          Methodology for Benchmarking Session Initiation Protocol (SIP) Devices:
                      Basic Session Setup and Registration

Abstract

   This document provides a methodology for benchmarking the Session
   Initiation Protocol (SIP) performance of devices. Terminology
   related to benchmarking SIP devices is described in the companion
   terminology document (RFC 7501). Using these two documents,
   benchmarks can be obtained and compared for different types of
   devices such as SIP Proxy Servers, Registrars, and Session Border
   Controllers. The term "performance" in this context means the
   capacity of the Device Under Test (DUT) to process SIP messages.
   Media streams are used only to study how they impact the signaling
   behavior. The intent of the two documents is to provide a normalized
   set of tests that will enable an objective comparison of the capacity
   of SIP devices. Test setup parameters and a methodology are
   necessary because SIP allows a wide range of configurations and
   operational conditions that can influence performance benchmark
   measurements.

Copyright Notice

Table of Contents

1. Introduction

   This document describes the methodology for benchmarking Session
   Initiation Protocol (SIP) performance as described in the Terminology
   document [RFC7501]. The methodology and terminology are to be used
   for benchmarking signaling plane performance with varying signaling
   and media load. Media streams, when used, are used only to study how
   they impact the signaling behavior. This document concentrates on
   benchmarking SIP session setup and SIP registrations only.

   The Device Under Test (DUT) is a network intermediary that is RFC
   3261 [RFC3261] capable and that plays the role of a registrar,
   redirect server, stateful proxy, a Session Border Controller (SBC) or
   a B2BUA. This document does not require the intermediary to assume
   the role of a stateless proxy. Benchmarks can be obtained and
   compared for different types of devices such as a SIP proxy server,
   Session Border Controllers (SBC), SIP registrars and a SIP proxy
   server paired with a media relay.

   The test cases provide metrics for benchmarking the maximum 'SIP
   Registration Rate' and maximum 'SIP Session Establishment Rate' that
   the DUT can sustain over an extended period of time without failures
   (extended period of time is defined in the algorithm in
   Section 4.10). Some cases are included to cover encrypted SIP. The
   test topologies that can be used are described in the Test Setup
   section. Topologies in which the DUT handles media as well as those
   in which the DUT does not handle media are both considered. The
   measurement of the performance characteristics of the media itself is
   outside the scope of these documents.

   Benchmark metrics could possibly be impacted by Associated Media.
   The selected values for Session Duration and Media Streams per
   Session enable benchmark metrics to be benchmarked without Associated
   Media. Session Setup Rate could possibly be impacted by the selected
   value for Maximum Sessions Attempted. The benchmark for Session
   Establishment Rate is measured with a fixed value for maximum Session
   Attempts.

   Finally, the overall value of these tests is to serve as a comparison
   function between multiple SIP implementations. One way to use these
   tests is to derive benchmarks with SIP devices from Vendor-A, derive
   a new set of benchmarks with similar SIP devices from Vendor-B and
   perform a comparison on the results of Vendor-A and Vendor-B. This
   document does not make any claims on the interpretation of such
   results.

2. Terminology

   In this document, the key words "MUST", "MUST NOT", "REQUIRED",
   "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT
   RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as
   described in BCP 14, conforming to [RFC2119] and indicate requirement
   levels for compliant implementations.

   RFC 2119 defines the use of these key words to help make the intent
   of Standards Track documents as clear as possible. While this
   document uses these keywords, this document is not a Standards Track
   document.

   Terms specific to SIP [RFC3261] performance benchmarking are defined
   in [RFC7501].

3. Benchmarking Topologies

   Test organizations need to be aware that these tests generate large
   volumes of data and consequently ensure that networking devices like
   hubs, switches, or routers are able to handle the generated volume.

   The test cases enumerated in Sections 6.1 to 6.6 operate on two test
   topologies: one in which the DUT does not process the media
   (Figure 1) and the other in which it does process media (Figure 2).
   In both cases, the tester or Emulated Agent (EA) sends traffic into
   the DUT and absorbs traffic from the DUT. The diagrams in Figures 1
   and 2 represent the logical flow of information and do not dictate a
   particular physical arrangement of the entities.

   Figure 1 depicts a layout in which the DUT is an intermediary between
   the two interfaces of the EA. If the test case requires the exchange
   of media, the media does not flow through the DUT but rather passes
   directly between the two endpoints. Figure 2 shows the DUT as an
   intermediary between the two interfaces of the EA. If the test case
   requires the exchange of media, the media flows through the DUT
   between the endpoints.

```
+--------+   Session   +--------+ Session   +--------+
|        |   Attempt   |        | Attempt   |        |
|        |------------>+        |------------>+        |
|        |             |        |           |        |
|        |   Response  |        | Response  |        |
| Tester +<-----------|  DUT   +<-----------| Tester |
|  (EA)  |             |        |           |  (EA)  |
|        |             |        |           |        |
+--------+             +--------+           +--------+
   /|\                                         /|\
    |            Media (optional)               |
    +=========================================+
```
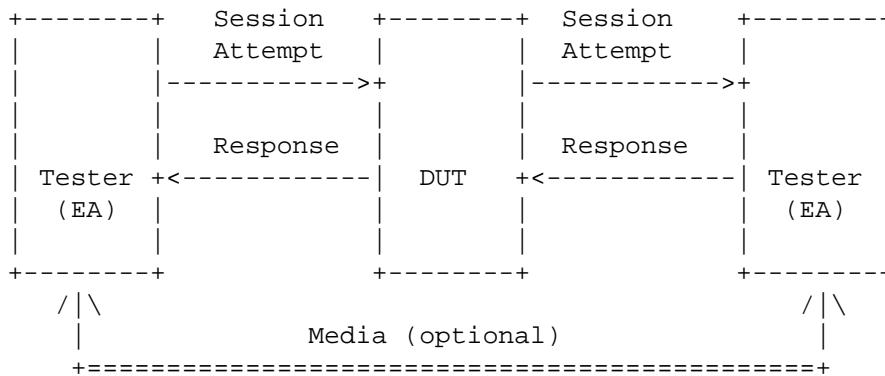
            Figure 1: DUT as an Intermediary, End-to-End Media

```
+--------+   Session   +--------+ Session   +--------+
|        |   Attempt   |        | Attempt   |        |
|        |------------>+        |------------>+        |
|        |             |        |           |        |
|        |   Response  |        | Response  |        |
| Tester +<-----------|  DUT   +<-----------| Tester |
|  (EA)  |             |        |           |  (EA)  |
|        |<==========>|        |           |<==========>|        |
+--------+   Media     +--------+   Media   +--------+
           (Optional)           (Optional)
```
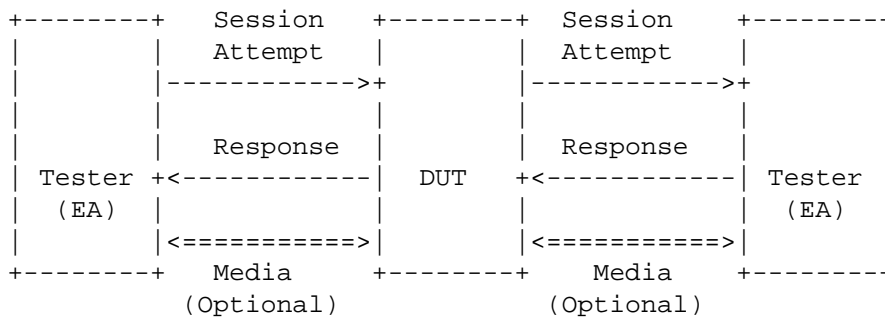
              Figure 2: DUT as an Intermediary Forwarding Media

   The test cases enumerated in Sections 6.7 and 6.8 use the topology in
   Figure 3 below.

```
+--------+ Registration +--------+
|        |   request    |        |
|        |------------->+        |
|        |              |        |
|        |   Response   |        |
| Tester +<-------------|  DUT   |
|  (EA)  |              |        |
|        |              |        |
+--------+              +--------+
```

              Figure 3: Registration and Re-registration Tests

   During registration or re-registration, the DUT may involve backend
   network elements and data stores. These network elements and data
   stores are not shown in Figure 3, but it is understood that they will
   impact the time required for the DUT to generate a response.

This document explicitly separates a registration test (Section 6.7) from a re-registration test (Section 6.8) because in certain networks, the time to re-register may vary from the time to perform an initial registration due to the backend processing involved. It is expected that the registration tests and the re-registration test will be performed with the same set of backend network elements in order to derive a stable metric.

4. Test Setup Parameters

4.1. Selection of SIP Transport Protocol

   Test cases may be performed with any transport protocol supported by SIP. This includes, but is not limited to, TCP, UDP, TLS, and websockets. The protocol used for the SIP transport protocol must be reported with benchmarking results.

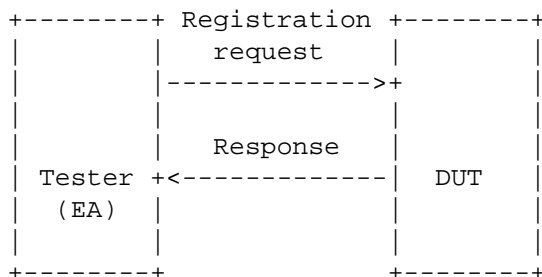   SIP allows a DUT to use different transports for signaling on either side of the connection to the EAs. Therefore, this document assumes that the same transport is used on both sides of the connection; if this is not the case in any of the tests, the transport on each side of the connection MUST be reported in the test-reporting template.

4.2. Connection-Oriented Transport Management

   SIP allows a device to open one connection and send multiple requests over the same connection (responses are normally received over the same connection that the request was sent out on). The protocol also allows a device to open a new connection for each individual request. A connection management strategy will have an impact on the results obtained from the test cases, especially for connection-oriented transports such as TLS. For such transports, the cryptographic handshake must occur every time a connection is opened.

   The connection management strategy, i.e., use of one connection to send all requests or closing an existing connection and opening a new connection to send each request, MUST be reported with the benchmarking result.

4.3. Signaling Server

   The Signaling Server is defined in the companion terminology document ([RFC7501], Section 3.2.2). The Signaling Server is a DUT.

4.4. Associated Media

   Some tests require Associated Media to be present for each SIP
   session. The test topologies to be used when benchmarking DUT
   performance for Associated Media are shown in Figure 1 and Figure 2.

4.5. Selection of Associated Media Protocol

   The test cases specified in this document provide SIP performance
   independent of the protocol used for the media stream. Any media
   protocol supported by SIP may be used. This includes, but is not
   limited to, RTP and SRTP. The protocol used for Associated Media
   MUST be reported with benchmarking results.

4.6. Number of Associated Media Streams per SIP Session

   Benchmarking results may vary with the number of media streams per
   SIP session. When benchmarking a DUT for voice, a single media
   stream is used. When benchmarking a DUT for voice and video, two
   media streams are used. The number of Associated Media Streams MUST
   be reported with benchmarking results.

4.7. Codec Type

   The test cases specified in this document provide SIP performance
   independent of the media stream codec. Any codec supported by the
   EAs may be used. The codec used for Associated Media MUST be
   reported with the benchmarking results.

4.8. Session Duration

   The value of the DUT's performance benchmarks may vary with the
   duration of SIP sessions. Session Duration MUST be reported with
   benchmarking results. A Session Duration of zero seconds indicates
   transmission of a BYE immediately following a successful SIP
   establishment. Setting this parameter to the value '0' indicates
   that a BYE will be sent by the EA immediately after the EA receives a
   200 OK to the INVITE. Setting this parameter to a time value greater
   than the duration of the test indicates that a BYE will never be
   sent. Setting this parameter to a time value greater than the
   duration of the test indicates that a BYE is never sent.

4.9. Attempted Sessions per Second (sps)

   The value of the DUT's performance benchmarks may vary with the
   Session Attempt Rate offered by the tester. Session Attempt Rate
   MUST be reported with the benchmarking results.

   The test cases enumerated in Sections 6.1 to 6.6 require that the EA
   is configured to send the final 2xx-class response as quickly as it
   can. This document does not require the tester to add any delay
   between receiving a request and generating a final response.

4.10. Benchmarking Algorithm

   In order to benchmark the test cases uniformly in Section 6, the
   algorithm described in this section should be used. A prosaic
   description of the algorithm and a pseudocode description are
   provided below, and a simulation written in the R statistical
   language [Rtool] is provided in Appendix A.

   The goal is to find the largest value, R, a SIP Session Attempt Rate,
   measured in sessions per second (sps), which the DUT can process with
   zero errors over a defined, extended period. This period is defined
   as the amount of time needed to attempt N SIP sessions, where N is a
   parameter of test, at the attempt rate, R. An iterative process is
   used to find this rate. The algorithm corresponding to this process
   converges to R.

   If the DUT vendor provides a value for R, the tester can use this
   value. In cases where the DUT vendor does not provide a value for R,
   or where the tester wants to establish the R of a system using local
   media characteristics, the algorithm should be run by setting "r",
   the session attempt rate, equal to a value of the tester's choice.
   For example, the tester may initialize "r = 100" to start the
   algorithm and observe the value at convergence. The algorithm
   dynamically increases and decreases "r" as it converges to the
   maximum sps value for R. The dynamic increase and decrease rate is
   controlled by the weights "w" and "d", respectively.

   The pseudocode corresponding to the description above follows, and a
   simulation written in the R statistical language is provided in
   Appendix A.

```
        ; ---- Parameters of test; adjust as needed
       N  := 50000  ; Global maximum; once largest session rate has
                    ; been established, send this many requests before
                    ; calling the test a success
       m  := {...}  ; Other attributes that affect testing, such
                    ; as media streams, etc.
       r  := 100    ; Initial session attempt rate (in sessions/sec).
                    ; Adjust as needed (for example, if DUT can handle
                    ; thousands of calls in steady state, set to
                    ; appropriate value in the thousands).
       w  := 0.10   ; Traffic increase weight (0 < w <= 1.0)
       d  := max(0.10, w / 2)    ; Traffic decrease weight

        ; ---- End of parameters of test

        proc find_R

           R = max_sps(r, m, N)  ; Setup r sps, each with m media
           ; characteristics until N sessions have been attempted.
           ; Note that if a DUT vendor provides this number, the tester
           ; can use the number as a Session Attempt Rate, R, instead
           ; of invoking max_sps()

        end proc

        ; Iterative process to figure out the largest number of
        ; sps that we can achieve in order to setup n sessions.
        ; This function converges to R, the Session Attempt Rate.
        proc max_sps(r, m, n)
            s     := 0    ; session setup rate
            old_r := 0    ; old session setup rate
            h     := 0    ; Return value, R
            count := 0

            ; Note that if w is small (say, 0.10) and r is small
            ; (say, <= 9), the algorithm will not converge since it
            ; uses floor() to increment r dynamically.  It is best
            ; to start with the defaults (w = 0.10 and r >= 100).

            while (TRUE) {
                s := send_traffic(r, m, n) ; Send r sps, with m media
                ; characteristics until n sessions have been attempted.
                if (s == n)  {
                    if (r > old_r)  {
                        old_r = r
                    }
                    else  {
                        count = count + 1
```

```
                        if (count >= 10)  {
                            # We've converged.
                            h := max(r, old_r)
                            break
                        }
                    }

                    r  := floor(r + (w * r))
                }
                else  {
                    r := floor(r - (d * r))
                    d := max(0.10, d / 2)
                    w := max(0.10, w / 2)
                }

            }
            return h
        end proc
```

5. Reporting Format

5.1. Test Setup Report

    SIP Transport Protocol = _____
    (valid values: TCP|UDP|TLS|SCTP|websockets|specify-other)
    (Specify if same transport used for connections to the DUT
    and connections from the DUT. If different transports
    used on each connection, enumerate the transports used.)

    Connection management strategy for connection oriented
    transports
        DUT receives requests on one connection = _____
        (Yes or no. If no, DUT accepts a new connection for
        every incoming request, sends a response on that
        connection, and closes the connection.)
        DUT sends requests on one connection = _____
        (Yes or no. If no, DUT initiates a new connection to
        send out each request, gets a response on that
        connection, and closes the connection.)

    Session Attempt Rate _____
    (Session attempts/sec)
    (The initial value for "r" in benchmarking algorithm in
    Section 4.10.)

    Session Duration = _____
    (In seconds)

```
        Total Sessions Attempted = _____
        (Total sessions to be created over duration of test)

        Media Streams per Session = _____
        (number of streams per session)

        Associated Media Protocol =  _____
        (RTP|SRTP|specify-other)

        Codec = _____
        (Codec type as identified by the organization that
        specifies the codec)

        Media Packet Size (audio only) =  _____
        (Number of bytes in an audio packet)

        Establishment Threshold time =  _____
        (Seconds)

        TLS ciphersuite used
        (for tests involving TLS) = _____
        (e.g., TLS_RSA_WITH_AES_128_CBC_SHA)

        IPsec profile used
        (For tests involving IPsec) = _____
```

5.2. Device Benchmarks for Session Setup

```
        Session Establishment Rate, "R" = _____
        (sessions per second)
        Is DUT acting as a media relay? (yes/no) = _____
```

5.3. Device Benchmarks for Registrations

```
        Registration Rate =  _____
        (registrations per second)

        Re-registration Rate =  _____
        (registrations per second)

        Notes = _____
        (List any specific backend processing required or
        other parameters that may impact the rate)
```

6. Test Cases

6.1. Baseline Session Establishment Rate of the Testbed

   Objective:
      To benchmark the Session Establishment Rate of the Emulated Agent
      (EA) with zero failures.

   Procedure:
      1. Configure the DUT in the test topology shown in Figure 1.
      2. Set Media Streams per Session to 0.
      3. Execute benchmarking algorithm as defined in Section 4.10 to
          get the baseline Session Establishment Rate. This rate MUST
          be recorded using any pertinent parameters as shown in the
          reporting format of Section 5.1.

   Expected Results: This is the scenario to obtain the maximum Session
      Establishment Rate of the EA and the testbed when no DUT is
      present. The results of this test might be used to normalize test
      results performed on different testbeds or simply to better
      understand the impact of the DUT on the testbed in question.

6.2. Session Establishment Rate without Media

   Objective:
      To benchmark the Session Establishment Rate of the DUT with no
      Associated Media and zero failures.

   Procedure:
      1. Configure a DUT according to the test topology shown in
          Figure 1 or Figure 2.
      2. Set Media Streams per Session to 0.
      3. Execute benchmarking algorithm as defined in Section 4.10 to
          get the Session Establishment Rate. This rate MUST be
          recorded using any pertinent parameters as shown in the
          reporting format of Section 5.1.

   Expected Results: Find the Session Establishment Rate of the DUT
      when the EA is not sending media streams.

6.3. Session Establishment Rate with Media Not on DUT

   Objective:
      To benchmark the Session Establishment Rate of the DUT with zero
      failures when Associated Media is included in the benchmark test
      but the media is not running through the DUT.

   Procedure:
        1. Configure a DUT according to the test topology shown in
            Figure 1.
        2. Set Media Streams per Session to 1.
        3. Execute benchmarking algorithm as defined in Section 4.10 to
            get the session establishment rate with media. This rate MUST
            be recorded using any pertinent parameters as shown in the
            reporting format of Section 5.1.

   Expected Results: Session Establishment Rate results obtained with
      Associated Media with any number of media streams per SIP session
      are expected to be identical to the Session Establishment Rate
      results obtained without media in the case where the DUT is
      running on a platform separate from the Media Relay.

6.4.   Session Establishment Rate with Media on DUT

   Objective:
      To benchmark the Session Establishment Rate of the DUT with zero
      failures when Associated Media is included in the benchmark test
      and the media is running through the DUT.

   Procedure:
        1. Configure a DUT according to the test topology shown in
            Figure 2.
        2. Set Media Streams per Session to 1.
        3. Execute benchmarking algorithm as defined in Section 4.10 to
            get the Session Establishment Rate with media. This rate MUST
            be recorded using any pertinent parameters as shown in the
            reporting format of Section 5.1.

   Expected Results: Session Establishment Rate results obtained with
      Associated Media may be lower than those obtained without media in
      the case where the DUT and the Media Relay are running on the same
      platform. It may be helpful for the tester to be aware of the
      reasons for this degradation, although these reasons are not
      parameters of the test. For example, the degree of performance
      degradation may be due to what the DUT does with the media (e.g.,
      relaying vs. transcoding), the type of media (audio vs. video vs.
      data), and the codec used for the media. There may also be cases
      where there is no performance impact, if the DUT has dedicated
      media-path hardware.

6.5.   Session Establishment Rate with TLS-Encrypted SIP

   Objective:
      To benchmark the Session Establishment Rate of the DUT with zero
      failures when using TLS-encrypted SIP signaling.

   Procedure:
        1. If the DUT is being benchmarked as a proxy or B2BUA, then
            configure the DUT in the test topology shown in Figure 1 or
            Figure 2.
        2. Configure the tester to enable TLS over the transport being
            used during benchmarking. Note the ciphersuite being used for
            TLS and record it in Section 5.1.
        3. Set Media Streams per Session to 0 (media is not used in this
            test).
        4. Execute benchmarking algorithm as defined in Section 4.10 to
            get the Session Establishment Rate with TLS encryption.

   Expected Results: Session Establishment Rate results obtained with
      TLS-encrypted SIP may be lower than those obtained with plaintext
      SIP.

6.6. Session Establishment Rate with IPsec-Encrypted SIP

   Objective:
      To benchmark the Session Establishment Rate of the DUT with zero
      failures when using IPsec-encrypted SIP signaling.

   Procedure:
        1. Configure a DUT according to the test topology shown in
            Figure 1 or Figure 2.
        2. Set Media Streams per Session to 0 (media is not used in this
            test).
        3. Configure tester for IPsec. Note the IPsec profile being used
            for IPsec and record it in Section 5.1.
        4. Execute benchmarking algorithm as defined in Section 4.10 to
            get the Session Establishment Rate with encryption.

   Expected Results: Session Establishment Rate results obtained with
      IPsec-encrypted SIP may be lower than those obtained with
      plaintext SIP.

6.7. Registration Rate

   Objective:
      To benchmark the maximum registration rate the DUT can handle over
      an extended time period with zero failures.

   Procedure:
        1. Configure a DUT according to the test topology shown in
            Figure 3.
        2. Set the registration timeout value to at least 3600 seconds.
        3. Each register request MUST be made to a distinct Address of
            Record (AoR). Execute benchmarking algorithm as defined in

Section 4.10 to get the maximum registration rate. This rate
MUST be recorded using any pertinent parameters as shown in
the reporting format of Section 5.1. For example, the use of
TLS or IPsec during registration must be noted in the
reporting format. In the same vein, any specific backend
processing (use of databases, authentication servers, etc.)
SHOULD be recorded as well.

Expected Results: Provides a maximum registration rate.

6.8. Re-registration Rate

Objective:
To benchmark the re-registration rate of the DUT with zero
failures using the same backend processing and parameters used
during Section 6.7.

Procedure:
1. Configure a DUT according to the test topology shown in
Figure 3.
2. Execute the test detailed in Section 6.7 to register the
endpoints with the registrar and obtain the registration rate.
3. After at least 5 minutes of performing Step 2, but no more
than 10 minutes after Step 2 has been performed, re-register
the same AoRs used in Step 3 of Section 6.7. This will count
as a re-registration because the SIP AoRs have not yet
expired.

Expected Results: Note the rate obtained through this test for
comparison with the rate obtained in Section 6.7.

7. Security Considerations

Documents of this type do not directly affect the security of the
Internet or corporate networks as long as benchmarking is not
performed on devices or systems connected to production networks.
Security threats and how to counter these in SIP and the media layer
is discussed in RFC 3261, RFC 3550, and RFC 3711, and various other
documents. This document attempts to formalize a set of common
methodology for benchmarking performance of SIP devices in a lab
environment.

8. References

8.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, March 1997,
              <http://www.rfc-editor.org/info/rfc2119>.

   [RFC7501] Davids, C., Gurbani, V., and S. Poretsky, "Terminology for
              Benchmarking Session Initiation Protocol (SIP) Devices:
              Basic Session Setup and Registration", RFC 7501, April
              2015, <http://www.rfc-editor.org/info/rfc7501>.

8.2. Informative References

   [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,
              A., Peterson, J., Sparks, R., Handley, M., and E.
              Schooler, "SIP: Session Initiation Protocol", RFC 3261,
              June 2002, <http://www.rfc-editor.org/info/rfc3261>.

   [Rtool]   R Development Core Team, "R: A Language and Environment
              for Statistical Computing", R Foundation for Statistical
              Computing Vienna, Austria, ISBN 3-900051-07-0, 2011,
              <http://www.R-project.org>.

Appendix A. R Code Component to Simulate Benchmarking Algorithm

```
     # Copyright (c) 2015 IETF Trust and the persons identified as
     # authors of the code.  All rights reserved.
     #
     # Redistribution and use in source and binary forms, with or
     # without modification, are permitted provided that the following
     # conditions are met:
     #
     # The author of this code is Vijay K. Gurbani.
     #
     # - Redistributions of source code must retain the above copyright
     #   notice, this list of conditions and
     #   the following disclaimer.
     #
     # - Redistributions in binary form must reproduce the above
     #   copyright notice, this list of conditions and the following
     #   disclaimer in the documentation and/or other materials
     #   provided with the distribution.
     #
     # - Neither the name of Internet Society, IETF or IETF Trust,
     #   nor the names of specific contributors, may be used to
     #   endorse or promote products derived from this software
     #   without specific prior written permission.
     #
     # THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
     # CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
     # INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
     # MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
     # DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
     # CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
     # INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
     # (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
     # GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
     # INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
     # WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
     # NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
     # OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
     # DAMAGE.

     w = 0.10
     d = max(0.10, w / 2)
     DUT_max_sps = 460      # Change as needed to set the max sps value
                           # for a DUT
```

```
   # Returns R, given r (initial session attempt rate).
   # E.g., assume that a DUT handles 460 sps in steady state
   # and you have saved this code in a file simulate.r.  Then,
   # start an R session and do the following:
   #
   # > source("simulate.r")
   # > find_R(100)
   # ... debug output omitted ...
   # [1] 458
   #
   # Thus, the max sps that the DUT can handle is 458 sps, which is
   # close to the absolute maximum of 460 sps the DUT is specified to
   # do.
   find_R <- function(r)  {
      s     = 0
      old_r = 0
      h     = 0
      count = 0

      # Note that if w is small (say, 0.10) and r is small
      # (say, <= 9), the algorithm will not converge since it
      # uses floor() to increment r dynamically.  It is best
      # to start with the defaults (w = 0.10 and r >= 100).

      cat("r   old_r    w     d \n")
      while (TRUE)  {
         cat(r, ' ', old_r, ' ', w, ' ', d, '\n')
         s = send_traffic(r)
         if (s == TRUE)  {      # All sessions succeeded

             if (r > old_r)  {
                 old_r = r
             }
             else  {
                 count = count + 1

                 if (count >= 10)  {
                     # We've converged.
                     h = max(r, old_r)
                     break
                 }
             }

             r  = floor(r + (w * r))
         }
```

```
        else  {
            r = floor(r - (d * r))
            d = max(0.10, d / 2)
            w = max(0.10, w / 2)
        }
      }

      h
    }

    send_traffic <- function(r)  {
      n = TRUE

      if (r > DUT_max_sps)  {
          n = FALSE
      }

      n
    }
```

Acknowledgments

Authors' Addresses

   Carol Davids
   Illinois Institute of Technology
   201 East Loop Road
   Wheaton, IL 60187
   United States

   Phone: +1 630 682 6024
   EMail: davids@iit.edu

   Vijay K. Gurbani
   Bell Laboratories, Alcatel-Lucent
   1960 Lucent Lane
   Rm 9C-533
   Naperville, IL 60566
   United States

   Phone: +1 630 224 0216
   EMail: vkg@bell-labs.com

   Scott Poretsky
   Allot Communications
   300 TradeCenter, Suite 4680
   Woburn, MA 08101
   United States

   Phone: +1 508 309 2179
   EMail: sporetsky@allot.com