Actual Test Report from the Landslide 2500 GGSN Performance Test Plan

November 3, 2004

Spirent Communications
Global Services

1 Executive Summary

This report summarizes the test results and observations notice during the execution of the <Major US Wireless Service Provider> GGSN Performance Test Plan.

Spirent Communications conducted performance analysis and benchmark testing of the Nokia, Cisco/Seimens and Nortel GGSN's system as part of <Major US Wireless Service Provider> RFP evaluation of GPRS Packet Data wireless infrastructure solutions to satisfy their next generation wireless network deployment.

The overall objective of the series of test cases executed as defined in the <Major US Wireless Service Provider> GGSN Performance Test Plan was to validate and determine suitability of the vendor's GGSN's and Content Billing Gateway to perform under extreme operational conditions and capacity loads.

The on-site testing occurred at the vendor's designated labs of Cisco Systems in Raleigh, North Carolina, Nortel Networks in Richardson, Texas and Nokia Networks in Mountain View, California over a six week period from September 13, 2004 through October 22, 2004. Spirent Engineers worked very closely with Test Engineers from the various vendors to ensure proper GGSN device configuration and testing.

Spirent' Landslide 2500 GPRS Performance Test System was used to conduct all testing and benchmarking. The Landslide 2500 GPRS system included one (1) Landslide Manager and four (4) Landslide 2500 Test Server.

US Wireless Customer and Spirent agreed to performed the following tests on each vendor's GGSN and content billing gateway:

1. Capacity Test with Fixed Distribution
2. Capacity Test with Continuous Data
3. Session Loading
4. Session Loading Rate Test
5. Inter-SGSN Handoff Tests
6. Session Loading with Mobility
7. Advanced Data Services (Content Billing)
8. Capacity Test with Multiple Primaries and Secondary Contexts

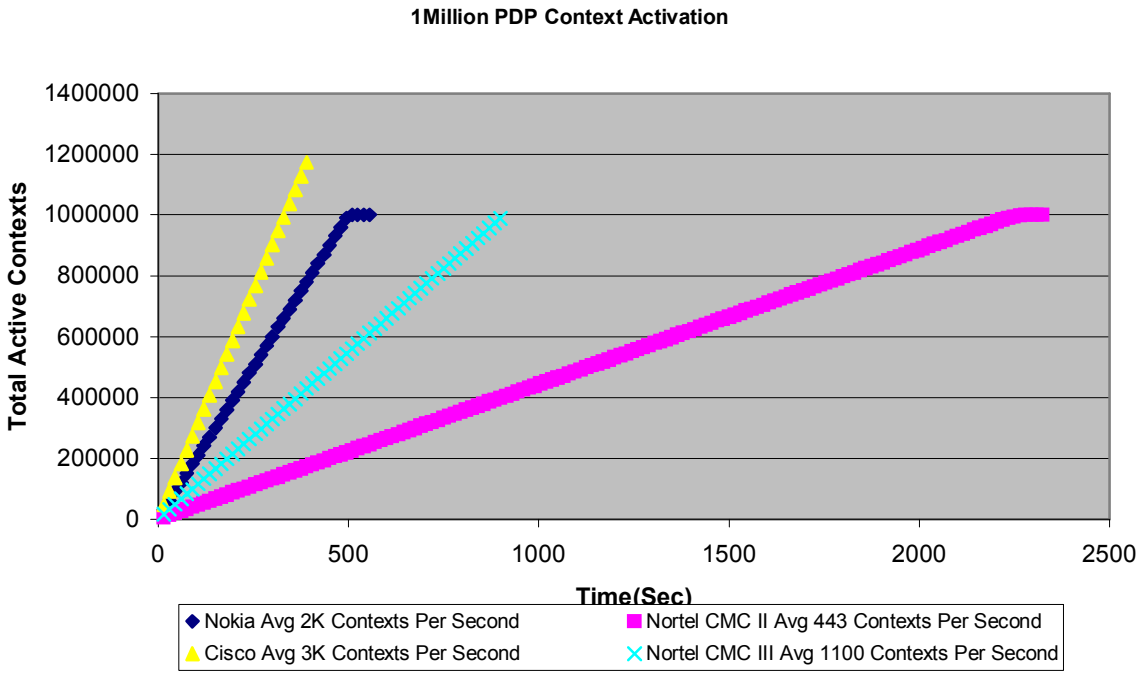
2 Test Operation and Results

2.1 Capacity Test with Fixed Distribution

For all of the capacity tests, the Landslide 2500 GPRS began establishing contexts at a rate specified by the user. This should be a rate less than the maximum context rate supported by the system under test. The Landslide 2500 Test Server continues to establish contexts until the number of context specified by the user has been reached or the user stops the test. Once the capacity of the GGSN has been reached, all subsequent context activation attempts fail. By setting the requested number of contexts higher than the supported capacity of the GGSN, the user can determine the actual capacity.

If all contexts are established successfully, the test server keeps all contexts active for the length of the specified period or until stopped, and then begins deactivating each of the contexts at the specified deactivation rate. A fixed distribution model provides a constant rate, which does not deviate from the user-selected rate. For example, if a user selects a session activation rate of 1000 sessions/second, selecting a fixed distribution model will result in 1000 activations every second.

The original objectives of this test case were to run at 1k activations per second, up to 1 Million active contexts. The actual objectives were adjusted, based upon the Vendor requests and capacities learned during the testing. The final results for the three Vendors are shown in the *1Million PDP Context Activation* chart below:



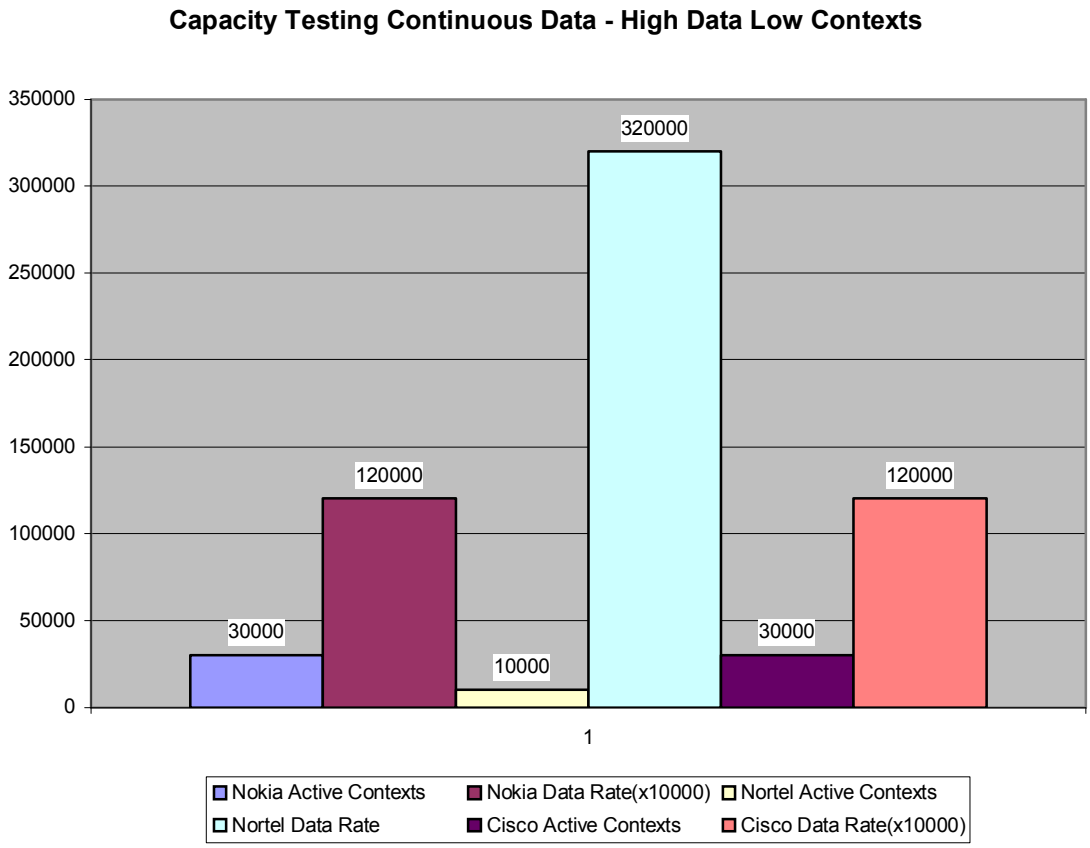
2.2 Capacity Test with Continuous Data

The methodology as described in the *Capacity Test with Fixed Distribution* applies here; the difference is data being generated with the protocol as defined by the user.

2.2.1 High data Low Contexts

The intention of the test case is to generate a large amount of data over a small number of PDP contexts. The original objective is to establish 30k active contexts, running 1 packet per second, at 1300 byte UDP packets.

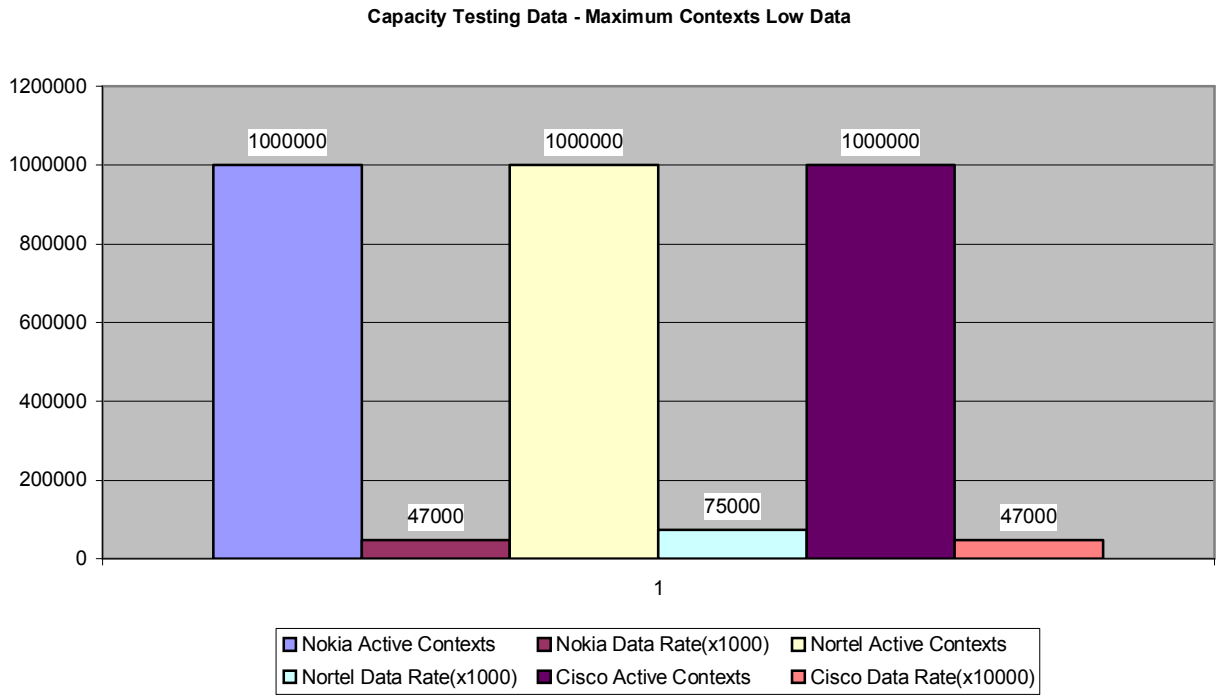
The actual objectives were adjusted, based upon the Vendor requests and capacities learned during the testing. The final results for the three Vendors are shown in the *Capacity Testing Continuous Data – High Data Low Contexts* chart below:



2.2.2 Small data over 1 Million Contexts

The intention of the test case is to generate a small amount of data over the maximum of 1 Million PDP contexts. The original objective is to establish 1 Million active contexts, running 1 packet per second, at 1300 byte UDP packets.

The final results for the three Vendors are shown in the *Capacity Testing Data - Maximum Contents Low Data* chart below:



2.3 Session Loading

A key test of a GGSN's performance is its ability to process real world activity for a long period of time. Session loading allows the user to emulate real world activity so that the GGSN is handling multiple events, such as context activations, and context deactivations at the same time.

During session loading test activity, the test case begins setting up contexts at a selected rate of activations per second. Each context is maintained for the specified session hold time. At that time, the context is deactivated and left idle for the specified session pending or idle time, and is then reactivated. The test continues in this manner until the test is stopped.

The original objectives of this test case are as follows: PDP contexts are brought up 500 per second, with possibly up to 400k total. This test will activate a PDP context; hold for 110 seconds, then at the end of this time the context will be de-activated. It will stay de-activated for 110 seconds, and then become activated (connected) again.

Additional PDP contexts would have been established from the other two test servers as required in testing capacities of each GGSN. The actual objectives were adjusted, based upon the Vendor requests and capacities learned during the testing. The final results for the three Vendors are shown in the *Session Loading (Approximate)* chart below:

2.4 Session Loading Rate Test

The purpose of the Session Loading Rate test is to determine the maximum processing rate of the GGSN. It works similar to the session loading test activity, with the difference being that the Session Loading Rate test continues to increase the rate of context activations until a user defined error threshold is exceeded.

During the Session Loading Rate test activity, the test case begins setting up contexts at a selected rate of activations per second for a user-defined session loading time. The contexts are maintained for the specified session hold time. At that time, the context is deactivated and left idle for the specified session pending or idle time, and is then reactivated.

The test continues in this manner until the selected session loading time expires. At that time, the test case examines the error rate, and determines whether the error threshold has been exceeded. If the error rate is not exceeded, the test decreases the hold and pending times by their respective adjustment values, and runs the test again for the specified session loading time.

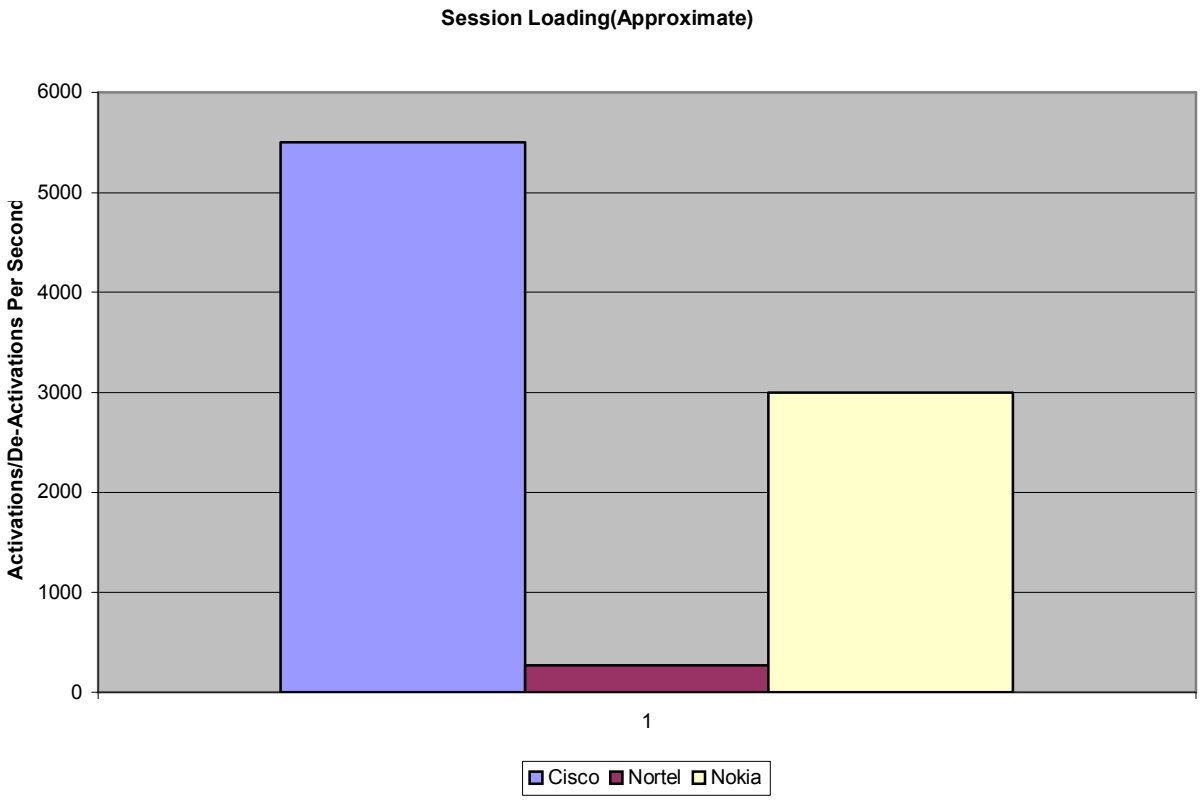
Once the error threshold has been exceeded, the test continues to run at the same rate, until stopped or until the error percentage drops below the error threshold. Should the session loading run time expire while the error percentage is below the error threshold, the session hold and pending times are decreased again.

This test case builds upon the *Session Loading test* done previously. The original objectives of this test case are as follows: PDP contexts are brought up 500 per second, with possibly up to 400k total. At the end of 100 seconds, the error threshold (1%) is checked, and if the error threshold is not met, then the hold/pending times are backed off by 5 seconds.

For the current hold/pend times, after a PDP context is activated, then the context is held in the active state for 110 seconds. At the end of 110 seconds the session is de-activated and held in the pending state for 110 seconds. Backing off every 5 seconds will continue until the 1% error threshold is met or exceeded.

Additional PDP contexts would have been established from the other two test servers as required in testing capacities of each GGSN. The actual objectives were adjusted, based upon the Vendor requests and capacities learned during

the testing. The final results for the three Vendors are shown in the *Session Loading (Approximate)* chart below:



2.5 Session Activation Rate Test

The purpose of the context activation rate test is to determine the maximum activation rate of the GGSN at various capacity levels. It is designed to determine if the SUT's maximum activation rate varies depending on the total number of contexts active.

During the Session Activation Rate test activity, the test case begins setting up contexts at a selected rate of activations per second for a user-defined time period or for a specified block of contexts. At the end of the time period, or once the block of contexts has been activated, the test checks to see if the error threshold has been exceeded. If not, the test initiates another activation run and activates the next block of contexts or activates contexts for a timed run.

If the error threshold is exceeded, the test deactivates the last set of contexts, decreases the activation rate by the specified back off percentage, and repeats

the round of activations at the lower rate. Once that round has been completed at a rate that does not exceed the error threshold, the test continues with the next round of activations

The original objectives of this test case are as follows: PDP contexts are brought up 1000 per second, with a block size of 40,000. At the end of 60 seconds, the error threshold (1%) is checked, if the error threshold is exceeded, the test deactivates the last block of sessions, decrease the activation rate by the 10%, and reactivates that block at the lower rate. Once that block has been activated at a rate that does not exceed the error threshold, the test continues with the next block or context period.

Additional PDP contexts would have been established from the other two test servers as required in testing capacities of each GGSN. The actual objectives were adjusted, based upon the Vendor requests and capacities learned during the testing.

There were software issues using the test equipment, therefore the results from the *Capacity Testing without Data* should be used as the stated tested capacities from each Vendor.

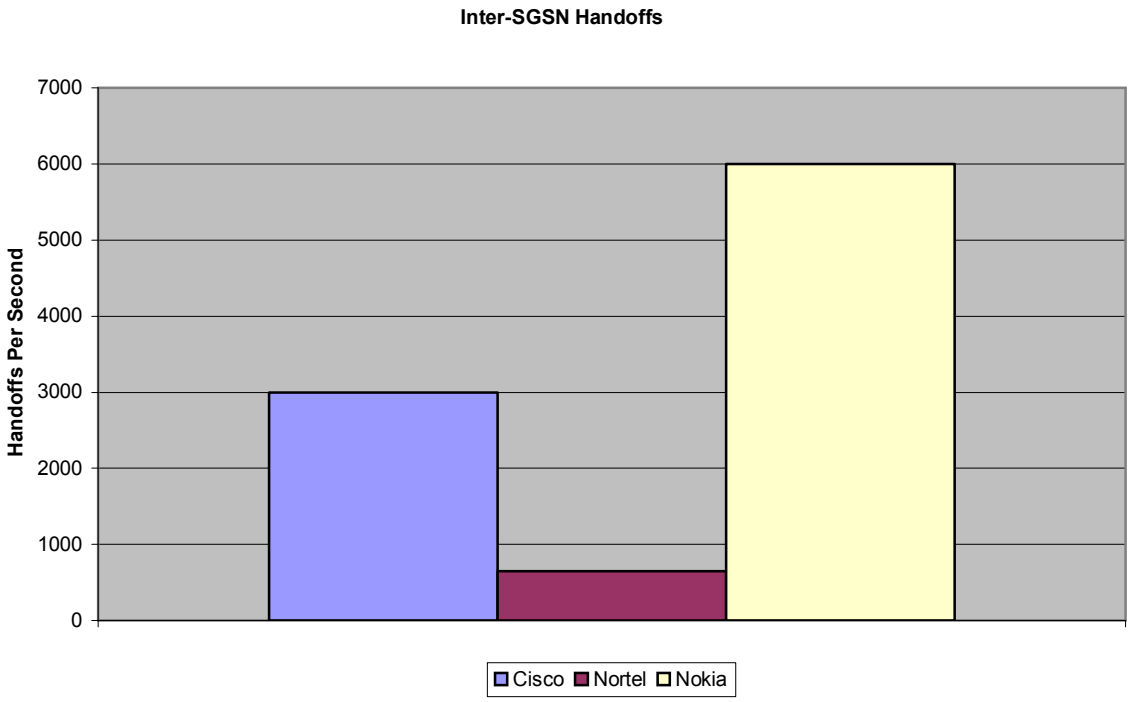
2.6 Inter-SGSN Handoff Test

The purpose of the Inter-SGSN handoff test activity is to determine the GGSN's ability to process handoffs. By running an Inter-SGSN handoff test at various capacity levels and various handoff rates, the user can determine the maximum handoff rate supported at various capacity levels.

During the Inter-SGSN handoff test activity, the test case begins setting up contexts at a selected rate of activations per second until the total number of activations is achieved.

This test case builds upon the *Capacity test with Fixed Distribution*. The original objectives of this test case are as follows: SGSN Handoffs are to occur after all PDP contexts have been established. PDP contexts are brought up at 1000 per second, with possibly up to 1,000,000 total. After a 15 second delay, handoffs will start occurring to the other SGSN (once) at the rate of 1000 per second.

The actual objectives were adjusted, based upon the Vendor requests and capacities learned during the testing. We were unable to determine the Cisco and Nokia maximum due to lack of test equipment and license. The final results as available for the three Vendors are shown in the *Inter-SGSN Handoffs* chart below:



2.7 Session Loading with Mobility

During session loading with mobility test activity, the test case begins setting up contexts at a selected rate of activations per second. Each context is maintained for the specified session hold time. After context activation, the test waits the specified handoff time, and then performs the specified number of handoffs. At the end of the specified session hold time, the context is deactivated and left idle for the specified session pending or idle time, and is then reactivated. The test continues in this manner until the test is stopped.

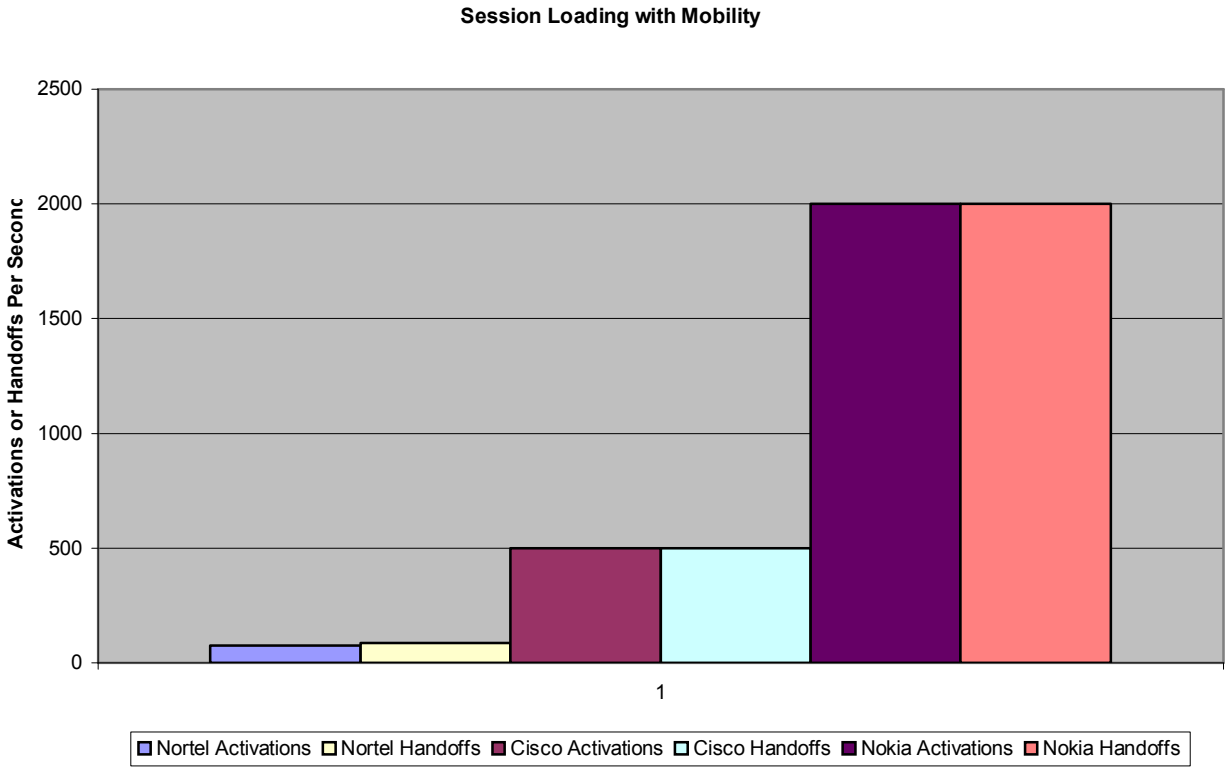
This results in context activations and deactivations, and handoffs occurring in a real world model. The user may also select data traffic during the session hold time, to further reflect real world conditions.

This test case is derived from the *Session Loading Test*. The original objectives of this test case are as follows: PDP contexts are activated at a rate of 500 per second, and then they are held active for 110 seconds. After 60 seconds, they are handed off to the other SGSN at 500 per second. Another 60-second delay is invoked, and then they are handed off back to the original SGSN. The PDP contexts then remain in this state for 110 seconds. At the end of 110 seconds, they are disconnected at a rate of 500 per second. They remained de-activated

(disconnected) for 110 more seconds. At the end of 110 seconds, they are activated again and the entire process repeats itself.

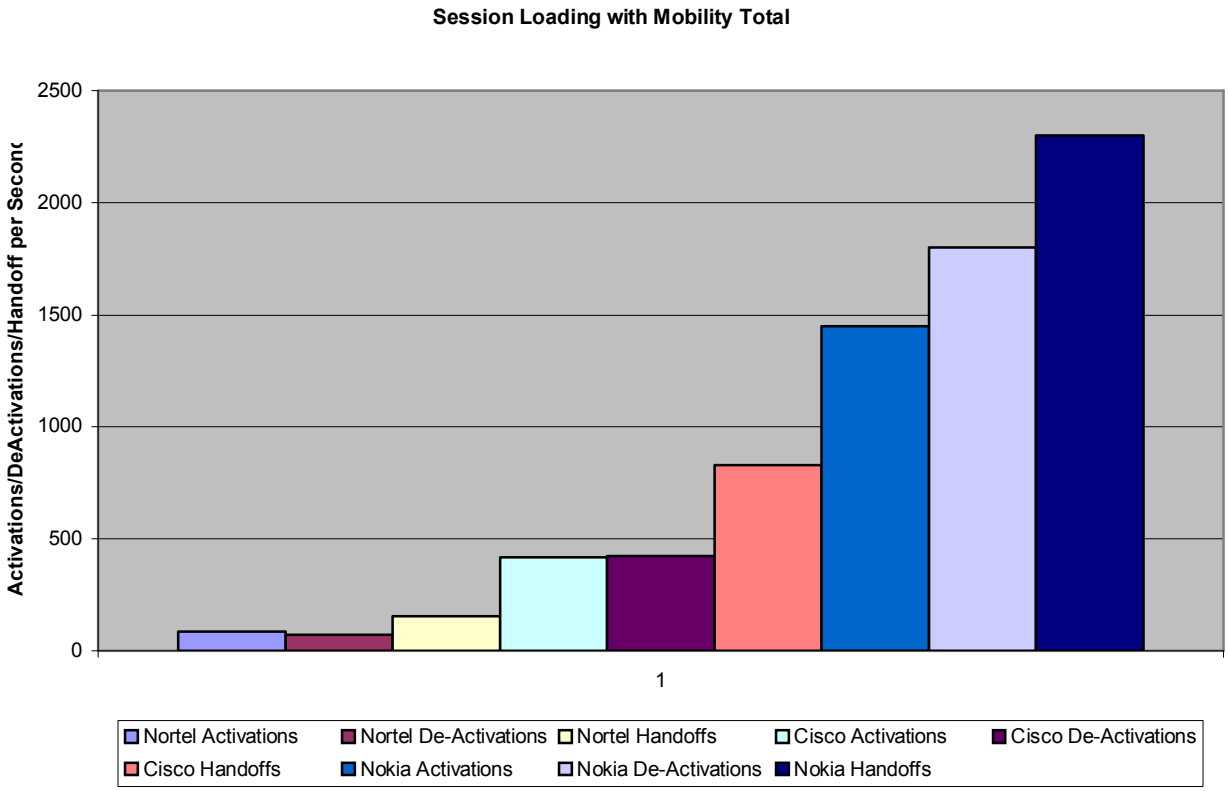
2.7.1 Recorded Activations and Handoffs per Second

Additional PDP contexts would have been established from the other two test servers as required in testing capacities of each GGSN. The actual objectives were adjusted, based upon the Vendor requests and capacities learned during the testing. The results for the Cisco GSSN for this test were limited due to time constraints because of an observed handoff failure. Time was allotted trying to determine this issue and as a result this objective was not completely reached. This should in no way reflect on Cisco's (or Nokia's) ability to go towards much higher capacities, which it Spirent believe is possible. The final results for the three Vendors is shown in the *Session Loading with Mobility* chart below:



2.7.2 Recorded Activations/De-Activations and Handoffs per Second

Additional PDP contexts would have been established from the other two test servers as required in testing capacities of each GGSN. The actual objectives were adjusted, based upon the Vendor requests and capacities learned during the testing. The maximum rates for this test case were not met. This should in no way reflect on Cisco or Nokia's ability to go towards much higher capacities, which Spirent believes is achievable. The final results for the three Vendors is shown in the *Session Loading with Mobility Total* chart below:



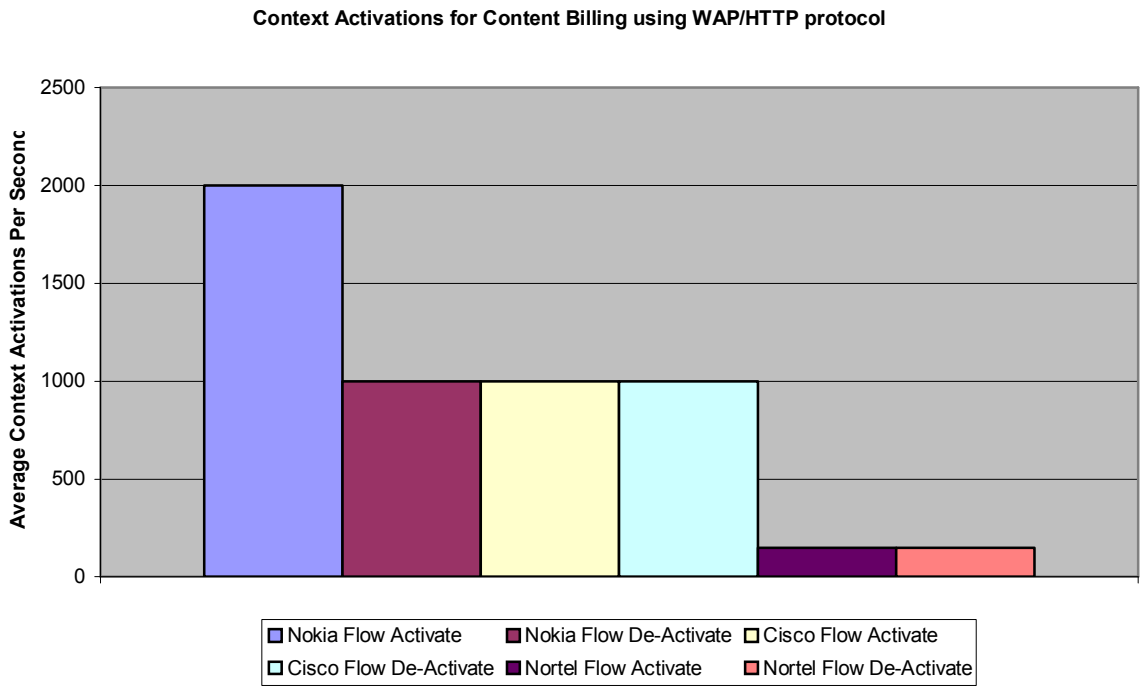
2.8 Advanced Data Services (Content Billing)

Content Billing test is to verify the ability of the GGSN and the Content Billing Server to process billing information based on real-time traffic inspection at the application layer (Layer 7):

2.8.1 Activations/Flows per Second over Total Contexts

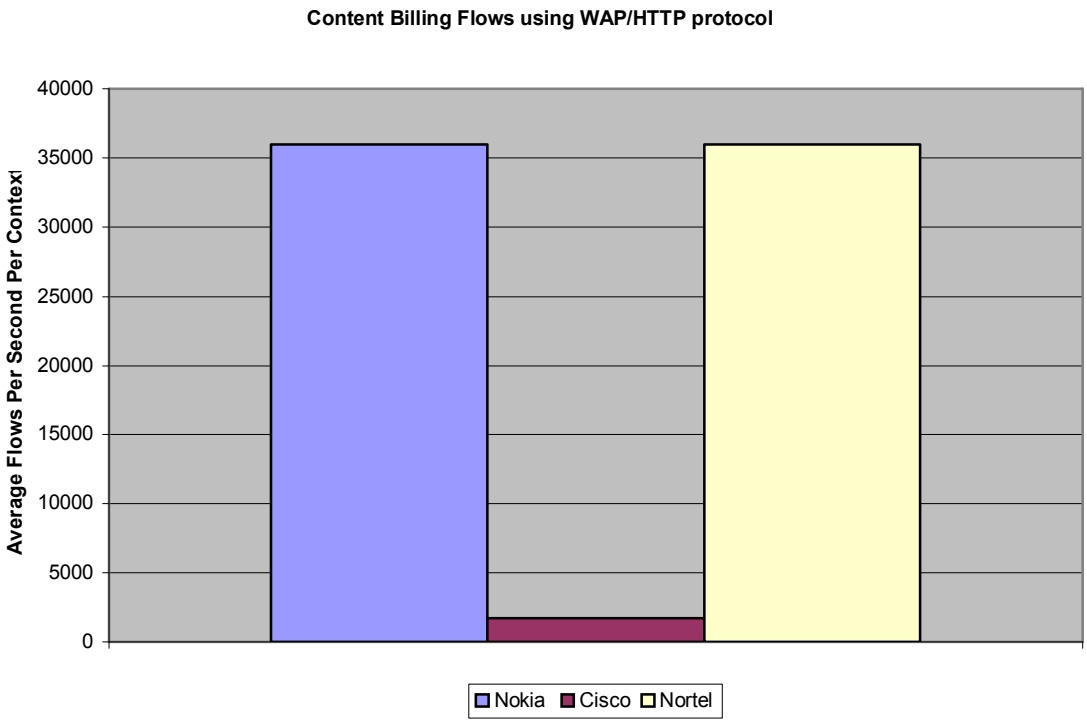
The original objectives of this test case followed *Capacity Testing with Fixed Distribution* and are as follows: The number of active PDP contexts should equal 1,000,000 per GGSN, running at 1k activations per second, with a final total of 1k WAP/HTTP flows per second. The actual objectives were adjusted, based upon the Vendor requests and capacities learned during the testing. The

final results for the three Vendors are shown in the *Context Activations for Content Billing using WAP/HTTP protocol* chart below:



2.8.2 Maximum Flows per Second over limited Contexts

The original objectives of this test are as follows: The number of active PDP contexts should equal 36,000 per GGSN, running at 1k activations per second. This was the maximum amount of data that could be generated with the number of test servers available. The actual objectives were adjusted, based upon the Vendor requests and capacities learned during the testing. The final results for the three Vendors are shown in the *Content Billing Flows using WAP/HTTP protocol* chart below:



2.9 Capacity Test with Multiple Primary & Secondary Contexts

For all of the capacity tests, the Landslide GPRS begins establishing contexts at a rate specified by the user. This should be a rate less than the maximum context rate supported by the system under test. The TS continues to establish contexts until the number of context specified by the user has been reached or the user stops the test. Once the capacity of the GGSN has been reached, all subsequent context activation attempts fail. By setting the requested number of contexts higher than the supported capacity of the GGSN, the user can determine the actual capacity. If all contexts are established successfully, the test server keeps all contexts active for the length of the specified duration or until stopped, and then begins deactivating each of the contexts at the specified deactivation rate.

Perform a capacity test with 2 primary contexts selected for each mobile station and four (4) secondary contexts selected for each primary context.

Due to time limitations, Nokia was the only Vendor we were able to officially record any data for this test. This particular test worked as described in the *Capacity Testing without Data* results.

Result:

3 Summary

This report summarizes the results of the Spirent Global Services engagement with <Major US Wireless Service Provider> during the 6-week period of September 13 to October 22, 2004. During the engagement, Spirent and Scientific Software Engineering (SSE) engineers worked in conjunction with Cisco/Seimens, Nokia, and Nortel engineers to conduct the performance and benchmark testing and configuration needed to achieve the goals set forth in the <Major US Wireless Service Provider> GGSN Performance Test Plan.

Spirent Communications and SSE appreciate the opportunity to work with <Major US Wireless Service Provider> on this project, and look forward to future opportunities to enable <Major US Wireless Service Provider> to deploy next generation network technologies and services with added assurances and reduced risk to network performance.

4