



Testing Session Initiation Protocol (SIP) Saturation

Effects, caveats, and issues encountered in testing SIP servers

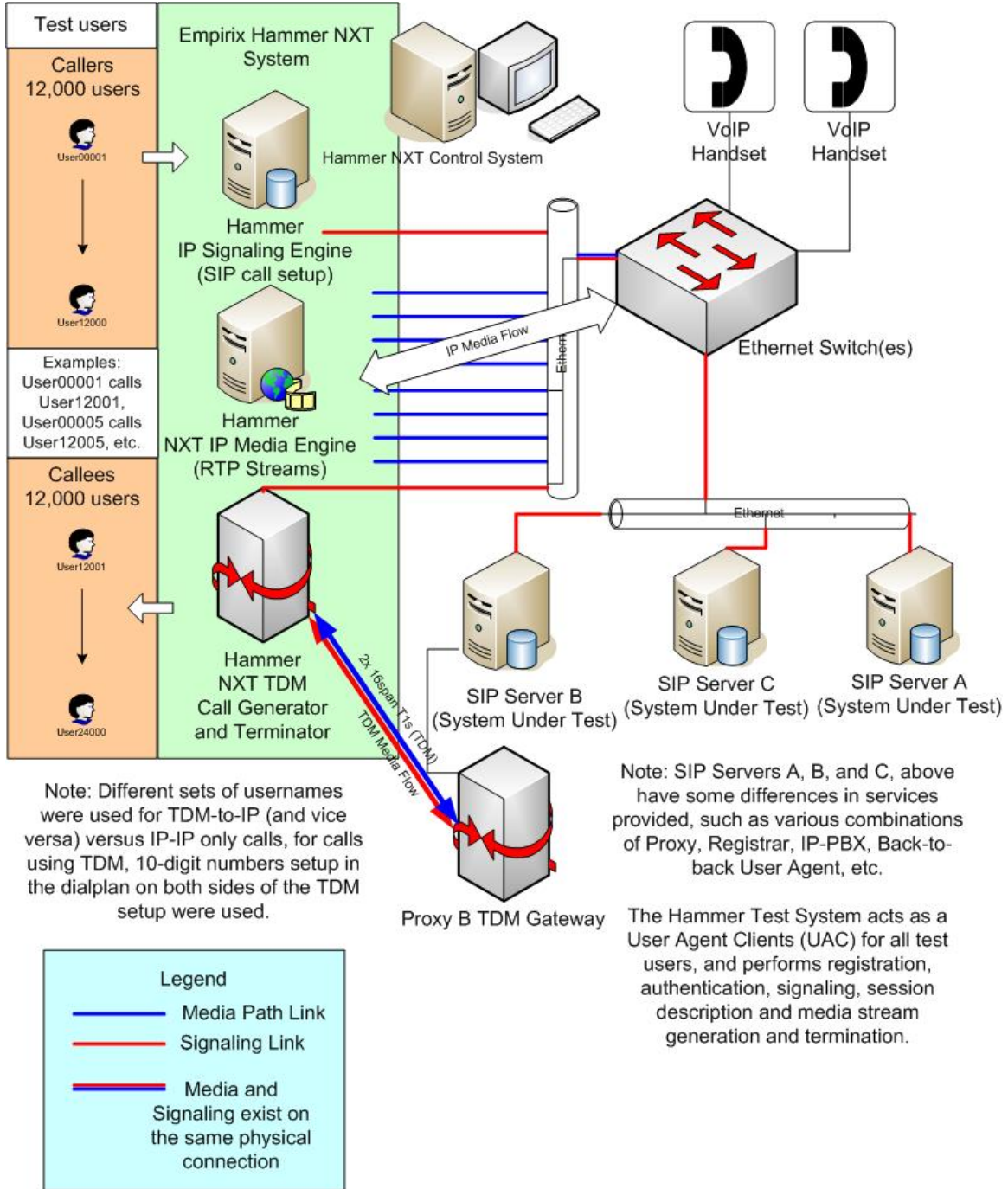
Introduction

The SIP (session initiation protocol) signaling protocol is emerging as the industry's preferred Voice over IP (VoIP) technology. As such, SIP's scalability is key to VoIP technology's continued adoption, and the need to test SIP user agents and servers under heavily saturated scenarios is becoming more acute. SIP servers must be able to bear heavy traffic loads for user registration and call setup functions that require reading from and writing to one or more large dynamic database(s). The corresponding test tools, call generators, terminators and analyzers must all be capable of processing the same significant loads. The University of New Hampshire InterOperability Laboratory (UNH-IOL) can test a large variety of VoIP and traditional telephony systems; however the focus of this paper is not to demonstrate the comparative performance of individual SIP servers, but rather to delineate issues involved in testing SIP servers under heavy call saturation.

For the purposes of this research the UNH-IOL chose to test against three different proxy servers, designated in this white paper as Proxy A, Proxy B and Proxy C. They ranged from open source to enterprise and Java-based implementations, respectively. The representative test equipment used for this set of investigations, provided by Empirix, included hardware and software from the Hammer NXT line of products.

The target call load for IP-to-IP based calls was 24,000 sustained SIP calls, as this was the capability of the test hardware in its current configuration. The TDM gateway functionality of Proxy B handled calling to or from TDM circuits. The channel count (number of calls) was directly tied to the number of TDM circuits available on the system under test (SUT). The testing strategy called for steadily ramping 12,000 callers to 12,000 callee user agents for a sustained call volume of 24,000 calls. The call rate (the slope of the ramp function) varied for different test runs, depending on how each of the SUTs could handle the various rates. Once the SIP servers were sufficiently saturated, the hardware and software of the SIP communication components displayed divergent behavior.

Overview



All of the tests detailed in this report employed the above network topology. The Hammer NXT IP system had a Hammer NXT Controller for controlling the Hammer NXT IP Signaling Engine and the Hammer NXT IP Media Engine. The Hammer NXT IP Signaling Engine generated the SIP call messages and the Hammer NXT IP Media

Engine generated RTP audio streams. The Hammer NXT TDM system generated the ISDN call messages and TDM media that was used in conjunction with the Hammer NXT IP system.

A single high-performance Ethernet switch/bridge, capable of operating at 10/100/1000 Mb/s connected the Hammer NXT system to the various SIP servers under test. Both the Hammer NXT system and the SIP servers were using the same logical IP subnet. Although real world scenarios would almost certainly involve multiple IPV4 route hops, a single test subnet was used to reduce testing variables. Future testing of SIP should include multi-hop and WAN/LAN scenarios including firewalls and other transport networks and networking devices.

The Hammer Media Server employs a unique method of generating audio streams and verifying the quality of those streams. It uses eight physical Fast Ethernet ports, each defined with an individual IP address and physical interface. This configuration results in the separation of (by IP Address) the SIP signaling and RTP audio traffic. The RTP stream's data payload was pre-encoded in the test's codec. When the terminating port of the Media Server receives the data, the Media Server executes a digital analysis of the RTP data. This allows it to establish 24,000 RTP streams and verify the integrity of media path and media content without using traditional DSP.

The testing methodology employed the standard three-way handshake of SIP. In this operation, before calling can commence, the Hammer Signaling Server must issue a REGISTER request for each of the 24,000 SIP endpoints. The SUT is expected to return a "200 OK" message for each. After the registration process is complete, the calling phase can begin. The SUT receives an INVITE request with an SDP payload that describes the desired RTP stream variables for each call. The SUT is expected to answer each call with a status message of "200 OK." The Signaling Server then establishes an RTP audio stream for each call and verifies the data sent across the switch. A side effect of separating the signaling traffic and the RTP traffic is that the SDP payload describes a media IP address that differs from the signaling IP address. Also since more than one RTP stream was sent per IP address, many ports must be used to provide source and destination UDP ports for all streams.

All tests consisted of a steady (fixed CPS) call ramp up function attempting to reach a sustained call volume of 24,000 calls. The Calls Per Second (CPS) rate offered by the Hammer NXT IP Signaling Server is dependent upon the particular proxy server tested. The offered CPS was anywhere from 5 to 50, depending on the proxy server and its observed performance. The rate at which the Signaling Server is configured to call determines the individual call length. This value is between 300 and 2,400 seconds. After a sustained volume of 24,000 calls, the Hammer NXT IP Signaling Server will terminate each call when it reaches the desired call length. The channels are then restarted (i.e. the call script is restarted) to ensure maximum sustained throughput. This process can run forever, and the concept is to maintain the maximum call volume for an extremely long amount of time to prove the stability of the SUT under load. Total test time could be as

short as 10 minutes, or indefinite. This procedure saturates the proxy server, simulating the real world phenomena of high call volume.

The series of tests used a method to validate that the media path was open, clear and useable via a script called ConfirmPath on the Hammer System. ConfirmPath is a simple but effective test. After establishing a SIP call from Side A of the Hammer NXT to the appropriate callee on Side B of the same device, an RTP stream is connected between the two end points. A series of tones is generated and sent through the streams, in both directions. Side A and B both send and receive tones in order to verify bidirectional audio capability. The tones were encoded with the G.723 algorithm at 5.3Kb/s. G.723 is an ITU specified codec that provides a balance between voice quality and low resource requirements.

The test's large amount of SIP calls and RTP data streams placed severe requirements on the network infrastructure and thus caused complications. For example, registering all 24,000 SIP users to a proxy server created an average of 207 megabytes of traffic in the call set-up phase. Actually setting up the calls created around 278 megabytes in SIP signaling alone (independent of registration size). The RTP audio streams averaged 2.8 gigabytes of data per test. This placed significant load on the IP stacks of the clients and the network infrastructure. Due to the high traffic volumes, an older model switch had to be replaced with a newer high-capacity switch in order for all of the SIP messages to pass properly between components.

Proxy A (SIP IP-to-IP only)

The UNH-IOL chose the open source Proxy A because of its reputation as a standards-compliant and high-performing SIP proxy. Using an easily debugged open source implementation made it easier to see the effects of call saturation. This proved to be useful during the course of the tests with this SUT.

Proxy A ran on a Sun Microsystems Ultra 60 machine with 512 megabytes of RAM running Solaris 9. Proxy A uses multiple threads with shared memory. Keeping up with the call volume demanded an increase in the allocation of shared memory for each process from 32 megabytes to 80 megabytes and an increase in the number of threads to 125. This provided sufficient memory space to handle 24,000 calls.

Registration proved the most problematic test process. Initially, Proxy A failed to respond to all registration requests. This failure was due, in part, to the presence of an older model switch that failed to pass all packets appropriately. After replacing it with a newer, more powerful switch, the registration issue was mostly resolved. Increasing the stagger time between SIP registration messages ensured that Proxy A could register all of the users.

Overall, Proxy A scaled relatively well and successfully ramped up to 24,000 calls without issue. However, Proxy A was unable to maintain the call volume without issues. Maintaining 24,000 sustained calls with a CPS rate of higher than 50 consistently resulted

in roughly 10 percent of the calls failing. Changing the CPS to 10 resulted in 100 percent call completion even at the sustained volume of 24,000 calls.

The following are typical test results. Each of these tests was repeated many times to ensure consistent results.

Proxy A failed to complete all of the calls at 50 CPS because the proxy software was unable to keep up with the influx of SIP INVITE requests. Although all of threads were in use, the aggregate CPU utilization did not increase above 20 percent during the tests. This implies the failures were not caused by the hardware or system limitations, of either the network switch or the machine running Proxy A, but the proxy software itself.

Test Number	CPS	Call Attempts	Call Completions	Failures	Reason for Failures
1	50	22456	20872	1584	408 Timeout
2	20	17180	17180	0	-
3	10	18244	18244	0	-

Proxy B (SIP IP-to-IP only)

Proxy B, the second server tested, was a media gateway as well as a pure IP SIP proxy server. In order to test real world scenarios, the UNH-IOL placed calls using ISDN from the Hammer NXT TDM system to the “SIP-aware” Hammer NXT IP system.

In this test, the SIP registration process presented the only scalability issue. Proxy B was configured with a Sun Microsystems SunFire V120 with 512 megabytes of RAM running Solaris 8. It managed registered SIP users through a MySQL database engine. Although the proxy server itself was able to keep up with the demands of heavy SIP registration, the added load of MySQL queries severely stalled registration. Solving this meant statically adding 24,000 test users to the MySQL database. This resolved the registration issue and enabled SIP signaling tests to continue. Not having to wait for all the registrations to complete speeded up testing considerably.

Proxy B was the only proxy of the three tested that was able to consistently scale up to 24,000 calls and maintain the call volume at rates of 50, 20 and 10 CPS. Unlike Proxy A, Proxy B was also able to maintain 100 percent connectivity after call saturation had been reached. This reflects the ability of Proxy B to maintain the internal software state necessary to consistently deal with repeated setup and teardown of a call for a particular SIP user.

Below are the results of the pure IP testing of Proxy B.

Test Number	CPS	Call Attempts	Call Completions	Failures	Reason for Failures
1	50	28733	28733	0	-
2	20	15820	15820	0	-
3	10	17618	17618	0	-

Proxy B (ISDN to SIP/IP)

The media gateway functionality of Proxy B was used to place 736 (two physical cards containing 16 T1s each) calls from the Hammer NXT TDM to the Hammer NXT IP. The Hammer NXT TDM placed the outbound calls over ISDN to the Proxy B media gateway which transcoded the data to the G.711 muLaw codec. For each ISDN call, a SIP INVITE request was sent from Proxy B to the Hammer NXT IP to establish the phone call.

The transcoded audio data sent from Proxy B is received by the Hammer NXT IP Media Server via RTP packets and were analyzed for data integrity. Rather than using G.723, as in previous tests, G.711 muLaw was used to facilitate ProxyB's transcoding capabilities.

The test of the media gateway functionality once again employed the ConfirmPath test script used for Proxy A. Using this test method, two types of tests were executed. One script sent tones in both directions between Hammer NXT TDM and Hammer NXT IP, and the other utilized pre-canned voice files. The Hammer NXT systems analyzed the received data streams ensuring the integrity of the data as it passed from TDM, into the media, and finally into the RTP packets.

Due to an issue with Proxy B regarding ISDN session setup and teardown rates, only CPS rates of 20 and 5 were tested.

Errors were observed in the TDM to IP tests. The errors were caused during the ISDN connection phase with a failure of 0x12 being reported. 0x12 is the ISDN error code returned when the terminating equipment fails to answer, which in this context appears to be the Proxy B TDM hardware failing to respond. A slight statistical improvement occurred when the CPS rate was reduced to five.

The actual reason for these errors is unknown and further testing is required to discover the exact cause. The errors could be the result of cabling issues, SUT configuration, testing equipment configuration or interoperability errors between Proxy B and the testing equipment. These kinds of problems are of direct interest to UNH-IOL as they offer an opportunity to discover the root cause of the issue through continued detailed testing.

No difference in performance was observed between voice and tone RTP tests. This is not surprising, as the volume of RTP traffic is more problematic than the type of data sent via RTP.

The results of the TDM to IP tests are below.

Test Number	CPS	Call Attempts	Call Completions	Failures	Reason for Failures
1	20	3231	2940	291	0x12 ISDN error
2	5	851	775	76	0x12 ISDN error

Proxy C

Proxy C was a commercial SIP proxy server with free educational use licenses. It featured a rich and flexible configuration interface allowing easy SIP configuration. This third proxy offered an alternative to the open source proxy and enterprise class software and hardware.

Proxy C was written using Java and was installed on an AMD Athlon at 1.4 gigahertz with 240 megabytes of RAM running Windows XP, a vanilla install without any modification of settings. Initially, Proxy C failed to respond to all SIP register requests. During the registration phase, the Java process responsible for fielding SIP registration requests used 100 percent of CPU resources. This proxy's failure to register all SIP users recalled the similar issued encountered with Proxy B. Both Proxy C and Proxy B needed to query a dynamic database to store the state of registered users. This provided a tradeoff with respect to flexibility and performance. By increasing registration stagger time to 1 second, Proxy C successfully updated the registration database.

Proxy C proved the least scalable of the three test proxies when subjected to the heavy call load of the Hammer NXT. During all tests, CPU utilization was maxed at 100 percent. As a result, a multitude of errors appeared at all CPS rates.

Of primary importance in this group of tests is the fact that Proxy C was written in Java and was probably never designed to be a high-performing, highly scalable SIP proxy server. The majority of SIP INVITE requests sent to the Proxy C proxy server were never answered with either a status message of "180 Trying" or a message of "200 OK." It is likely that the Java SIP process dropped most of the incoming packets in order to answer a few of the INVITES.

Proxy C was unable to sustain the saturation volume of 24,000 in any of the test scenarios. Proxy C's reached its maximum sustained call volume at 3,234.

Below are three typical test results at all CPS rates.

Test Number	CPS	Call Attempts	Call Completions	Failures	Reason for Failures
1	50	11901	1179	10722	408 Timeout
2	20	12637	1612	11025	408 Timeout
3	10	12720	1389	11331	408 Timeout



Summary

As expected, saturating SIP communications components exposes problems related to scalability and SIP testing in general. The tests described in this paper uncovered divergent behaviors for each of the proxies tested, related in each case to larger numbers of calls per second. This confirms that scalability is an issue for many SIP products and will need to be addressed in light of wider reliance on the protocol. The testing process also revealed issues related to testing SIP in high-volume environments.

Proxy A handled the call volume with relative ease, but showed problems maintaining those connections over time. Although the system Proxy A was running on did not display any significant signs of saturation such as running out of memory or reaching 100 percent CPU utilization, the software itself had trouble keeping up with the test equipment.

Using the Hammer NXT readily revealed the performance ceiling of Proxy A, and in fact all the SUTs. This test solution allowed for greater scalability in terms of call volume than previously available. It also provided a single management interface for both the TDM and IP test scenarios. This permitted full media capability testing of the saturation effects of each SUT without turning to prohibitively expensive traditional digital signal processor solutions.

The enterprise class Proxy B scaled extremely well. It was able to handle the load of 24,000 calls both as the call volume was increasing and while it was being maintained.

The Proxy C server experienced problems ramping up to 24,000 calls at all of the tested call rates. This was not unexpected considering that Proxy C is not designed to process a mass amount of incoming SIP calls.

The various issues encountered during these sets of high-volume call tests suggest some of the key caveats and considerations, especially as they relate to configuration and time, involved in testing SIP systems. As the industry becomes more dependent on SIP, additional testing will be needed to ensure adequately robust and interoperable SIP communications networks.

For more information regarding SIP saturation testing, please contact Gerard Goubert or James Unger at the UNH-IOL.