Load Testing 2U Rockbochs System

The purpose of this paper is to discuss the results of load testing the 2U system from Rockbochs. The system in question had the following hardware:

- Intel® Celeron® Processor E1400 (512K Cache, 2.00 GHz, 800 MHz FSB)
- Super Micro X7SLM-L Motherboard
- 1GB Crucial PC2-5300 RAM
- WD800JD Western Digital Hard drive
- Sangoma A200DE+2 Remora boards with 6 FXO Modules (12 FXO channels)…A20006DE

The system was tested using the following software:

- PBXnSIP 3.3.1.3177
- NBE Express 2.0 with included Wanpipe-6.0.9.12 for Windows
- Windows XP Professional, Service Pack 3
- CentOS 5.3 w/2.6.18-128.el5
- Wanpipe-3.4.1 for Linux
- Dahdi-linux 2.1.0.4 and Dahdi-tools 2.1.0.2
- Asterisk-1.6.0.9
- Sipp-3.1

This testing was performed by:

Konrad Hammel

e: konrad@sangoma.com

Software Engineer/Field Application Engineer

Sangoma Technologies Inc.

SANGOMA

Table of Contents

50 McIntosh Drive, Suite 120, Markham, ON L3R 9T3, Canada • t. 1 905 474 1990 • toll free in North America 1 800 388 2475 • f. 1 905 474 9223 • www.sangoma.com

Notes Regarding Testing Procedure

- For the purpose of these tests a fail occurs when the UAC or UAS observed messages arriving too late or an unexpected message arrived.
- The system used to generate load and the system under test were isolated on their own network using a Linksys SD208 switch and CAT6 Ethernet cabling.
- The system that was used to generate and receive calls was able to handle 400 calls, at 50 calls/sec, while maintaining a cpu idle >50%. Calls were generated using Sipp (UAC), sent to Asterisk, and the received by Sipp (UAS). The testing of the load generator box shows that it is not a limiting factor in the tests.

Test 1: Base Line PBXnSIP – SIP to SIP

Test Description

To start the testing a baseline needed to be established of the number of calls PBXnSIP can handle without seeing any errors in Sipp. As mentioned in the notes, any error messages or retransmissions were considered a fail.

For this test, and the other tests, a single system was used to generated and receive the calls. The calls were generated using the Sipp built-in UAC scenario. The calls were received by Asterisk which was registered as an account in PBXnSIP. Figure 1 shows the test scenario.

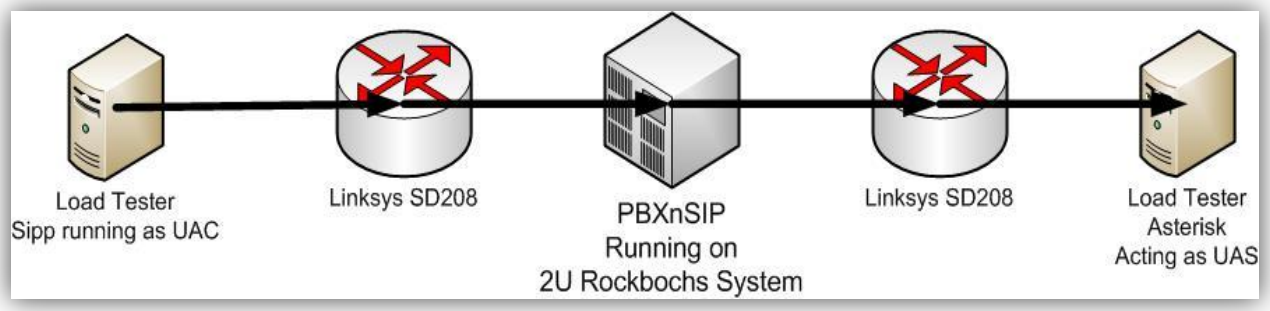

Figure 1 – Base Line Test Scenario

The Sipp UAC client was not transmitting any RTP audio packages but Asterisk was setup to answer all calls and connect the call to the "echo" application to generate an RTP stream.

Both the incoming Sip trunk and Sip account in PBXnSIP we set to use U-law encoding only.

Test Procedure

First, the proper operation of PBXnSIP was confirmed by placing a call from a soft phone to PBXnSIP which then routed the call to the Asterisk system and clear audio was confirmed.

Sipp was then started using the initial parameters of 100 calls at 10 calls per second with call duration of 15 seconds. Within the first minute Sipp was reporting retransmission as well as several "temporary unavailable" messages.

The max number of calls was then lowered to 80 calls at 10 calls per second with call duration of 15 seconds. Again it was less than a minute before calls were being rejected with "temporary unavailable"

The number of calls per second was then lowered to 8, max calls was set to 80 with call duration of 15 seconds. After 20 periods Sipp had not reported any retransmissions, time outs or unexpected messages. The system load on the PBXnSIP system was showing CPU idle dips down to less than 50%

when 8 calls were being established at once. Figure 2 shows the Windows Performance monitor result for this test.

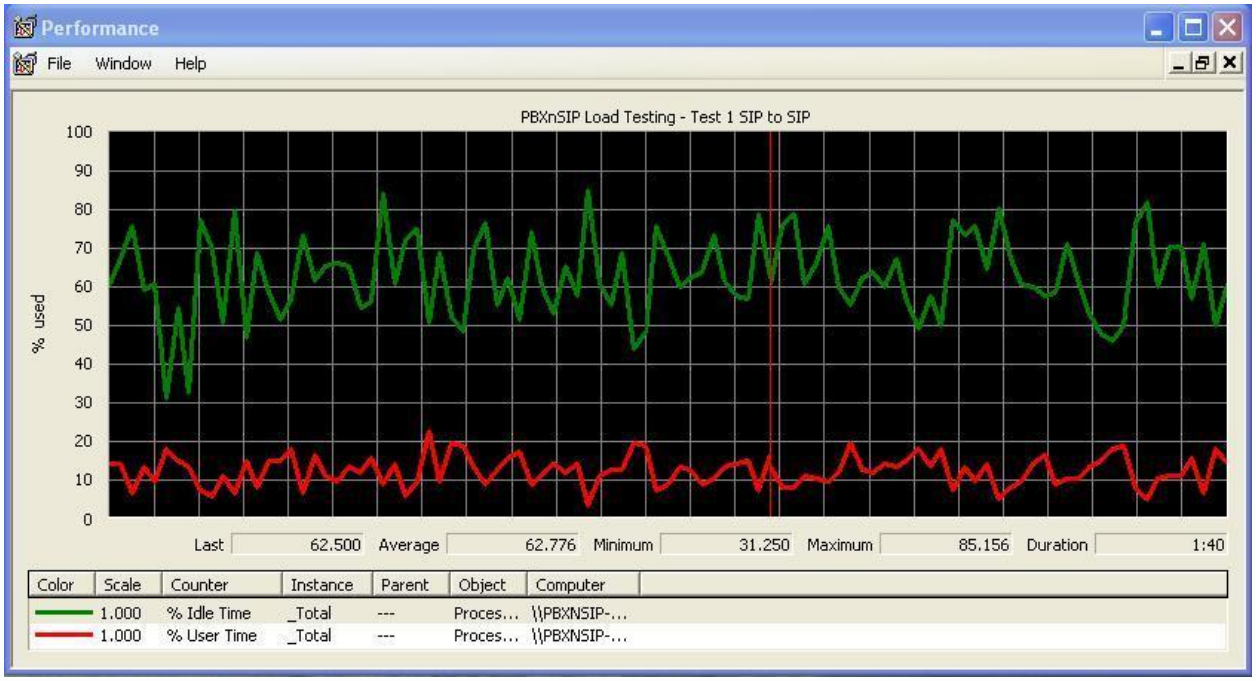

Figure 2 – Test 1 - 80 calls, 8cps, 15 seconds

The number of active calls was then increased by increments of 10 until Sipp reported errors. At 110 calls the first error message (retransmissions) were reported. Figure 3 shows the load results for this test run.

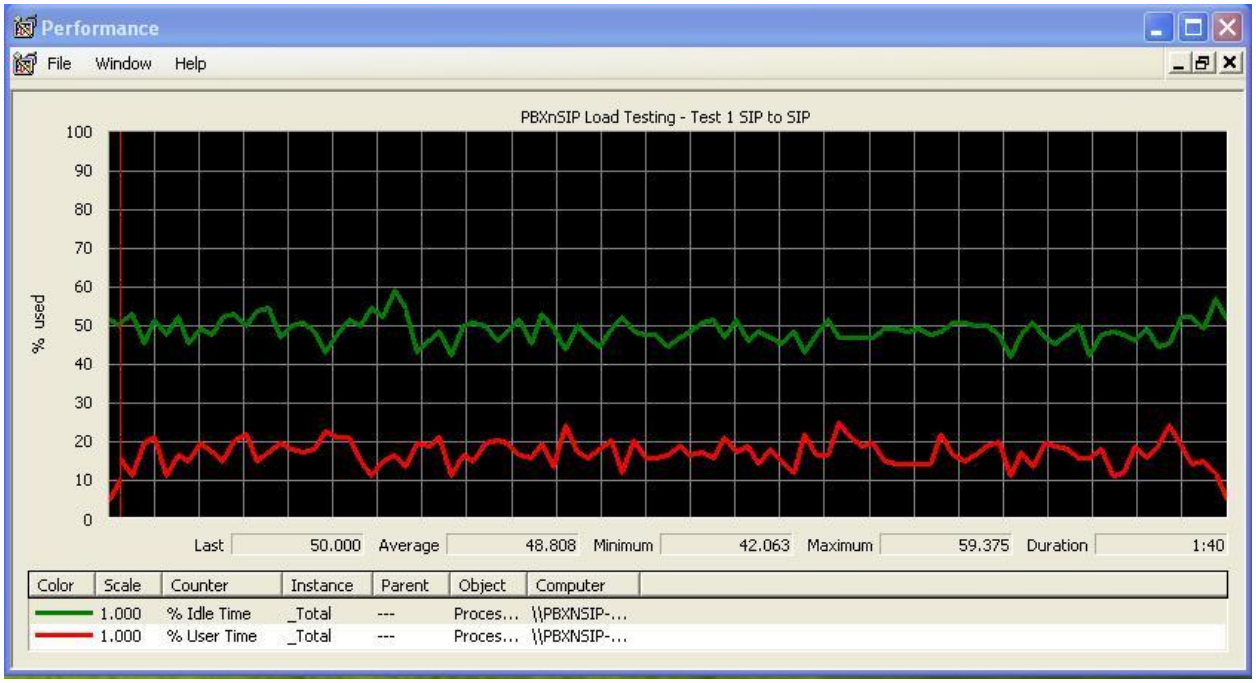

Figure 3 – Test 1 – 110 calls, 8cps, 15 seconds

The call duration was the lowered to 13 seconds, this would keep maximum call load active on the system when the max calls is 110 at 8 cps. After 36 cycles no errors were observed. Figure 4 shows the load results for this test.

50 McIntosh Drive, Suite 120, Markham, ON L3R 9T3, Canada • t. 1 905 474 1990 • toll free in North America 1 800 388 2475 • f. 1 905 474 9223 • www.sangoma.com

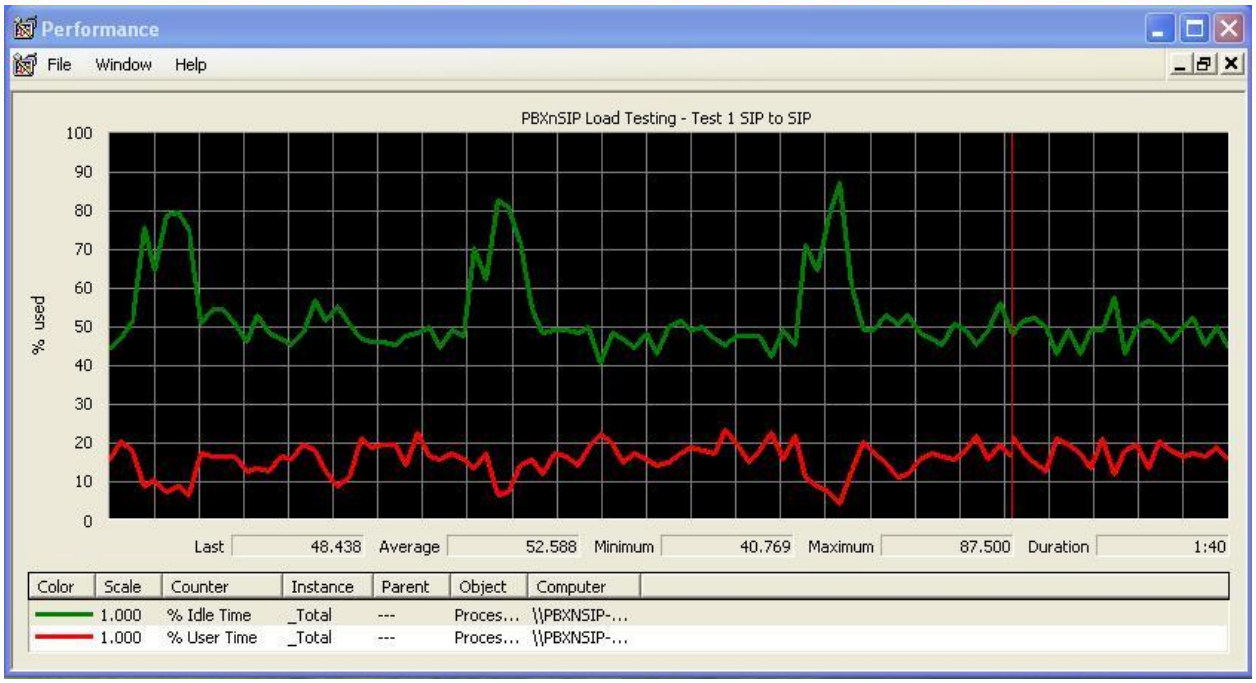

Figure 4 – Test 1 – 110 calls, 8 cps, 13 seconds

Results

The base line testing shows that PBXnSIP running on the 2U Rockbochs server can easily handle 110 calls at 8 cps. Increasing the number of calls per second to 10 resulted in calls being rejected due to lack of processing power.

Test 2: PBXnSIP + NBE2.0 w/ A20006DE TDM-SIP

NBE-2.0 Installation

For this test NBE 2.0 was installed on the system and connected to an A20006DE (a200D with 2 Remoras running 6 FXO modules).

The installation of NBE-2.0 is quite simple and only took a couple of minutes. Before starting the installation of NBE-2.0 make sure to uninstall all previous copies of NBE and Sangoma Windows drivers (make sure to do a reboot as well to fully confirm complete removal).

To start the installation of NBE-2.0, double click on its installer. It is best to choose the default options that installer provides for you. The installer will install the NBE software as well as the correct Sangoma driver. During the install of the Wanpipe drivers you will be prompted with the following window:

Figure 5 – Wanpipe Driver install

Click the “cancel” button to continue. Sangoma drivers are not “plug and play” and therefore and cannot be installed by the Windows “plug and play” driver installer. Clicking “cancel” will allow the Sangoma driver installer to finish the driver install.

Once NBE and the Sangoma Wanpipe drivers are installed the installer will prompt you to restart the system to continue. When Windows restarts NBE will automatically launch your default web browser and go to the NBE Gateway Manager’s Quick Setup tab; your screen should show the following:

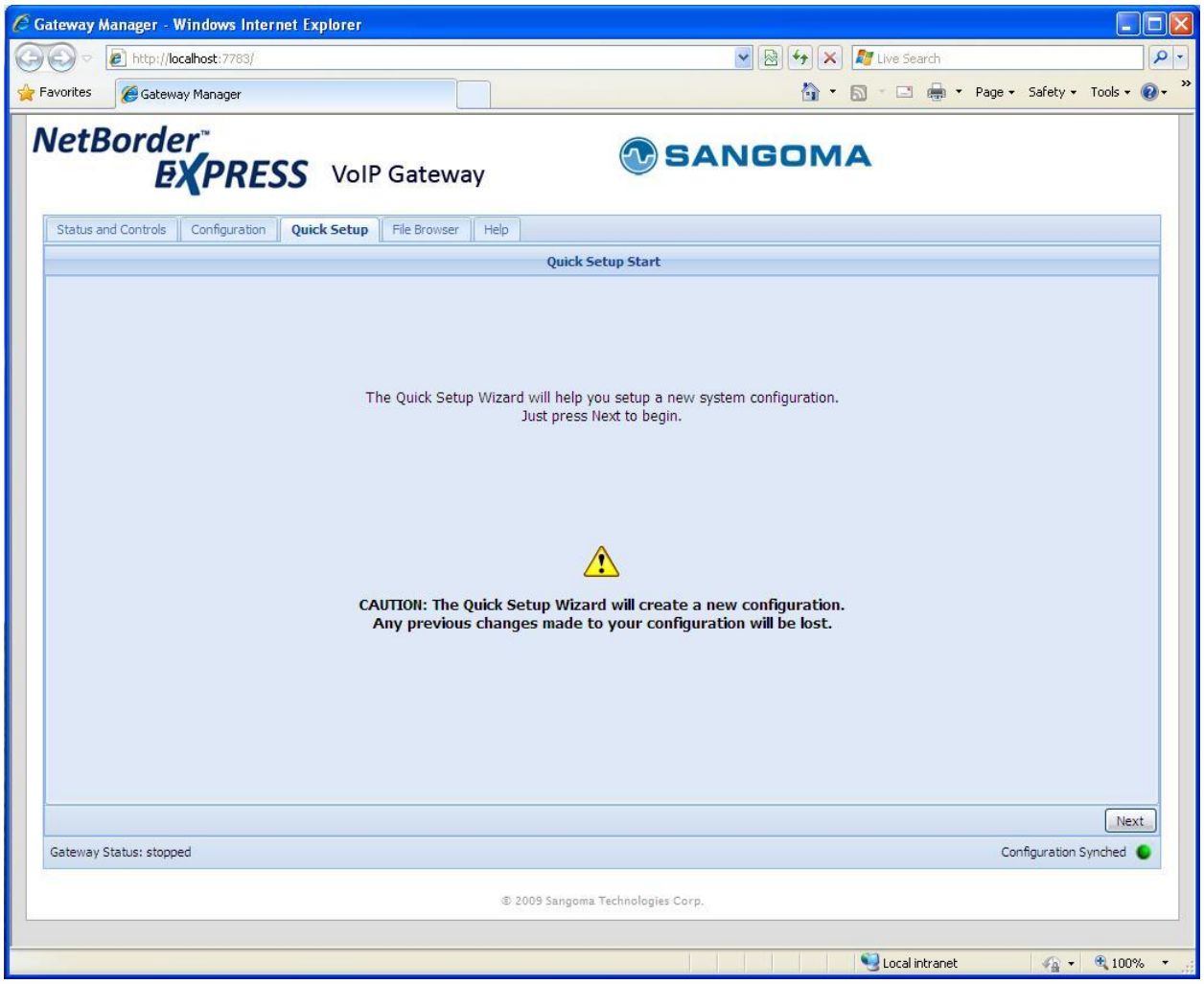

Figure 6 – NBE Gateway Manager Quick Setup Tab

Click the “next” button to start configuring your gateway.

The next screen you will see is where you inform NBE of what Sangoma cards you would like to use with the gateway.

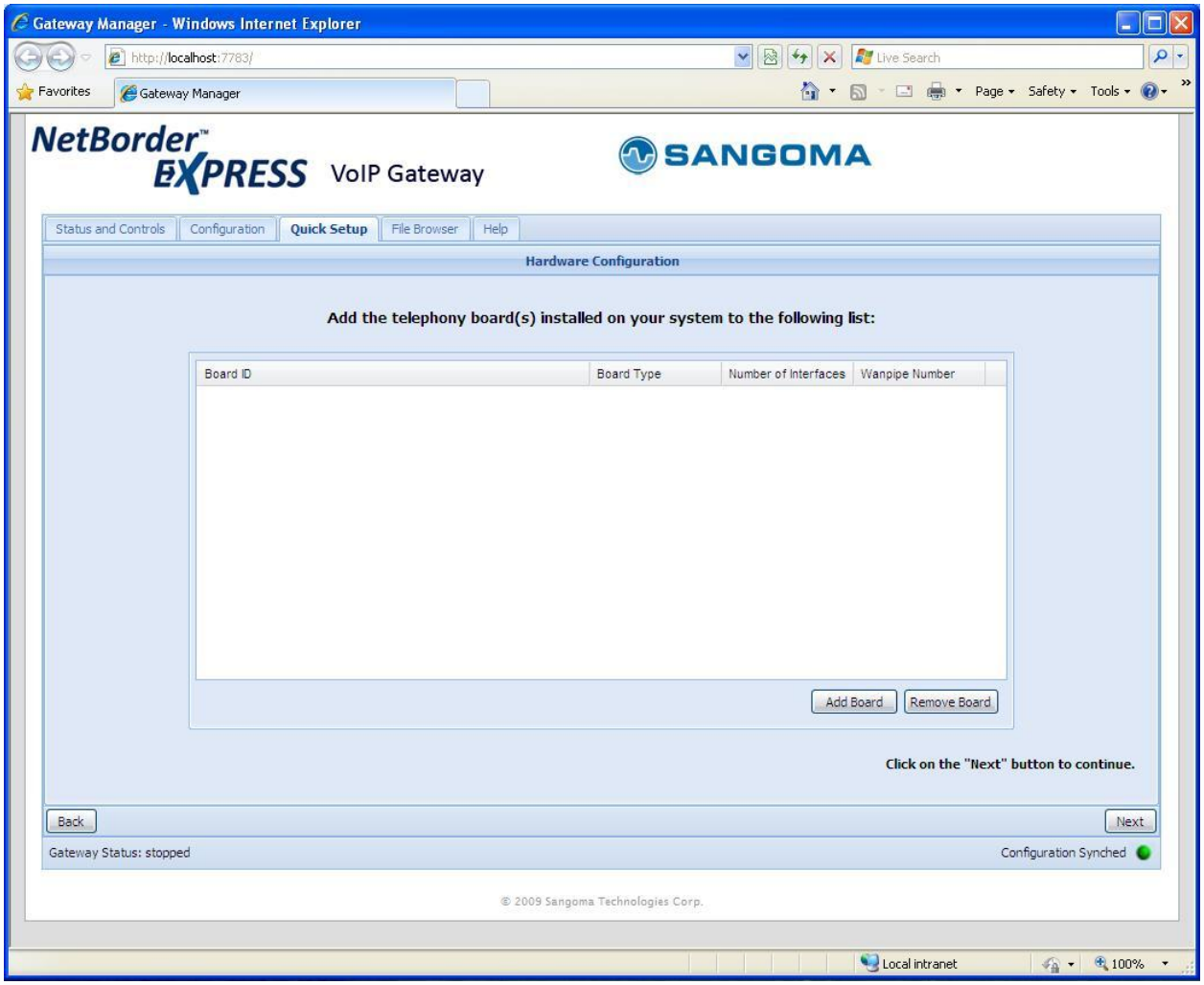

Figure 7 – NBE Quick Setup Hardware Configuration

Click the “add board” button to add the first card. You will then be prompted to select the type of card and in the case of an analog card the number of modules present.

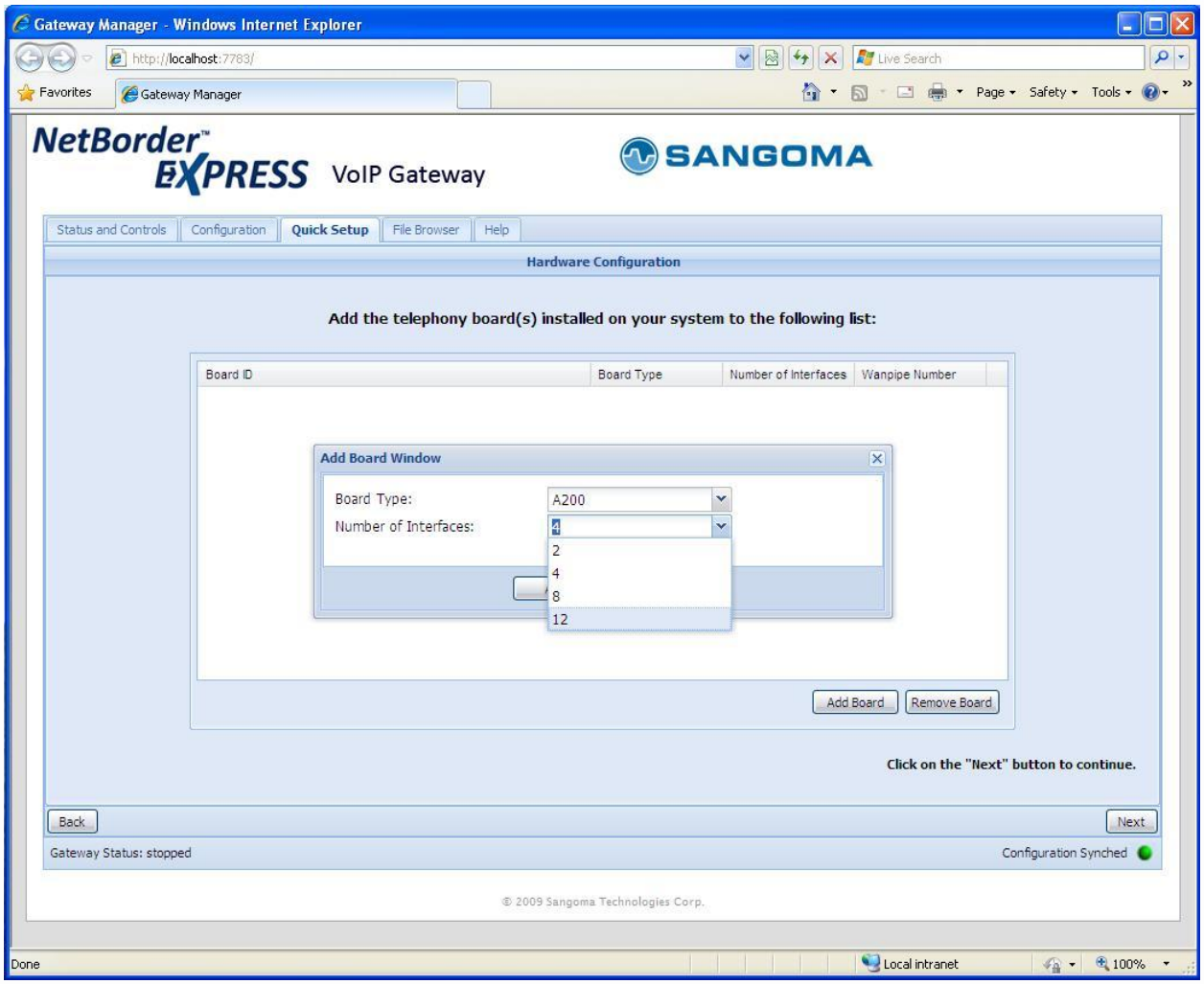

Figure 8 – NBE Quick Setup Hardware Configuration 2

In the case of this testing an A200 card along with 12 FXO channels (6 modules). Once you have selected the card (and number of modules) click the “ok” button to continue. You will now go back to the hardware configuration screen, click the “next” button to continue.

The next screen requires you to select options specific to your card, in the case of the analog card you will see the following screen.

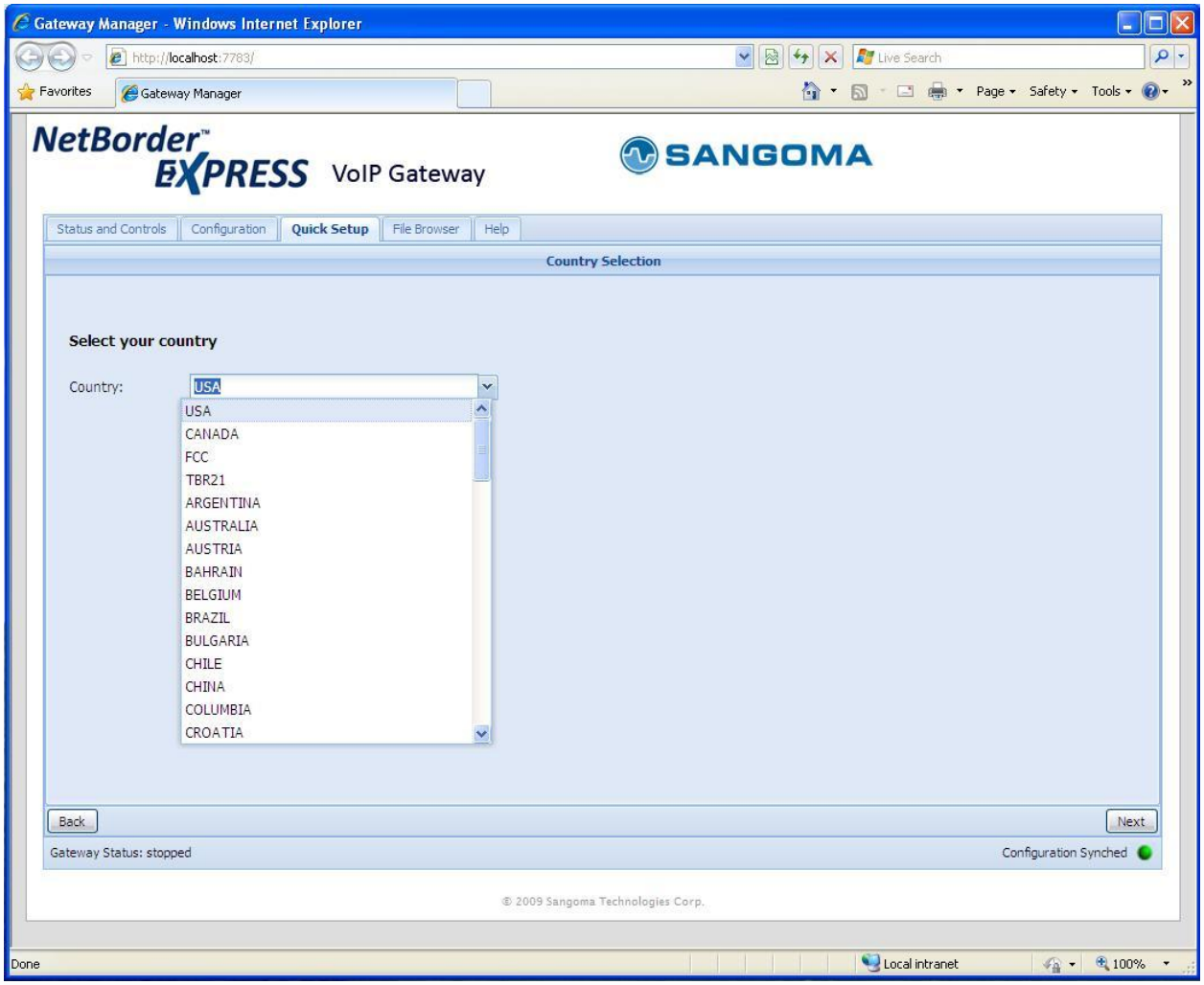

Figure 9 – NBE Quick Setup Analog Country Selection

From the drop down box you should select the country setting required for the location that his installation will be used in. For the purpose of these tests the country setting was set to USA. Click the "next" button to go the final screen.

The last screen allows you to change the IP address and port that calls from the PSTN will go to and the port that NBE will listen for outgoing call requests. Since NBE will be connecting to PBXnSIP in the same system the Default SIP address and port needs to be changed to 127.0.0.1:5060.

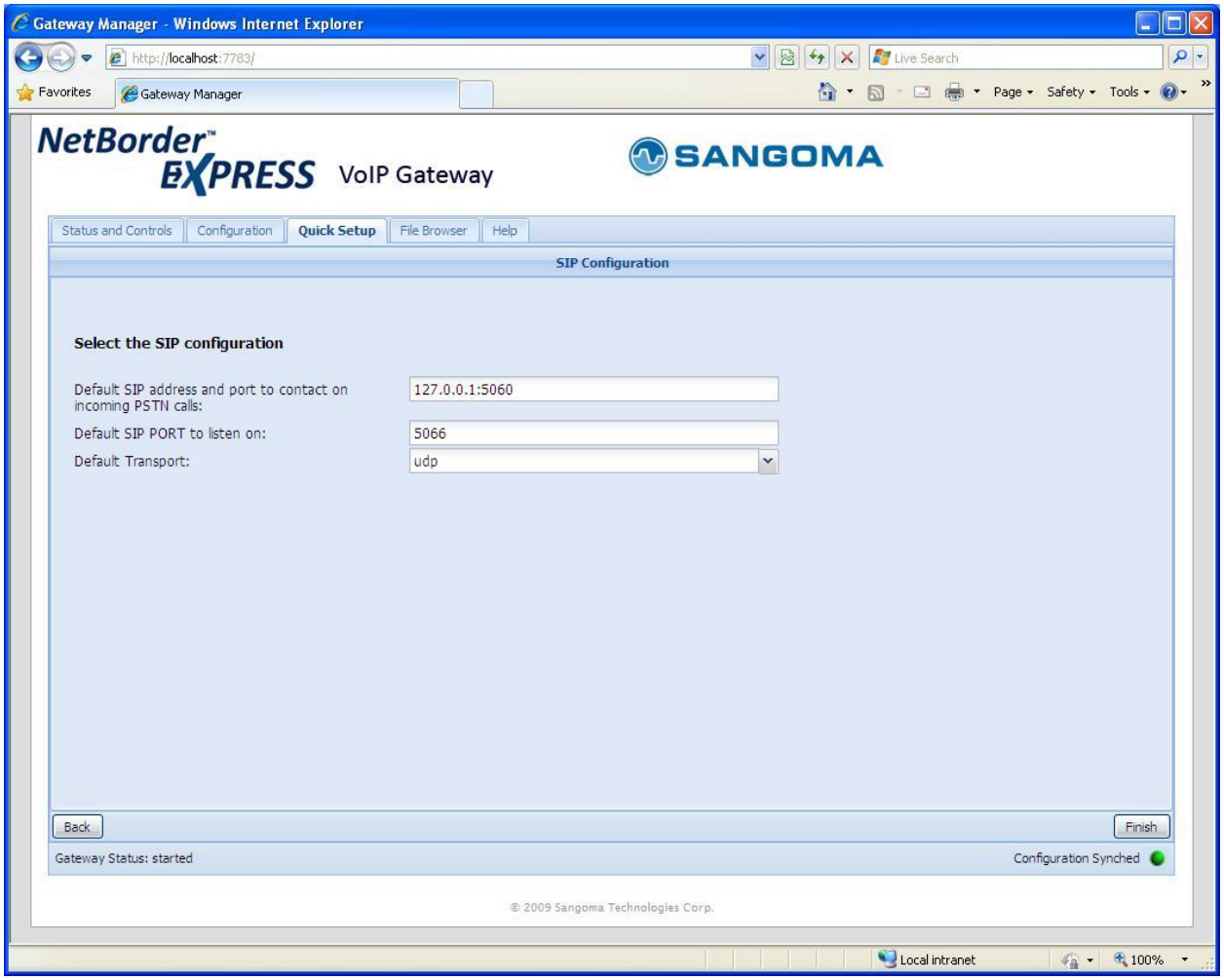

Figure 10 – NBE Quick Setup Sip Configuration

Click the "finish" button to complete the PSTN configuration and be prompted to start NBE.

PBXnSIP Configuration

The configuration of PBXnSIP is also quite simple. Start by logging into the PBXnSIP manager using the domain username and password. Click on the "trunks" tab to start adding a new trunk. Give the new NBE trunk an appropriate name and set it as a SIP Gateway as per Figure 11.

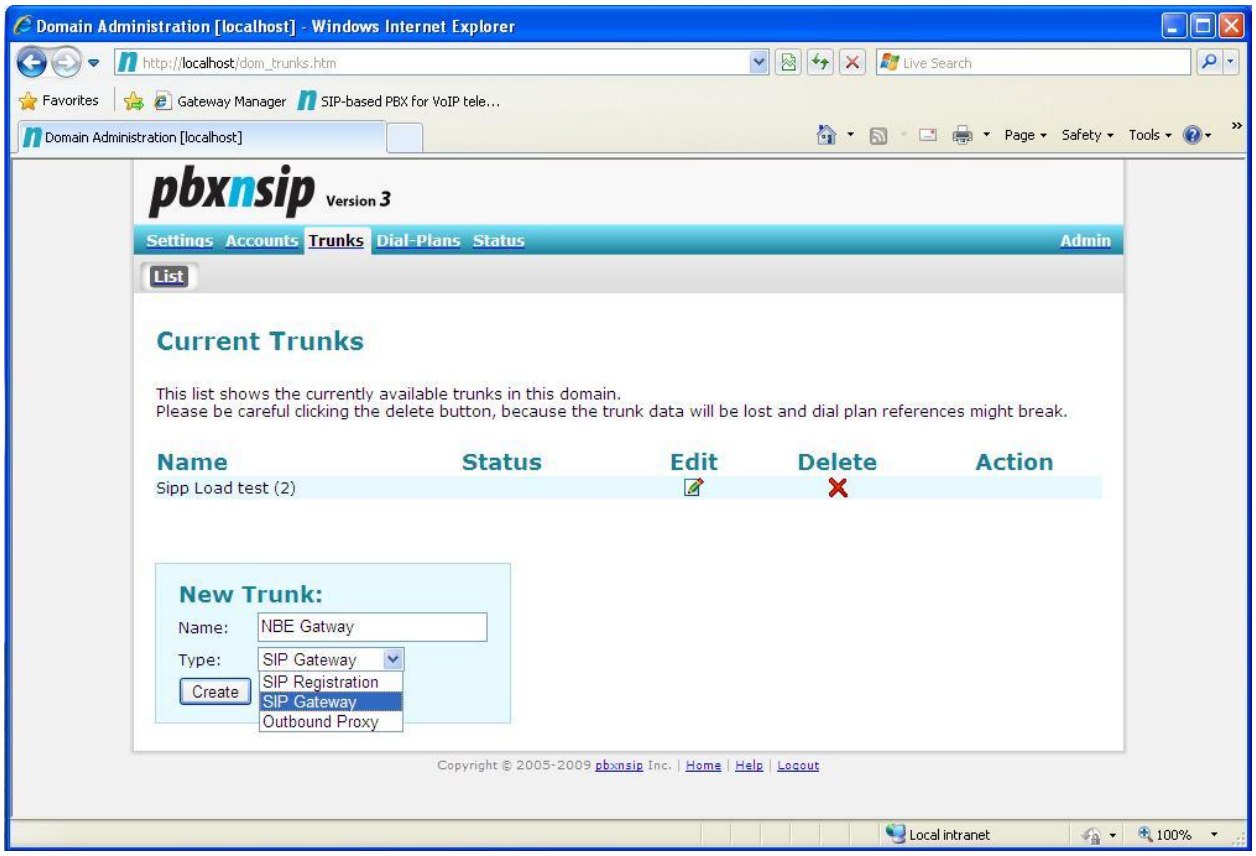

Figure 11 – PBXnSIP Trunks Tab

Click the “create” button to add the trunk. The screen will reload and you will see that the new trunk is now part of the list. Click the “edit” icon to finish configuring the new NBE trunk.

On the Trunk settings tab there are only 2 options that need to be added. First set the “outbound proxy” to the IP address and port that NBE is listening on….127.0.0.1:5066. Next you need specify the default extension to send incoming calls to; for this test I set this to the account number that I attached my UAS client to.

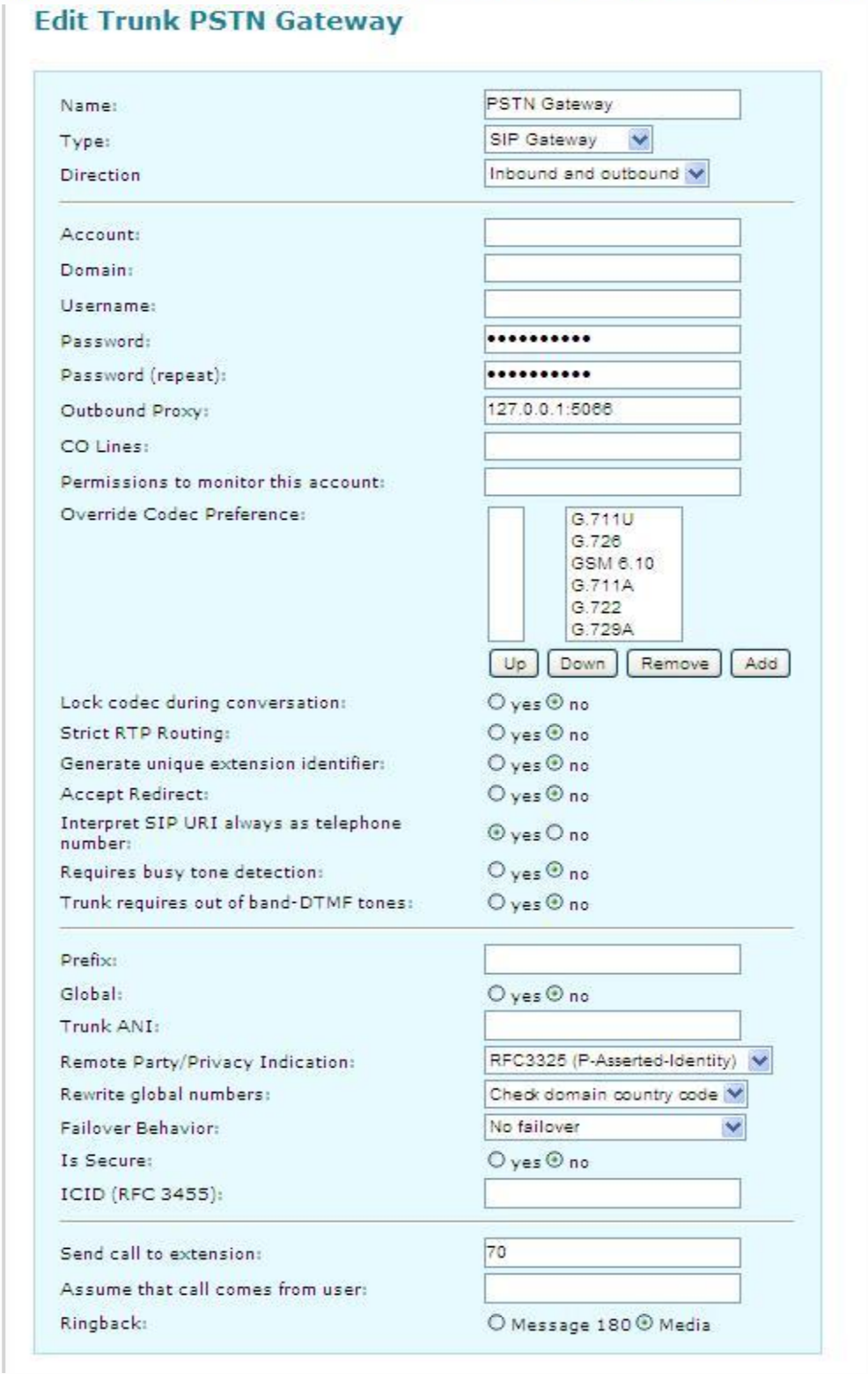

Figure 12 – PBXnSIP Trunk Settings

Test Description

This test required the analog channels of the A200 card in the PBXnSIP system to be loaded. This was done using another server running an A400 with 12 FXS channels and Asterisk.

A custom Sipp scenario was used to generate the calls and send them to a local Asterisk server. The difference between the custom UAC scenario and the built-in scenario was a small delay at the end of the SIP call. It was found that the system generating load was sending calls to quickly to a channel analog that had just been hung up causing the remote side to not detect the new call.

The Asterisk server then sent the calls out FXS interfaces to the PBXnSIP system running NBE. PBXnSIP was set to send all calls to a UAS client.

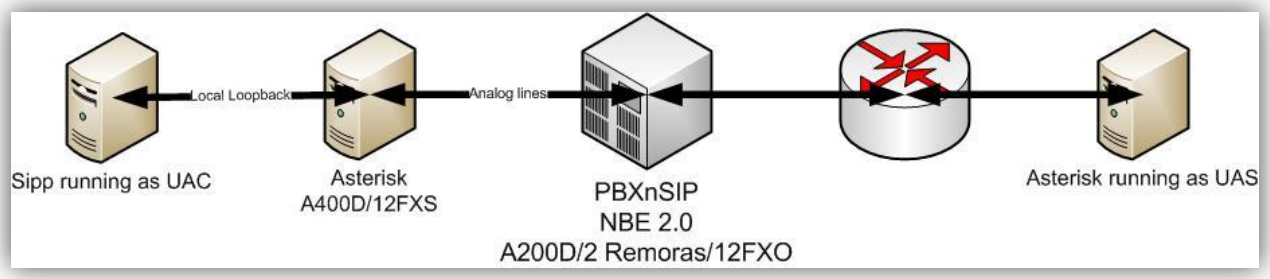

Figure 13 – TDM to SIP Test Scenario

The Sipp UAC client was not transmitting any RTP audio packages but Asterisk was setup to answer all calls and connect the call to the "echo" application to generate an RTP stream.

Test Procedure

Before loading the system with calls from the FXO interface, the baseline load on the on system can be seen in figure 14. Figure 14 shows the system load with NBE-2.0 and PBXnSIP running but not processing any calls.

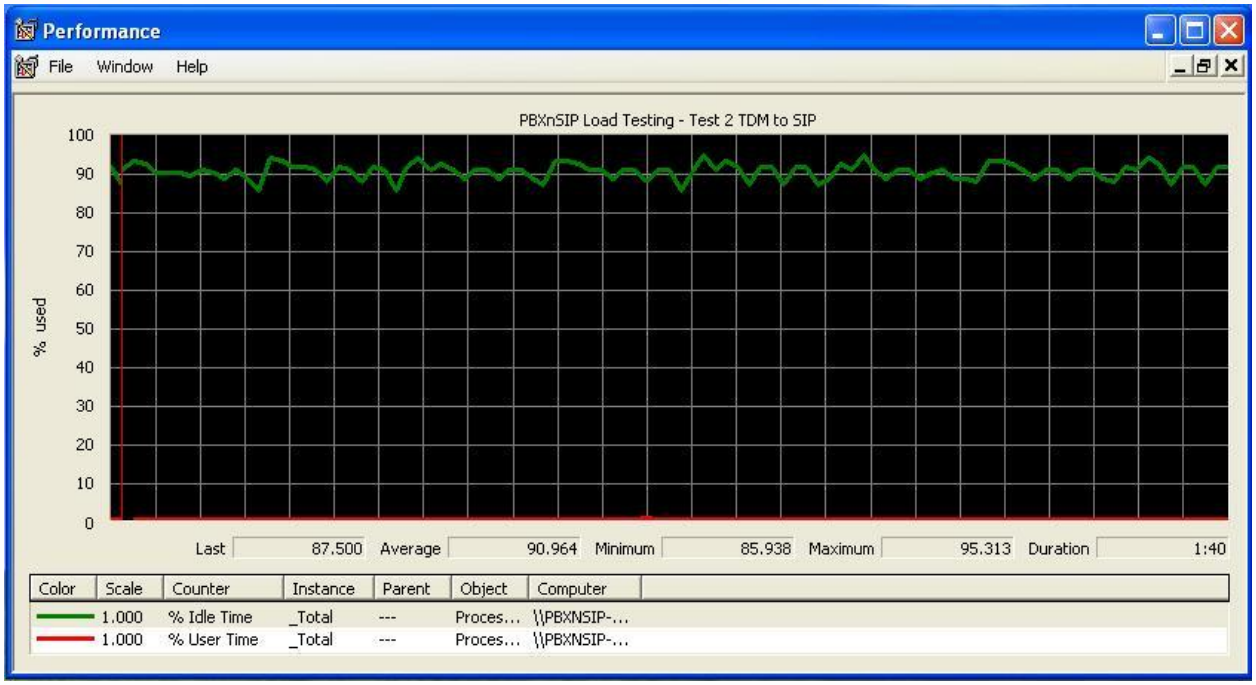

Figure 14 – PBXnSIP + NBE-2.0 No call load

Since earlier testing had shown that PBXnSIP on the server worked best at 8 cps (with 110 calls total). This was used a starting point for load testing. The first test scenario run was 12 calls at 8 cps with duration of 10 seconds.

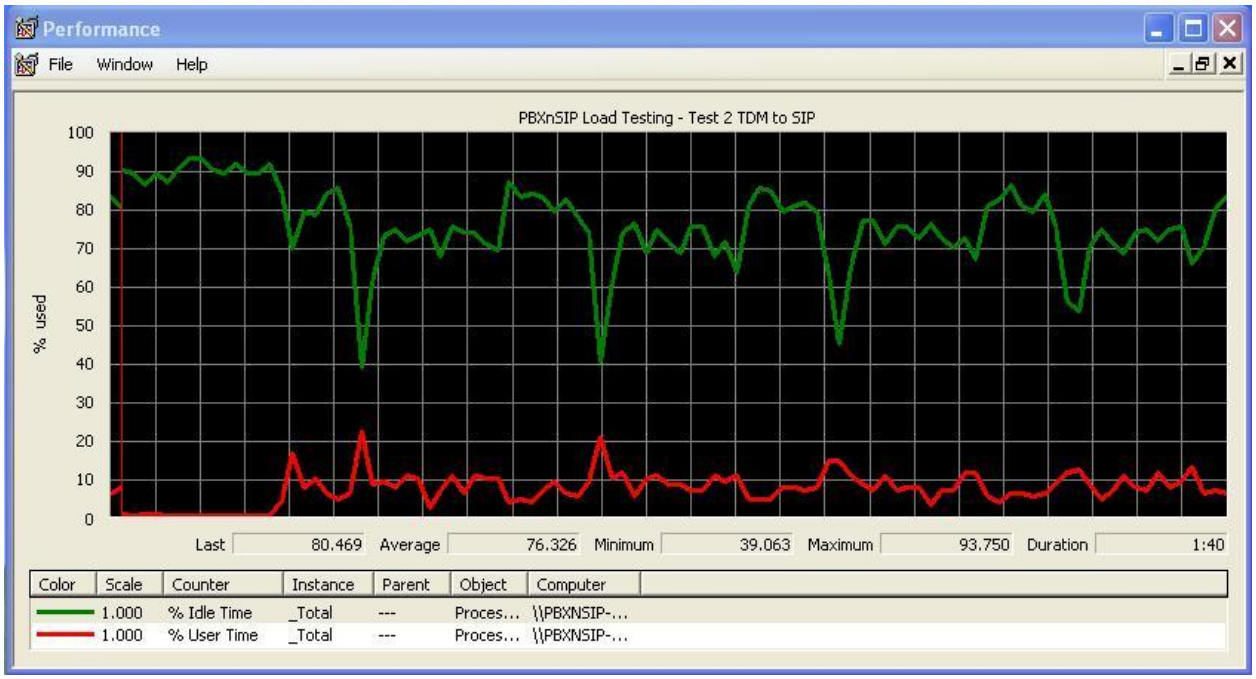

Figure 15 – Test 2 – 12 calls, 8cps, 10 seconds

Since the system was able to handle 8cps from NBE, the calls per second was increased to 12 for the next test. Figure 16 shows the load results of 12 calls at 12 cps with duration of 10 seconds.

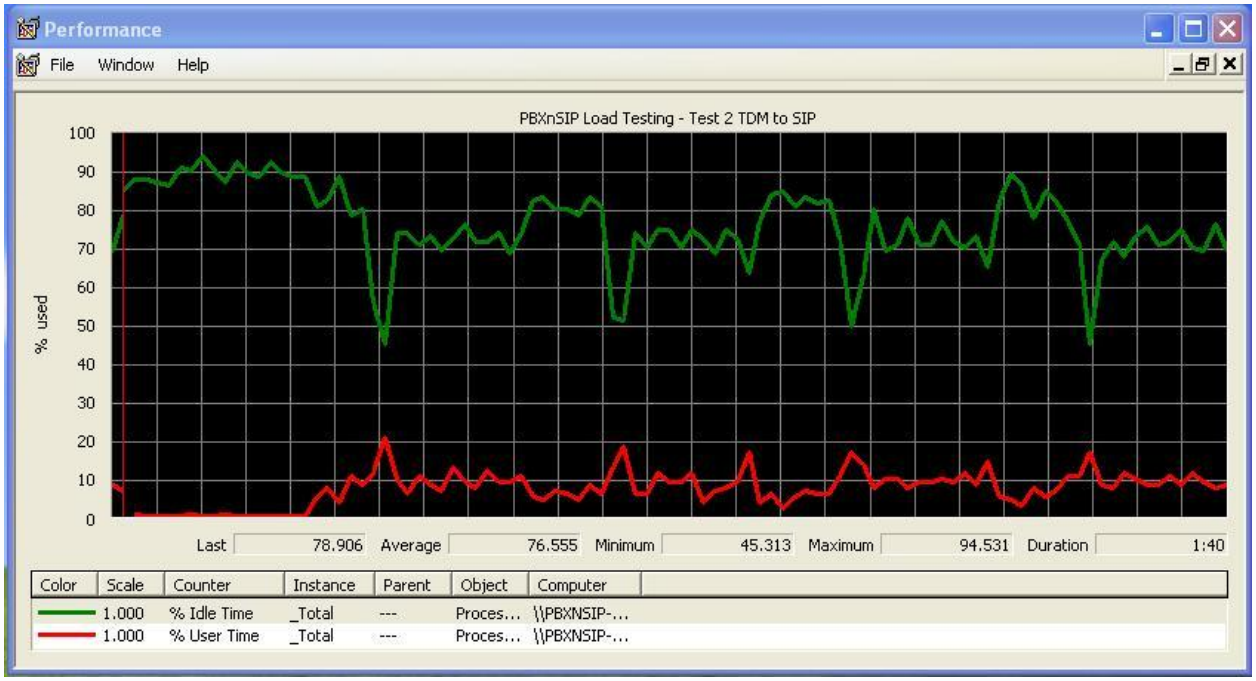

Figure 16 – Test 2 – 12 calls, 12 cps, 10 seconds

Results

The testing has shown that the 2U Rockbochs system can easily handle the load generated by using PBXnSIP, NBE-2.0, and a Sangoma A200 with 12 FXO lines. At the worst case scenario, with all 12 analog lines ringing at once the load of the system did not go below 40% during channel setup and stayed around 70% idle once the call was connected.

Test 2: PBXnSIP + NBE2.0 w/ A20006DE TDM-IVR

Test Description

This test was to see what the system load was like when calls were sent from NBE-2.0 to PBXnSIP's internal IVR. The call diagram for this test is shown in figure 17.

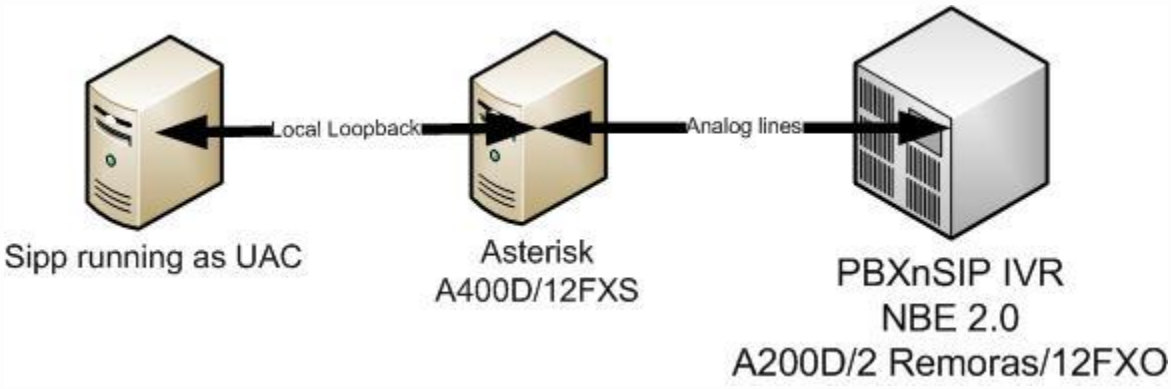

Figure 17 – TDM to IVR Test Scenario

The same custom SIPP scenario was used to generate the calls and send them to an Asterisk system. The Asterisk server then sent the calls out FXS interfaces to the PBXnSIP system running NBE-2.0. PBXnSIP was set answer the calls and playback an IVR greeting.

Test Procedure

To start the test Sipp was set to generate 12 calls at 12 cps with duration of 5 seconds. With these test parameters a strange behavior was observed from the NBE/PBXnSIP system. One of the 12 calls was never answered, causing the test to fail. Figure 18 shows the log files from PBXnSIP.

```
[5] 2009/06/19 09:06:26: Identify trunk (IP address and domain match) 4
[5] 2009/06/19 09:06:26: Trunk NBE Gatway (not global) sends call to account 70 in domain localhost
[5] 2009/06/19 09:06:26: Identify trunk (IP address and domain match) 4
[5] 2009/06/19 09:06:26: Trunk NBE Gatway (not global) sends call to account 70 in domain localhost
[3] 2009/06/19 09:06:26: DoS protection: Not accepting more calls
[5] 2009/06/19 09:06:41: Identify trunk (IP address and domain match) 4
[5] 2009/06/19 09:06:41: Trunk NBE Gatway (not global) sends call to account 70 in domain localhost
```

Figure 18 – Test 3 – PBXnSIP log file

According to PBXnSIP when the system load gets over 75% PBXnSIP will stop accepting calls to defend against a Denial of Service attack. Figure 19 shows the load results from this test run confirming that the load gets to this level (NOTE: the load graphs are generated by Windows Performance Monitor at 1 sec intervals)

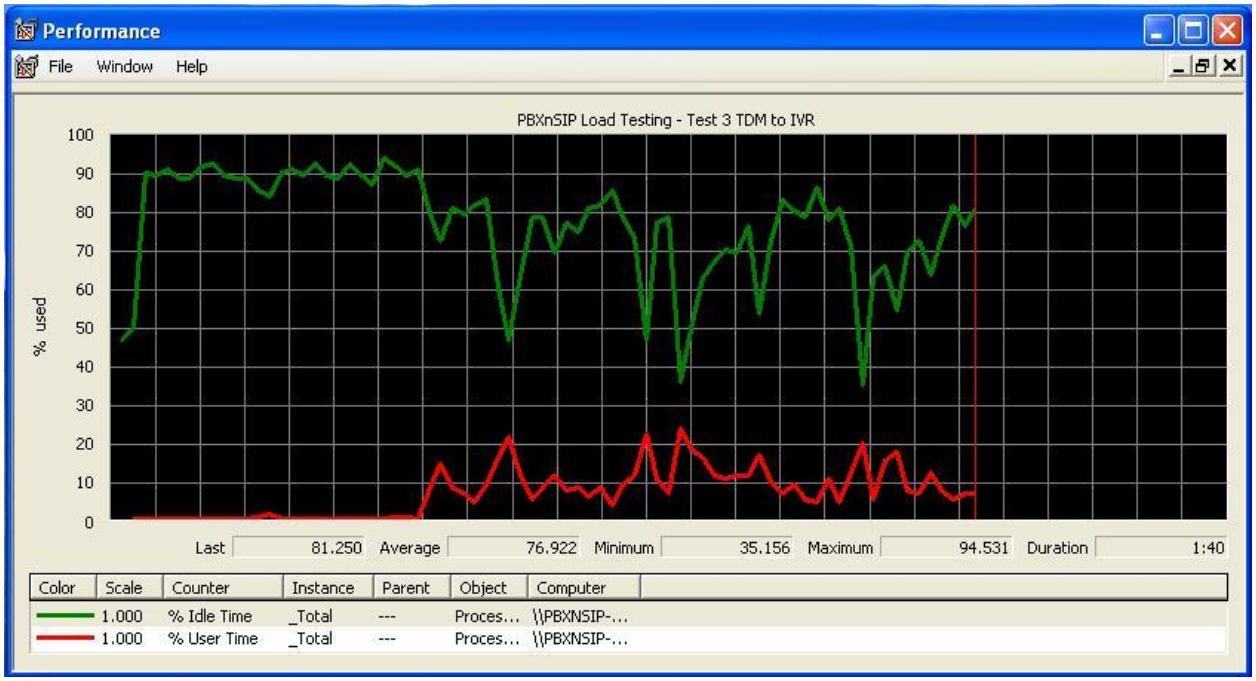

Figure 19 – Test 3 – 12 calls, 12 cps, 5 second

Since 12 cps was loading the system too much the next test was performed with 10 cps. Figure 20 shows the load results from this test run.

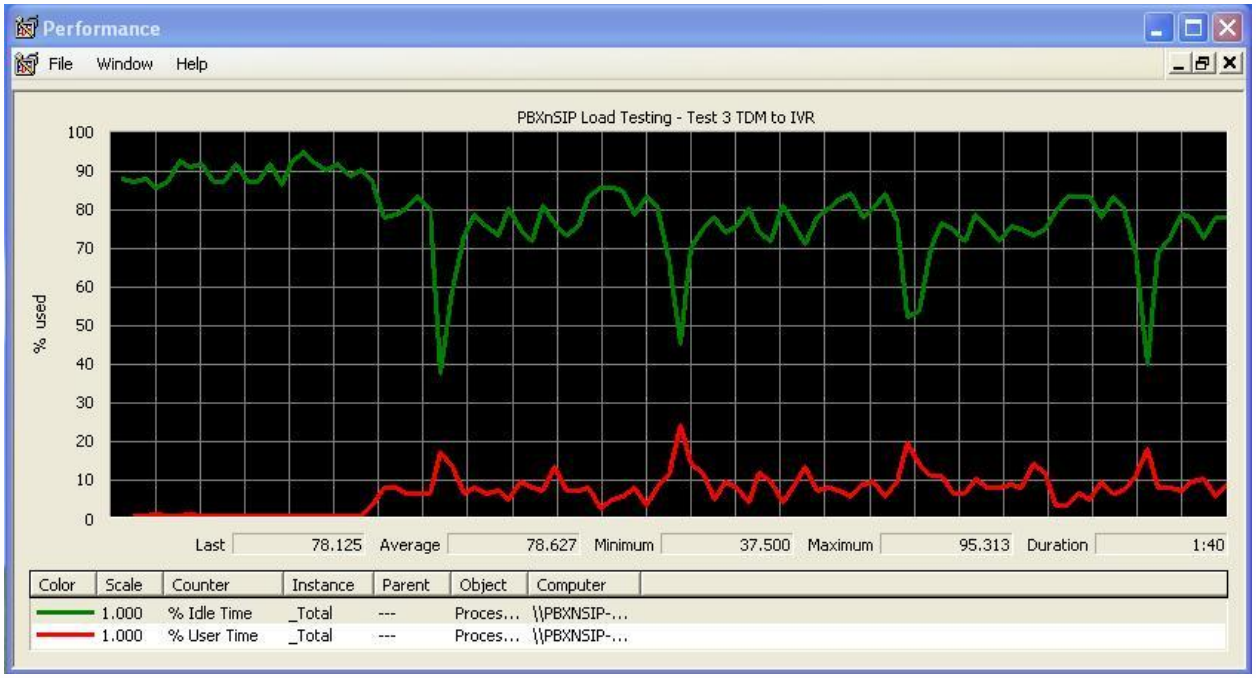

Figure 20 – Test 3 – 12 calls, 10 cps, 10 second

50 McIntosh Drive, Suite 120, Markham, ON L3R 9T3, Canada • t. 1 905 474 1990 • toll free in North America 1 800 388 2475 • f. 1 905 474 9223 • www.sangoma.com

Results

Testing showed that sending the calls to an IVR reduced the number of calls per second that the system could handle as the load dropped below an acceptable level. If the system will send all calls to an IVR system it is therefore recommended that the incoming calls be limited to 10 calls per second.

Testing also showed that IVR playback does not affect the steady state load once the channel was started indicating that channel setup was the main factor in the number of calls the system is able to handle.