ILLINOIS INSTITUTE OF TECHNOLOGY

Benchmarking the Session Initiation Protocol (SIP)

Yueqing ZHANG, Illinois Institute of Technology, SAT
yzhan230@hawk.iit.edu

Arthur CLOUET, Illinois Institute of Technology, SAT
aclouet@hawk.iit.edu

Oluseyi S. AWOTAYO,Illinois Institute of Technology, SAT,
oawotayo@hawk.iit.edu

Carol Davids, Illinois Institute of Technology, SAT
davids@iit.edu

Vijay Gurbani, Bell Laboratories, Alcatel-Lucent
vkg@bell-labs.com

IEEE CQR 2015

Problem Statement

Performance can be defined in many ways.

- Generally the term is used to describe the rate of consumption of system resources.
- Time or processing latency is often the metric used to quantify the resource consumption.
- The probability that an operation or transaction will succeed is another metric that has been proposed.
- The metrics are generally collected while the Device Under Test (DUT) is processing a well-defined offered load. The characteristics of the offered load— whether or not media is being used, the specific transport used for testing, codecs used, etc. — are an important part of the metric definition.

Problem Statement

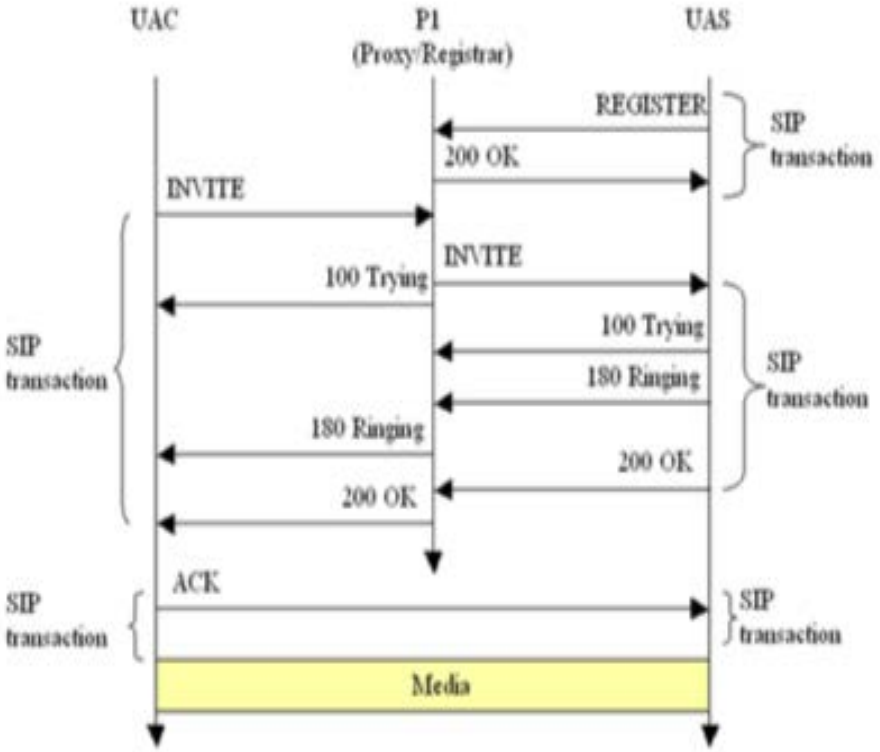
In this study we use a different type of metric and a different type of load.

- Loads are chosen whose individual calls have a constant arrival rate and a constant duration.
- The value assigned to a given load is the number of calls per second, also referred to as session attempt rate .
- The loads are not necessarily designed to emulate any naturally occurring offered load, rather they are designed to be easily reproduced and capable of exercising the DUT to the point of failure.
- We ‘test to failure’, looking for the ‘performance knee’ or ‘break point’ of the DUT while applying a well calibrated pre-defined load.
- The measure of performance is the value of the highest offered load that when applied to the DUT produces zero failures, the next highest rate attempted having produced at least one failure.

Related work

- IETF RFC 7501: Terminology for Benchmarking Session Initiation Protocol (SIP) Devices: Basic Session Setup and Registration
- IETF RFC 7502: Methodology for Benchmarking Session Initiation Protocol (SIP) Devices: Basic Session Setup and Registration
- H. G. Schulzrinne, S. Narayanan, J. Lennox, and M. Doyle, “SIPstone:Benchmarking SIP server performance,” Columbia University, Tech., Rep. CUCS-005-02, 2002.
- Standard Performance Evaluation Corporation. (2014, December) SPEC SIP_Infrastructure. [Online]. Available: http://www.spec.org/specsip
- M. Cortes, J. R. Ensor, and J. O. Esteban, “On SIP performance,” Bell Labs Technical Journal, vol. 9, no. 3, pp. 155–172, 2004.

Brief Introduction to SIP



- SIP is a Signaling Protocol. It is designed to set up, manage, and tear down media sessions. SIP messages do not themselves carry media.

- The top two lines show the User Agent registering with a SIP registrar function.
- The Registrar records the UA's current IP address and UDP port in a location data base.
- At some later time, another UA sends an INVITE message to the first UA.
- The INVITE is forwarded by a Proxy that consulted the location data base to learn the destination socket to which the INVITE should be sent.
- The 100 Trying messages are exchanged only between adjacent SIP Functional elements.
- The 180 Ringing is sent end to end from the destination to the originator, to indicate that the called party is being alerted.
- The 200 OK, also sent end-to-end indicates that the called party has answered the call.
- The ACK is sent by the originator to indicate that it has received the information it needs in order to send media to the called party.

Measuring the Session Establishment Rate (SER)

The algorithm (1):

- The algorithm was programmed using the Unix Bash shell scripting language and is available for public download at **//https://github.com/ITM546-PerfSipBench/SIPpScript**
- The algorithm finds the largest session-attempt rate at which the DUT can process requests successfully for a pre-defined, extended period of time with zero-failures.
- The name given to this session rate is “Session Establishment Rate” (SER [6]).
- The period of time is large enough that the DUT can process with zero errors while allowing the system to reach steady state.

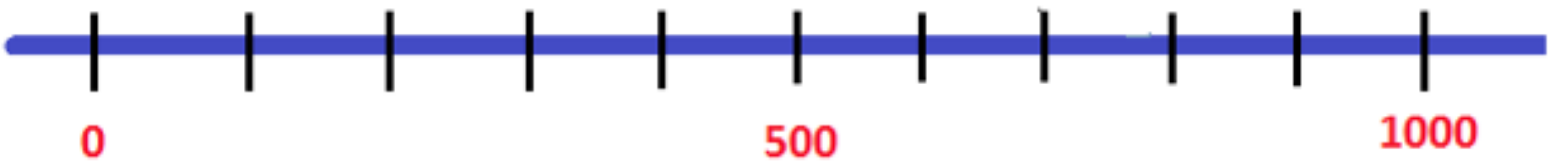
Measuring the Session Establishment Rate (SER)

The algorithm (2):

- In the algorithm we defined a parameter called the Granularity, G.
- This is a measure of how precise we wanted the result to be.
- When the difference between the two an upper bound and a lower bound was less than or equal to the granularity, we accepted the lower bound as the SER.
- While one would always want the G=1, we thought that perhaps to save test time an organization might choose a larger G.
- We used G=3 in our tests often to save time.
- In view of the results that we achieved, the algorithm was changed so as not to include the granularity parameter

Measuring the Session Establishment Rate (SER)

Example:

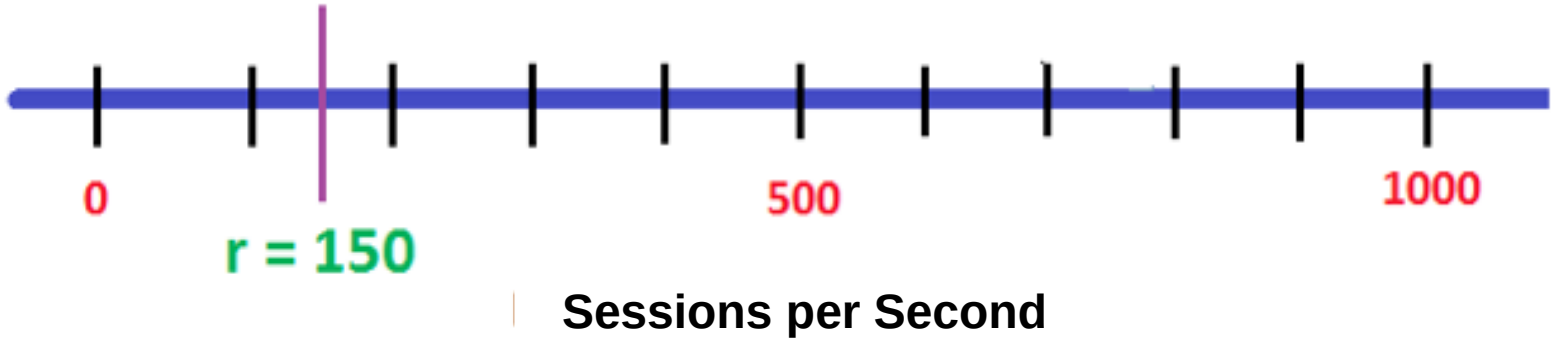


Sessions per Second

- Let the Session Establishment Rates be represented as points on the horizontal axis.
- Here we show rates from 0 session-attempts/second through 1000 session-attempts per second.

Measuring the Session Establishment Rate (SER)

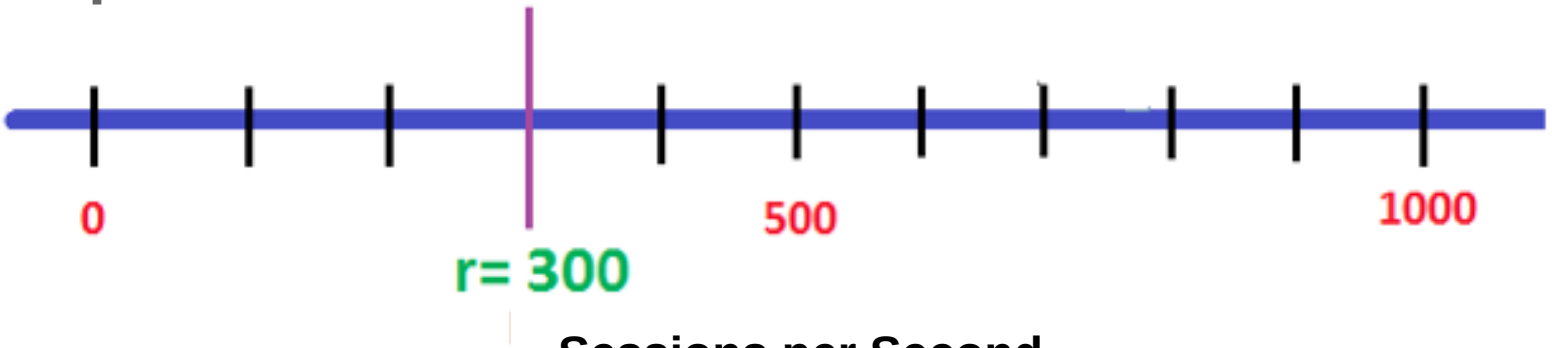
Example:



- The tester sets "r = 150" and runs a load at this rate for the defined length of time.
- Result: No Errors occur during that time at r =150

Measuring the Session Establishment Rate (SER)

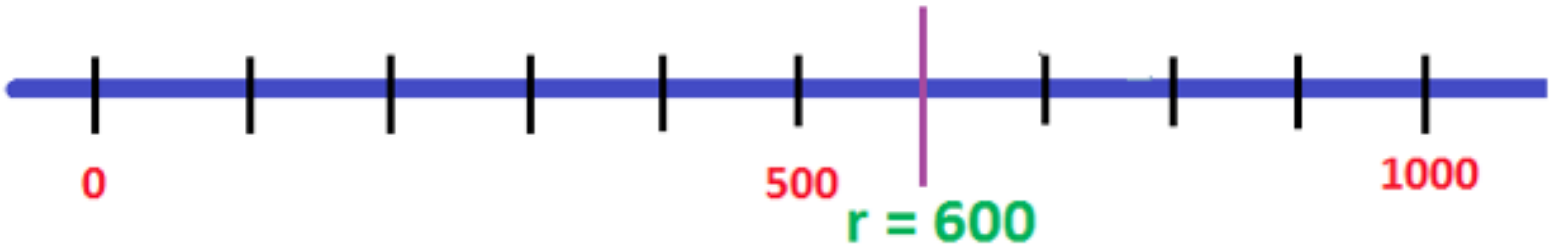
Example:



- Tester increases the value of r to 300, and again runs the load for the defined time interval.
- Result: No error at r = 300

Measuring the Session Establishment Rate (SER)

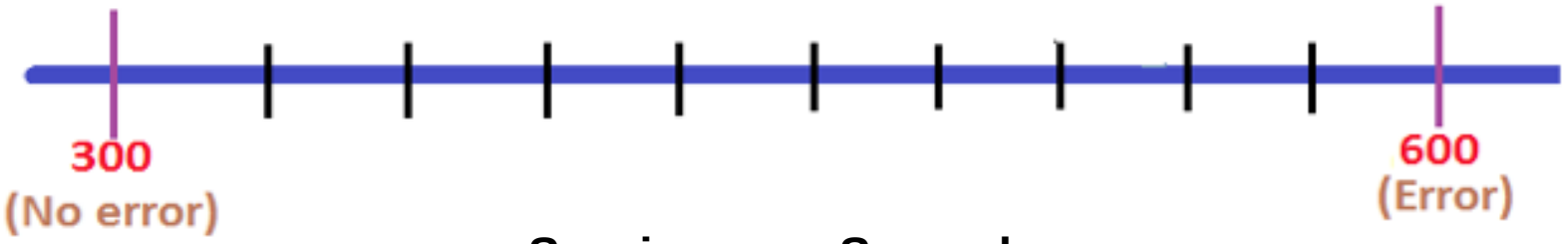
Example:



- The tester again doubles the rate to r=600 and runs the load for the defined time interval.
- Result: Error at r = 600

Measuring the Session Establishment Rate (SER)

Example:

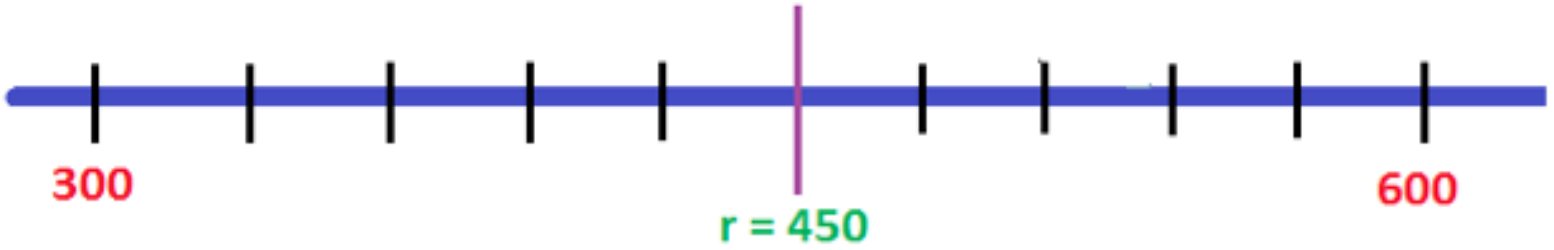


Sessions per Second

- So now we have the lower boundary of r = 300 and an upper boundary of r = 600
- The SER is between these two values.

Measuring the Session Establishment Rate (SER)

Example:

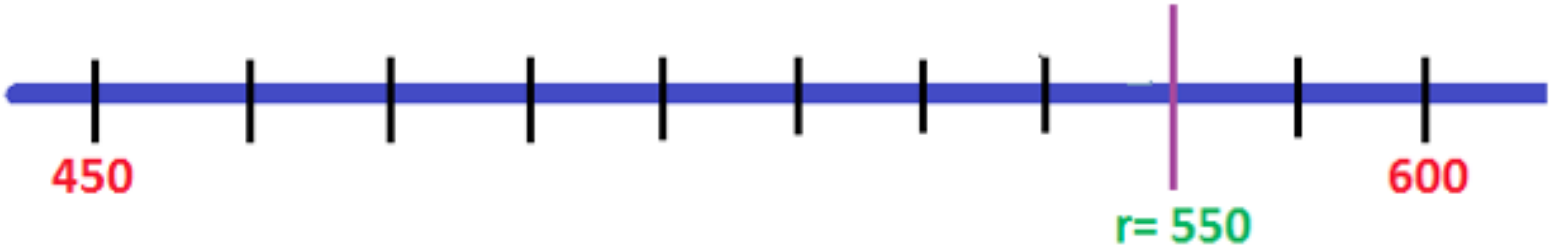


Sessions per Second

- The tester now chooses a new r value, between the highest previous lower limit and the lowest previous upper limit.
- r = 450
- Result: No error at r = 450

Measuring the Session Establishment Rate (SER)

Example:



Sessions per Second

- Again increase the rate to one half way between the last upper bound and the last lower bound. Set r = 550
- Result: Error at r = 550

The SER Algorithm

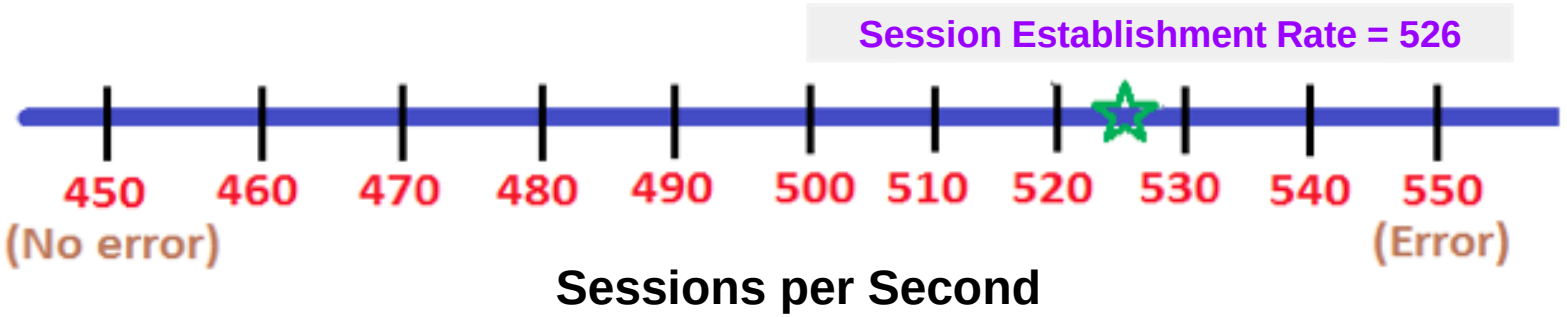
Example:

Sessions per Second

- The lower bound is r = 450 and the new upper bound is r = 550
- The SER is between these two values.

Measuring the Session Establishment Rate (SER)

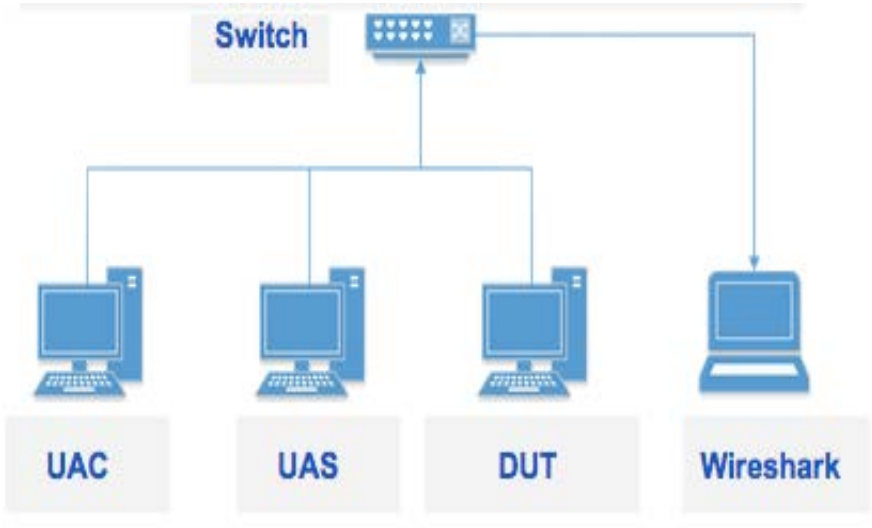
Example:



- This convergent process continues till we find the accurate maximum SER without failure, which is defined to be the Maximum rate r at which all calls succeed. Any higher rate has been shown to produce a failure during the defined time period.

- The actual SER is shown by the green star in the illustration.

Experimental Test Bed



- The UAC and the UAS are components of the SIPp test tool.
- The DUT that we report on in this paper is an Asterisk SIP Server. This free software plays the role of a B2B UA and so may process media as well as signaling messages

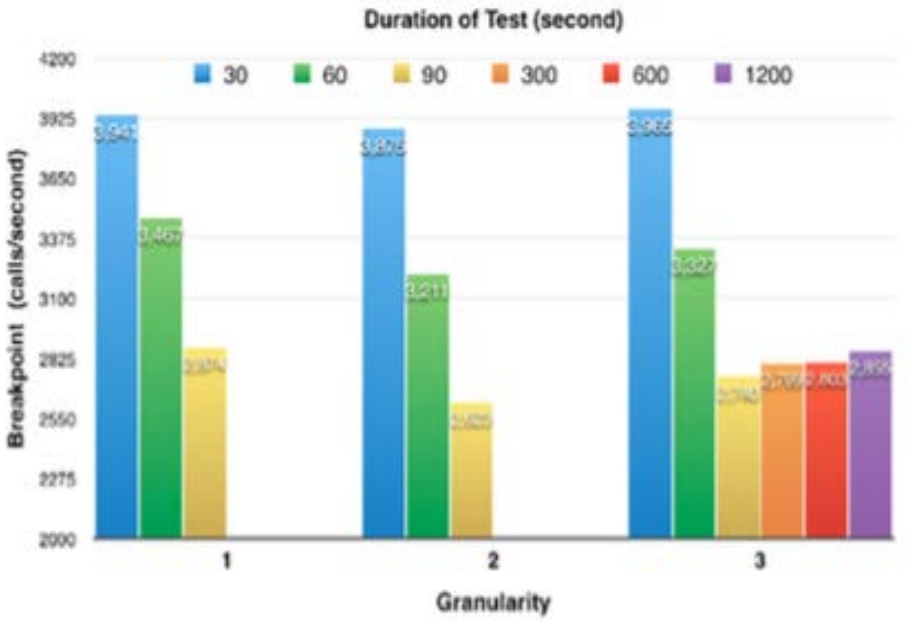
Types of calls tested

- We are interested in the performance of the Signaling Protocol, SIP, not in the overall call performance.
- Using the SIPp test tool we can create call loads that use SIP only or we can create call loads that include a media component.
- We chose to study the behavior of the DUT under both types of load.
- So in the results that follow we provide the SER of the DUT under both types of load.

Testing the Test Harness

- Before testing any DUT, it is necessary to discover the SER of the test harness itself.
- The harness includes the Bash test script, the SIPp load generator, the platforms upon which these run, and the switch that connects them all.
- Use of a slow machine to run the SIPp elements, for example, would limit the rate at which calls are generated, so that carrier-grade DUTs that operate at or near line speed, might not be testable using a test harness unable to generate the load necessary to produce failures.

Testing the Test Harness



Call load does not include RTP

- *All SIP calls were set to last for 9 seconds.*
- *Tests of 30,60 and 90 seconds were followed by longer tests of 5, 10 and 20 minutes, for G=3.*

Granularity comparison:

- Results did not vary much between granularity 1, 2 and 3.
- However, we think that in general a test group will want to use a granularity of 1.

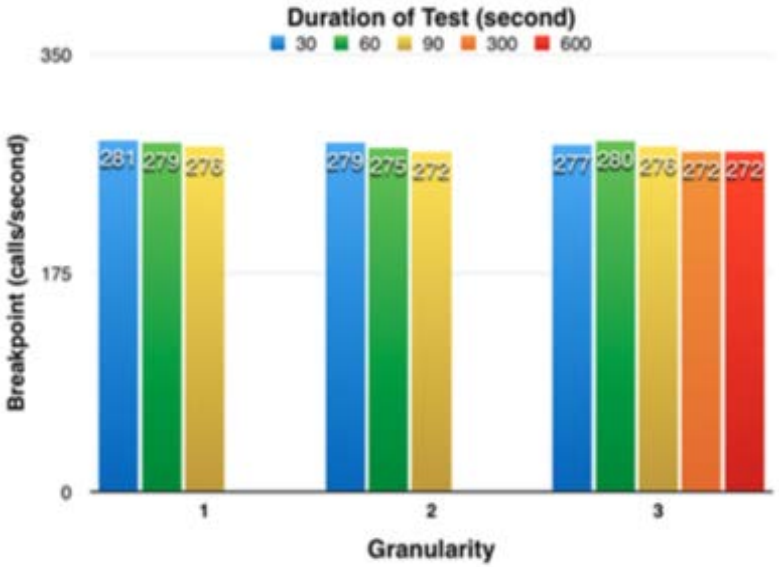
Test Duration Comparison:

- Tests of longer duration had lower SERs.
- This was expected since the probability of an error is greater in proportion to the length of the test.

Conclusion:

- The test harness can deliver a load of 2,700 sps when there is no associated RTP and when the call duration is set for 9 seconds.
- This means that a system that can support a higher call rate than 2,700 sps, cannot be tested with this harness.

Testing the Test Harness



Call load includes RTP

- *All SIP calls were set to last for 9 seconds.*
- *Tests of 30,60 and 90 seconds were followed by longer tests of 5, 10 and 20 minutes for G=3..*

Granularity comparison:

- Results did not vary much between granularity 1, 2 and 3.

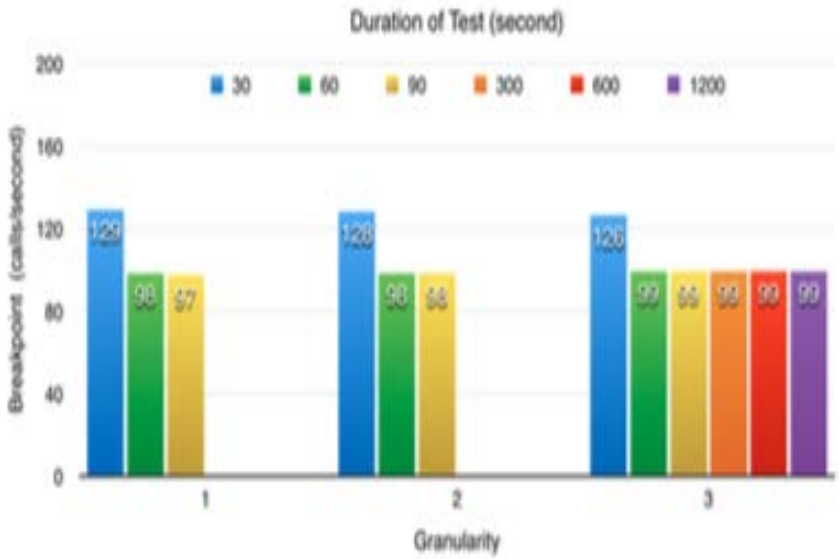
Test Duration Comparison:

- Tests of longer duration had lower SERs.
- This was expected since the probability of an error is greater in proportion to the length of the test.

Conclusion:

- The test harness can deliver a load of 272 sps when RTP is associated with the calls, when the call duration is set for 9 seconds.
- This means that a system that can support a higher call rate than 272 sps with associated RTP, cannot be tested with this harness.

Testing the DUT



Call load without RTP

- *All SIP calls were set to last for 9 seconds.*
- *Tests of 30,60 and 90 seconds were followed by longer tests of 5, 10 and 20 minutes for G=3.*
- *The DUT was configured to handle RTP.*

Granularity comparison:

- Results did not vary much between granularity 1, 2 and 3.

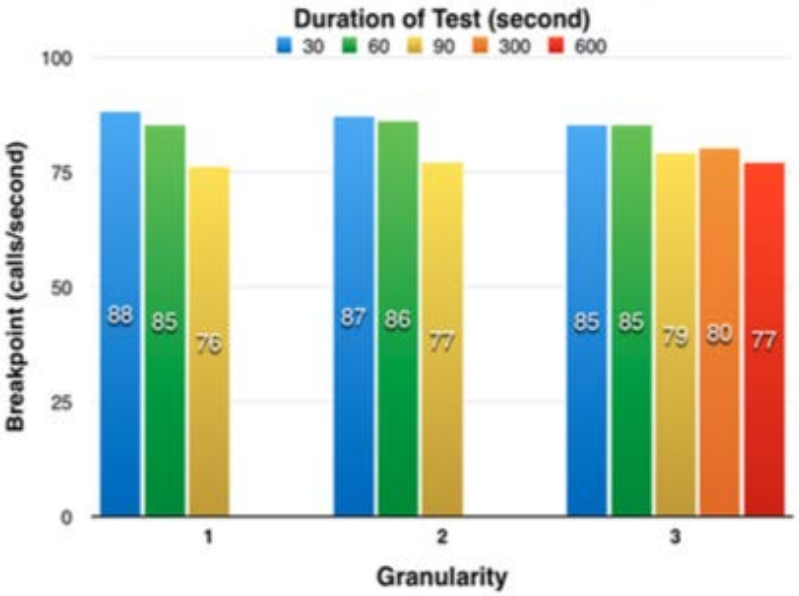
Test Duration Comparison:

- Tests of longer duration had lower SERs.
- The SER for long-duration tests converged to 99 sps.

Conclusion:

- The SER of the DUT is 99 sessions second for a call duration of 9 seconds and when the DUT is set to handle RTP.

Testing the DUT



Call load with RTP

- *All SIP calls were set to last for 9 seconds.*
- *Tests of 30,60 and 90 seconds were followed by longer tests of 5, 10 and 20 minutes for G=3.*
- *The DUT was configured to handle RTP.*

Granularity comparison:

- Results did not vary much between granularity 1, 2 and 3.

Test Duration Comparison:

- Tests of longer duration had lower SERs.
- The SER for long-duration tests appear to converge to a value of 77-80 sps.
- A longer duration of test will be conducted to observe final convergence.

Conclusion:

- The SER of the DUT is ~77 sessions second for a call duration of 9 seconds and when the DUT is set to handle RTP.

Script – DEMO

UAC and UAS are SIPp code.
Goal: Find the largest value of SER.

```
------------------------------ Statistics Screen ------- [1-9]: Change Screen --
  Start Time              | 2014-11-14   12:14:41:890    1415988881.890213
  Last Reset Time         | 2014-11-14   12:15:20:910    1415988920.910413
  Current Time            | 2014-11-14   12:15:20:910    1415988920.910683
--------------------------+---------------------------+--------------------------
  Counter Name            | Periodic value            | Cumulative value
--------------------------+---------------------------+--------------------------
  Elapsed Time            | 00:00:00:000              | 00:00:39:020
  Call Rate               |    0.000 cps              |   76.884 cps
--------------------------+---------------------------+--------------------------
  Incoming call created   |        0                  |        0
  OutGoing call created   |        0                  |     3000
  Total Call created      |                           |     3000
  Current Call            |        0                  |
--------------------------+---------------------------+--------------------------
  Successful call         |        0                  |     3000
  Failed call             |        0                  |        0
--------------------------+---------------------------+--------------------------
  Response Time 1         | 00:00:00:000              | 00:00:00:000
  Call Length             | 00:00:00:000              | 00:00:09:003
------------------------------ Test Terminated --------------------------------
```

Conclusion and Future Work

- **Additional tests**, including **Registration Rate** that are identified in the RFC, using the current test script and also updating the script.
- **Test other SIP Serv**ers, including Open Source offerings such as FreeSwitch and Kamailio, and commercial products.
- Test the same SIP server when running on **different Operating Systems** and with **different parameters**, to compare SER and registration rate variations under these different conditions.

ILLINOIS INSTITUTE OF TECHNOLOGY

Any Questions or comments?