SIP Performance Benchmarking

draft-ietf-bmwg-sip-bench-term-03

draft-ietf-bmwg-sip-bench-meth-03

March 28, 2011

Prof. Carol Davids, Illinois Inst. of Tech.
Dr. Vijay Gurbani, ALU
Scott Poretsky, Allot Communications

Changes in -03

- Terminology
  - Added Figures 3 and 4 (Baseline performance for DUT acting as a UAC and UAS with associated media).
  - Added Figures 8 and 9 (DUT/SUT performance benchmark for session establishment with multiple DUTs. Media flows end-to-end in Figure 8 and hop-by-hop in Figure 9).

Changes in -03

- Methodology:
  - Removed test topologies from Section 3 (caused confusion since these topologies are also presented in the terminology document).
  - Minor edits to align figure numbers.

Experimental data

- Automated tool that collects a subset of these metrics and uses SIPp at its core, was created in the IIT Real-Time Communications Lab.

- Early results are available for Asterisk and OpenSER (kamailio).

Experimental data

- Registration Rate:
- 9 trials each, for Kamailio and for Asterisk. Maximum sessions attempted is set at 50,000
- In the case of Asterisk: 173 rps is the average fastest registration rate achieved across all 9 attempts before the first error is detected.
- In the case of Kamailio: 245 rps is the average fastest registration rate achieved across all 9 attempts before the first error is detected. We are investigating anomalies in this set of data.

Experimental data

- Session Establishment Rate:
- 9 trials each, for Kamailio and for Asterisk. Maximum sessions attempted is set at 50,000
- In the case of Asterisk: 47 sps is the average fastest session attempt rate achieved across all 9 attempts before the first error is detected.
- In the case of Kamailio: 316 sps is the average fastest session attempt rate achieved across all 9 attempts before the first error is detected.

Experimental data

- Analytical data outside scope of draft :
- We collected CPU usage data and also memory usage data while performing the non-automated tests.
- We varied the methodology by making the attempted sps equal the maximum session attempts – hitting the system with the entire load in one second - in what we call ‘avalanche’ testing.
- Both these methods could be used by developers to optimize their systems.

Next steps

- Work is complete.
- Request chair to move the work ahead.
- Early experimental results follow