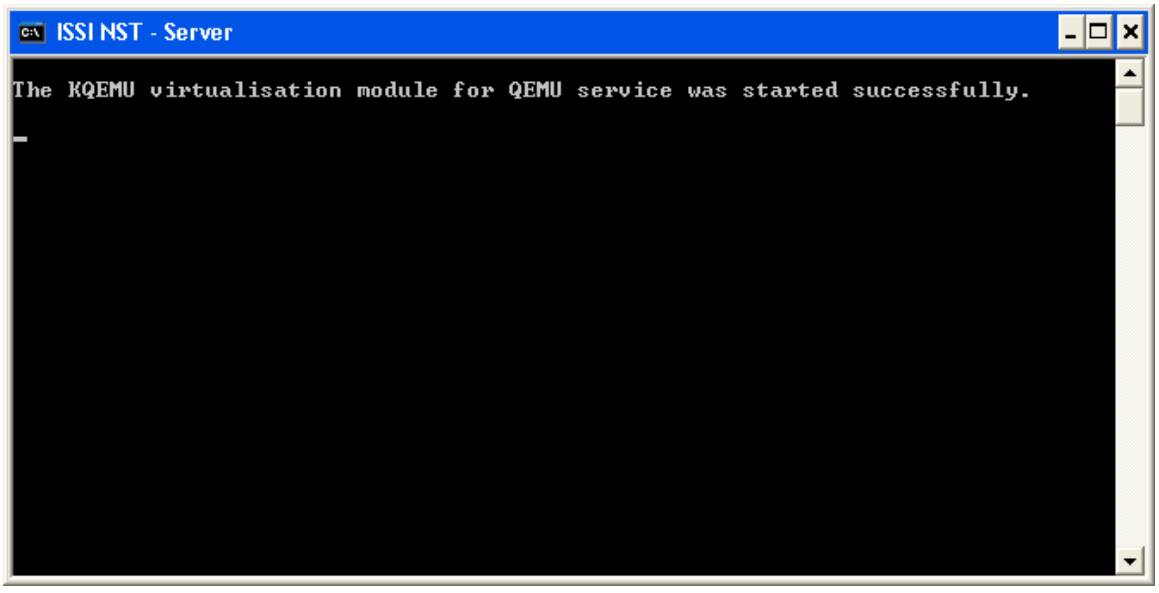
It is possible to skip the configuration at this point, but we will need to run the tool at least once before we can launch the server:



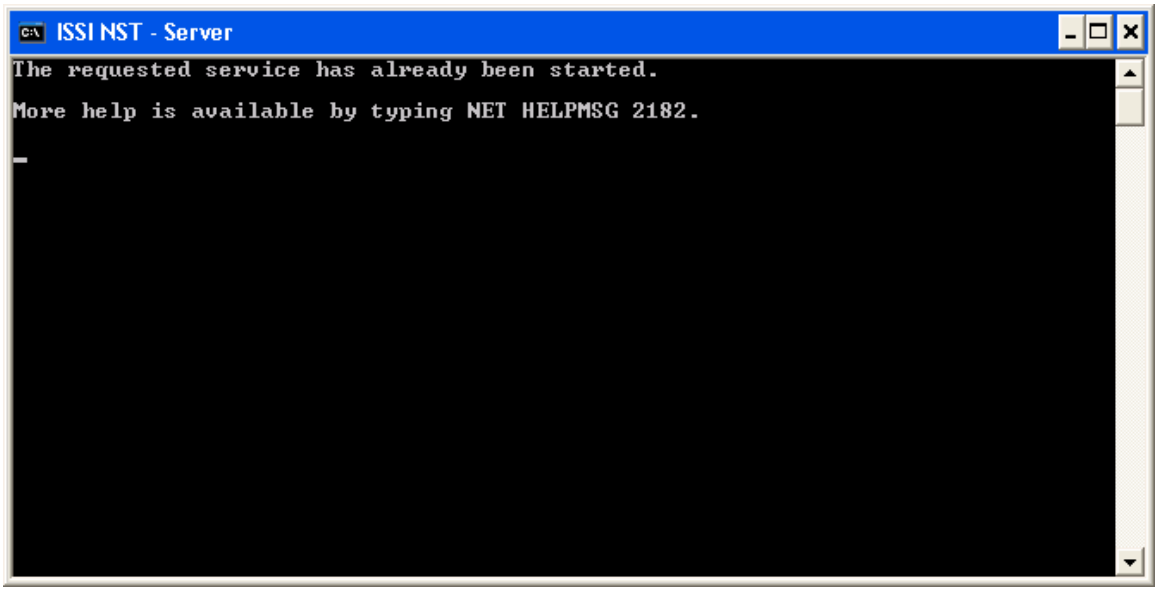
Once the installation has finished, and the configuration tool has been run at least once, we can launch the server using the `ISSI NST – Server` icon in the `ISSI Network Simulation Tool` menu:



If we chose a silent start, we will only get a window with a message about the KQEMU service starting status:

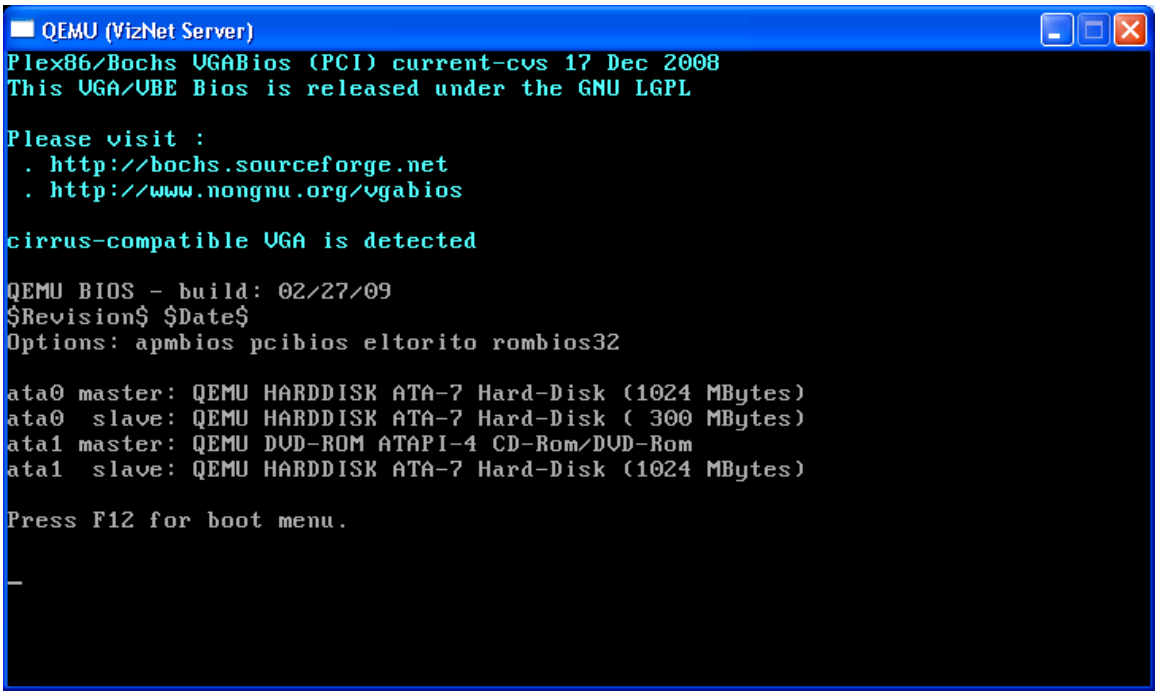


Note: This message may be an error saying that the service had been already started. This may be expected, as the service is started the first time you launch the server after a system boot:



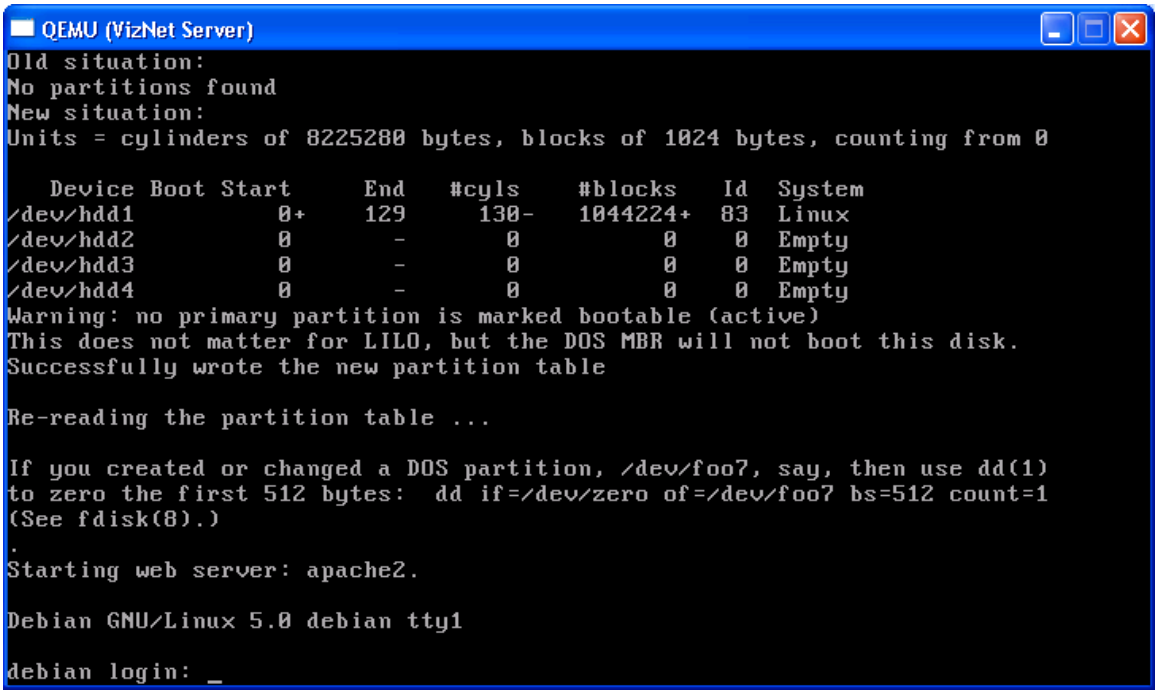
Closing this window will shut down the server (just the simulation server; the KQEMU service will keep running).

If we did not select a silent launch in the VM configuration tool, we will see an additional window with the server console:

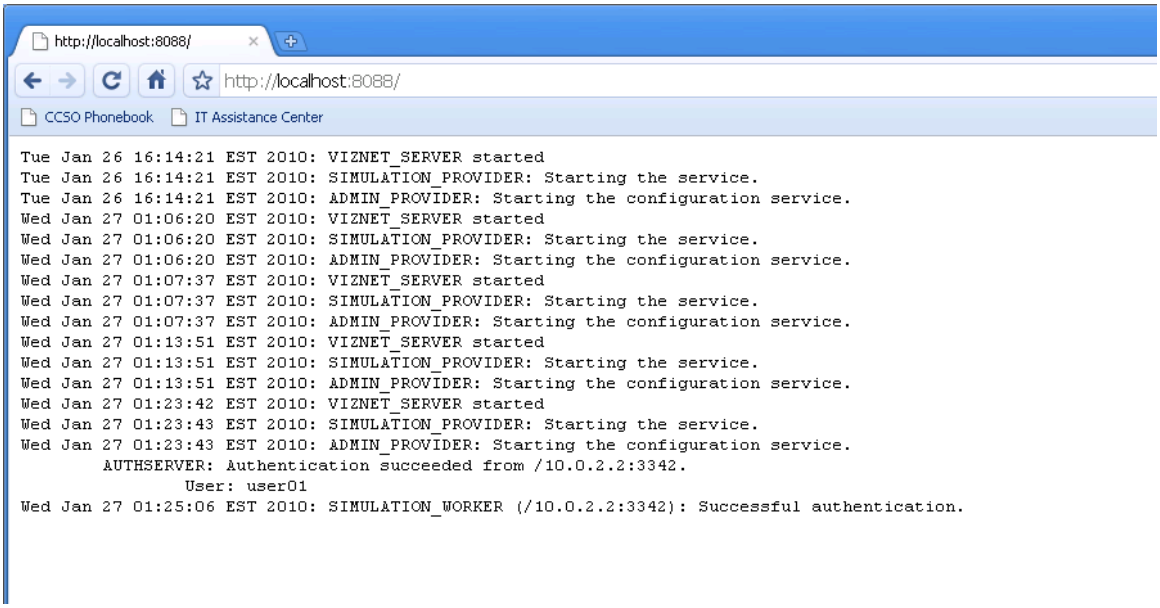


This option is recommended whenever the VM configuration is modified, especially that of the simulations virtual disk, as the system may perform maintenance operations during

the boot process, and this is the only way to verify when those operations are over and the server is available [4]:



Once the server is running, it is possible to see the server log by launching an internet browser and opening the following URL: `http://localhost:<log_port>`, where `<log_port>` is the port defined in the VM configuration tool as the Local log view port (see Section B.1).



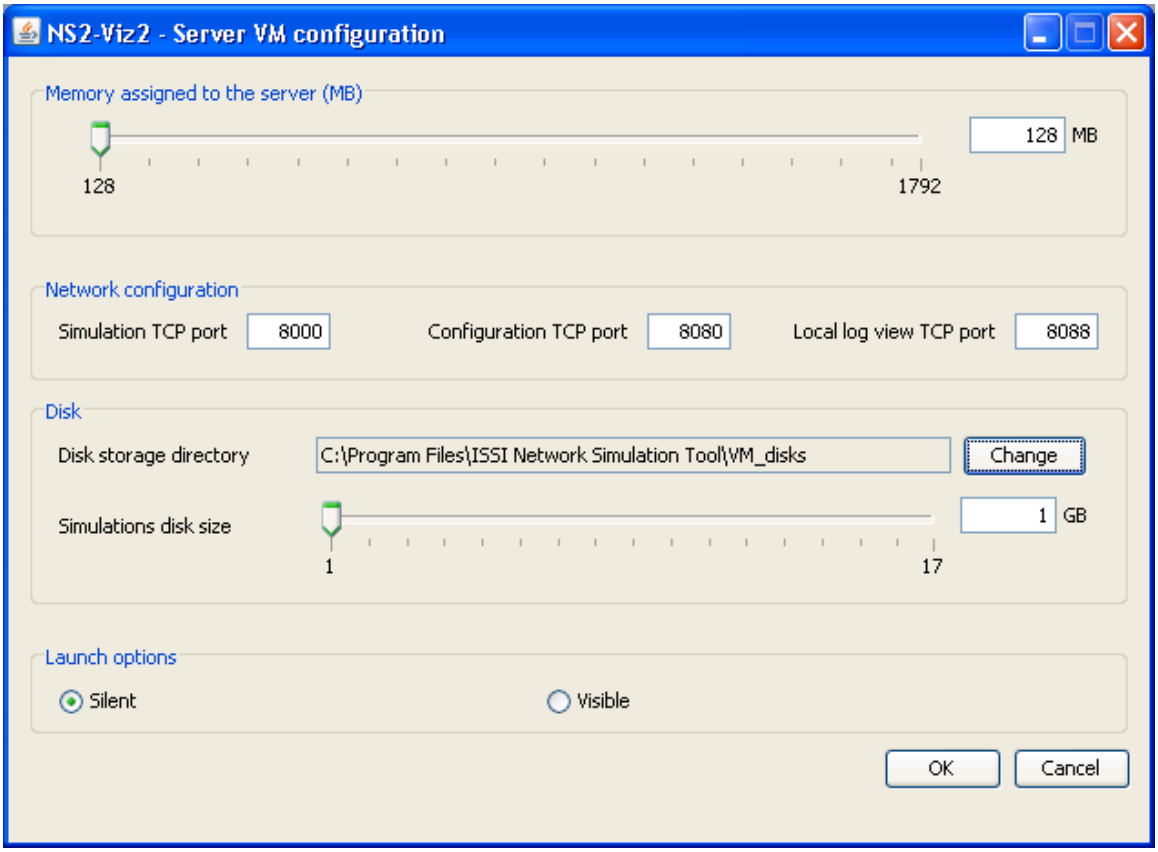
[4] As the performance of the virtual machines depends heavily on each system configuration, the time elapsed from the launch of the server until it becomes available for receiving simulation requests varies from one system to another. We recommend launching the server in visible mode the first times to get an idea of what this required time is in each system.

B.1. VM Configuration Tool

The VM configuration tool is a program that allows us to configure the parameters used to launch the virtual machine with the simulation server. We may launch it using the VM configuration tool icon:

The tool consists of a single screen where the following parameters can be configured:

- **Memory**: Amount of memory used by the virtual machine. Running the server requires at least 128 MB of memory, and the program will reserve at least 256 MB for the host operating system, which means that the minimum memory a system needs to run the server is 384 MB.
- **Network ports**: This configures the network ports to use for the simulation and configuration of the server. **WARNING!** These ports will be open and any other computer will be able to connect to them. You may want to limit the systems that may establish connections to these ports using other means (e.g., a firewall):
    - **Simulation TCP port**: The TCP port the clients will connect to for launching the simulations.
    - **Configuration TCP port**: The TCP port to connect to in order to configure the server (see Section 4.2).
    - **Local log view TCP port**: The TCP port to use for serving the server's log as a web page. This allows to see the log just by opening the URL `http://localhost:<port>` in an internet browser.
- **Disk storage directory**: Directory where the virtual disk with the simulations is stored.
    - **Size of the simulations disk**: Size (in GB) of the disk that will store the directory where the simulations are run. 1 GB is the minimum required.
- **Launch options**: We can choose between launching the server in 'Silent' mode or 'Visible' mode.


NS2-Viz2 - Server VM configuration
Memory assigned to the server (MB)
128
MB
128
1792
Network configuration
Simulation TCP port
8000
Configuration TCP port
8080
Local log view TCP port
8088
Disk
Disk storage directory
C:\Program Files\ISSI Network Simulation Tool\VM_disks
Change
Simulations disk size
1
GB
1
17
Launch options
Silent
Visible
OK
Cancel