Benchmarking Methodology for IPv6 Transition Technologies
draft-ietf-bmwg-ipv6-tran-tech-benchmarking-07.txt

Abstract

   There are benchmarking methodologies addressing the performance of
   network interconnect devices that are IPv4- or IPv6-capable, but the
   IPv6 transition technologies are outside of their scope. This
   document provides complementary guidelines for evaluating the
   performance of IPv6 transition technologies. More specifically,
   this document targets IPv6 transition technologies that employ
   encapsulation or translation mechanisms, as dual-stack nodes can be
   very well tested using the recommendations of RFC2544 and RFC5180.
   The methodology also includes a metric for benchmarking load
   scalability.

Copyright Notice

Table of Contents

1. Introduction

   The methodologies described in [RFC2544] and [RFC5180] help vendors
   and network operators alike analyze the performance of IPv4 and
   IPv6-capable network devices. The methodology presented in [RFC2544]
   is mostly IP version independent, while [RFC5180] contains
   complementary recommendations, which are specific to the latest IP
   version, IPv6. However, [RFC5180] does not cover IPv6 transition
   technologies.

   IPv6 is not backwards compatible, which means that IPv4-only nodes
   cannot directly communicate with IPv6-only nodes. To solve this
   issue, IPv6 transition technologies have been proposed and
   implemented.

   This document presents benchmarking guidelines dedicated to IPv6
   transition technologies. The benchmarking tests can provide insights
   about the performance of these technologies, which can act as useful
   feedback for developers, as well as for network operators going
   through the IPv6 transition process.

The document also includes an approach to quantify performance when operating in overload. Overload scalability can be defined as a system's ability to gracefully accommodate greater numbers of flows than the maximum number of flows which the Device under test (DUT) can operate normally. The approach taken here is to quantify the overload scalability by measuring the performance created by an excessive number of network flows, and comparing performance to the non-overloaded case.

1.1. IPv6 Transition Technologies

Two of the basic transition technologies, dual IP layer (also known as dual stack) and encapsulation are presented in [RFC4213]. IPv4/IPv6 Translation is presented in [RFC6144]. Most of the transition technologies employ at least one variation of these mechanisms. In this context, a generic classification of the transition technologies can prove useful.

We can consider a production network transitioning to IPv6 as being constructed using the following IP domains:

o Domain A: IPvX specific domain

o Core domain: which may be IPvY specific or dual-stack(IPvX and IPvY)

o Domain B: IPvX specific domain

Note: X,Y are part of the set {4,6}, and X NOT.EQUAL Y.

According to the technology used for the core domain traversal the transition technologies can be categorized as follows:

1. Dual-stack: the core domain devices implement both IP protocols.

2. Single Translation: In this case, the production network is assumed to have only two domains, Domain A and the Core domain. The core domain is assumed to be IPvY specific. IPvX packets are translated to IPvY at the edge between Domain A and the Core domain.

3. Double translation: The production network is assumed to have all three domains; Domains A and B are IPvX specific, while the core domain is IPvY specific. A translation mechanism is employed for the traversal of the core network. The IPvX packets are translated to IPvY packets at the edge between Domain A and the Core domain. Subsequently, the IPvY packets are translated back to IPvX at the edge between the Core domain and Domain B.

   4. Encapsulation: The production network is assumed to have all
      three domains; Domains A and B are IPvX specific, while the core
      domain is IPvY specific. An encapsulation mechanism is used to
      traverse the core domain. The IPvX packets are encapsulated to
      IPvY packets at the edge between Domain A and the Core domain.
      Subsequently, the IPvY packets are de-encapsulated at the edge
      between the Core domain and Domain B.

   The performance of Dual-stack transition technologies can be fully
   evaluated using the benchmarking methodologies presented by
   [RFC2544] and [RFC5180]. Consequently, this document focuses on the
   other 3 categories: Single translation, Encapsulation and Double
   translation transition technologies.

   Another important aspect by which the IPv6 transition technologies
   can be categorized is their use of stateful or stateless mapping
   algorithms. The technologies that use stateful mapping algorithms
   (e.g. Stateful NAT64 [RFC6146]) create dynamic correlations between
   IP addresses or {IP address, transport protocol, transport port
   number} tuples, which are stored in a state table. For ease of
   reference, the IPv6 transition technologies which employ stateful
   mapping algorithms will be called stateful IPv6 transition
   technologies. The efficiency with which the state table is managed
   can be an important performance indicator for these technologies.
   Hence, for the stateful IPv6 transition technologies additional
   benchmarking tests are RECOMMENDED.

   Table 1 contains the generic categories as well as associations with
   some of the IPv6 transition technologies proposed in the IETF.

            Table 1. IPv6 Transition Technologies Categories

	Generic category	IPv6 Transition Technology
1	Dual-stack	Dual IP Layer Operations [RFC4213]
2	Single translation	NAT64 [RFC6146], IVI [RFC6219]
3	Double translation	464XLAT [RFC6877], MAP-T [RFC7599]
4	Encapsulation	DSLite[RFC6333], MAP-E [RFC7597]
		Lightweight 4over6 [RFC7596]
		6RD [RFC5569], 6PE [RFC4798], 6VPE
		6VPE [RFC4659]

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [RFC2119] significance.

Although these terms are usually associated with protocol requirements, in this document the terms are requirements for users and systems that intend to implement the test conditions and claim conformance with this specification.

3. Terminology

A number of terms used in this memo have been defined in other RFCs. Please refer to those RFCs for definitions, testing procedures and reporting formats.

Throughput (Benchmark) - [RFC2544]

Frame Loss Rate (Benchmark) - [RFC2544]

Back-to-back Frames (Benchmark) - [RFC2544]

System Recovery (Benchmark) - [RFC2544]

Reset (Benchmark) - [RFC6201]

Concurrent TCP Connection Capacity (Benchmark) - [RFC3511]

Maximum TCP Connection Establishment Rate (Benchmark) - [RFC3511]

4. Test Setup

The test environment setup options recommended for IPv6 transition technologies benchmarking are very similar to the ones presented in Section 6 of [RFC2544]. In the case of the tester setup, the options presented in [RFC2544] and [RFC5180] can be applied here as well. However, the Device under test (DUT) setup options should be explained in the context of the targeted categories of IPv6 transition technologies: Single translation, Double translation and Encapsulation transition technologies.

   Although both single tester and sender/receiver setups are
   applicable to this methodology, the single tester setup will be used
   to describe the DUT setup options.

   For the test setups presented in this memo, dynamic routing SHOULD
   be employed. However, the presence of routing and management frames
   can represent unwanted background data that can affect the
   benchmarking result. To that end, the procedures defined in
   [RFC2544] (Sections 11.2 and 11.3) related to routing and management
   frames SHOULD be used here. Moreover, the "Trial description"
   recommendations presented in [RFC2544] (Section 23) are also valid
   for this memo.

   In terms of route setup, the recommendations of [RFC2544] Section 13
   are valid for this document assuming that an IPv6 version of the
   routing packets shown in appendix C.2.6.2 is used.

4.1. Single translation Transition Technologies

   For the evaluation of Single translation transition technologies, a
   single DUT setup (see Figure 1) SHOULD be used. The DUT is
   responsible for translating the IPvX packets into IPvY packets. In
   this context, the tester device SHOULD be configured to support both
   IPvX and IPvY.
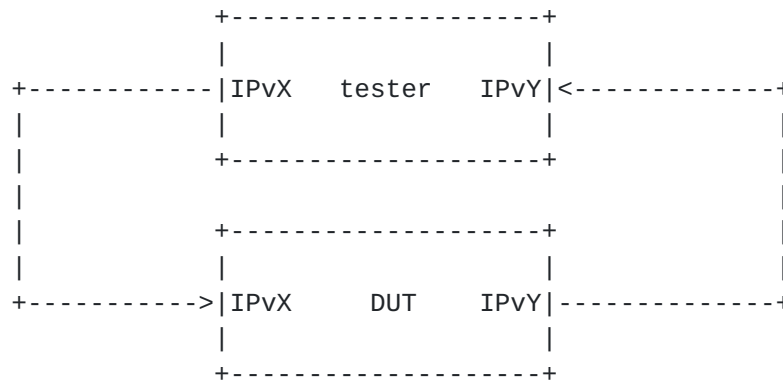
```
                      +--------------------+
                      |                    |
        +------------|IPvX    tester   IPvY|<-------------+
        |             |                    |              |
        |             +--------------------+              |
        |                                                 |
        |             +-------------------+               |
        |             |                   |               |
        +----------->|IPvX      DUT    IPvY|--------------+
                      |                    |
                      +-------------------+
                      Figure 1. Test setup 1
```

4.2. Encapsulation/Double translation Transition Technologies

   For evaluating the performance of Encapsulation and Double
   translation transition technologies, a dual DUT setup (see Figure 2)
   SHOULD be employed. The tester creates a network flow of IPvX
   packets. The first DUT is responsible for the encapsulation or
   translation of IPvX packets into IPvY packets. The IPvY packets are
   de-encapsulated/translated back to IPvX packets by the second DUT
   and forwarded to the tester.

```
                    +-------------------+
                    |                   |
    +-------------------|IPvX   tester   IPvX|<------------------+
    |                   |                   |                    |
    |                   +-------------------+                    |
    |                                                            |
    |       +-------------------+      +-------------------+     |
    |       |                   |      |                   |     |
    +----->|IPvX    DUT 1  IPvY |----->|IPvY   DUT 2   IPvX |------+
            |                   |      |                   |
            +-------------------+      +-------------------+
                       Figure 2. Test setup 2
```

One of the limitations of the dual DUT setup is the inability to
reflect asymmetries in behavior between the DUTs. Considering this,
additional performance tests SHOULD be performed using the single
DUT setup.

Note: For encapsulation IPv6 transition technologies, in the single
DUT setup, in order to test the de-encapsulation efficiency, the
tester SHOULD be able to send IPvX packets encasulated as IPvY.

5. Test Traffic

The test traffic represents the experimental workload and SHOULD
meet the requirements specified in this section. The requirements
are dedicated to unicast IP traffic. Multicast IP traffic is outside
of the scope of this document.

5.1. Frame Formats and Sizes

[RFC5180] describes the frame size requirements for two commonly
used media types: Ethernet and SONET (Synchronous Optical Network).
[RFC2544] covers also other media types, such as token ring and
FDDI. The recommendations of the two documents can be used for the
dual-stack transition technologies. For the rest of the transition
technologies, the frame overhead introduced by translation or
encapsulation MUST be considered.

The encapsulation/translation process generates different size
frames on different segments of the test setup. For instance, the
single translation transition technologies will create different
frame sizes on the receiving segment of the test setup, as IPvX
packets are translated to IPvY. This is not a problem if the
bandwidth of the employed media is not exceeded. To prevent
exceeding the limitations imposed by the media, the frame size
overhead needs to be taken into account when calculating the maximum
theoretical frame rates. The calculation method for the Ethernet, as

   well as a calculation example, are detailed in Appendix A. The
   details of the media employed for the benchmarking tests MUST be
   noted in all test reports.

   In the context of frame size overhead, MTU recommendations are
   needed in order to avoid frame loss due to MTU mismatch between the
   virtual encapsulation/translation interfaces and the physical
   network interface controllers (NICs). To avoid this situation, the
   larger MTU between the physical NICs and virtual
   encapsulation/translation interfaces SHOULD be set for all
   interfaces of the DUT and tester. To be more specific, the minimum
   IPv6 MTU size (1280 bytes) plus the encapsulation/translation
   overhead is the RECOMMENDED value for the physical interfaces as
   well as virtual ones.

5.1.1. **Frame Sizes to Be Used over Ethernet**

   Based on the recommendations of [RFC5180], the following frame sizes
   SHOULD be used for benchmarking IPvX/IPvY traffic on Ethernet links:
   64, 128, 256, 512, 768, 1024, 1280, 1518, 1522, 2048, 4096, 8192 and
   9216.

   Note: for single translation transition technologies (e.g. NAT64) in
   the IPv6 -> IPv4 translation direction, 64 byte frames SHOULD be
   replaced by 84 byte frames. This would allow the frames to be
   transported over media such as the ones described by the IEEE 802.1Q
   standard. Moreover, this would also allow the implementation of a
   frame identifier in the UDP data.

   The theoretical maximum frame rates considering an example of frame
   overhead are presented in Appendix A.

5.2. **Protocol Addresses**

   The selected protocol addresses should follow the recommendations of
   [RFC5180](Section 5) for IPv6 and [RFC2544](Section 12) for IPv4.

   Note: testing traffic with extension headers might not be possible
   for the transition technologies, which employ translation. Proposed
   IPvX/IPvY translation algorithms such as IP/ICMP translation
   [RFC7915] do not support the use of extension headers.

5.3. **Traffic Setup**

   Following the recommendations of [RFC5180], all tests described
   SHOULD be performed with bi-directional traffic. Uni-directional

   traffic tests MAY also be performed for a fine grained performance
   assessment.

   Because of the simplicity of UDP, UDP measurements offer a more
   reliable basis for comparison than other transport layer protocols.
   Consequently, for the benchmarking tests described in Section 7 of
   this document UDP traffic SHOULD be employed.

   Considering that a transition technology could process both native
   IPv6 traffic and translated/encapsulated traffic, the following
   traffic setups are recommended:

   i) IPvX only traffic (where the IPvX traffic is to be
        translated/encapsulated by the DUT)
   ii) 90% IPvX traffic and 10% IPvY native traffic
   iii) 50% IPvX traffic and 50% IPvY native traffic
   iv) 10% IPvX traffic and 90% IPvY native traffic

   For the benchmarks dedicated to stateful IPv6 transition
   technologies, included in Section 8 of this memo (Concurrent TCP
   Connection Capacity and Maximum TCP Connection Establishment Rate),
   the traffic SHOULD follow the recommendations of [RFC3511], Sections
   5.2.2.2 and 5.3.2.2.

6. Modifiers

   The idea of testing under different operational conditions was first
   introduced in [RFC2544](Section 11) and represents an important
   aspect of benchmarking network elements, as it emulates, to some
   extent, the conditions of a production environment. Section 6 of
    [RFC5180] describes complementary testing conditions specific to
   IPv6. Their recommendations can also be followed for IPv6 transition
   technologies testing.

7. Benchmarking Tests

   The following sub-sections contain the list of all recommended
   benchmarking tests.

7.1. Throughput

   Use Section 26.1 of RFC2544 unmodified.

7.2. Latency

   Objective: To determine the latency. Typical latency is based on the
   definitions of latency from [RFC1242]. However, this memo provides a
   new measurement procedure.

   Procedure: Similar to [RFC2544], the throughput for DUT at each of
   the listed frame sizes SHOULD be determined. Send a stream of frames
   at a particular frame size through the DUT at the determined
   throughput rate to a specific destination. The stream SHOULD be at
   least 120 seconds in duration.

   Identifying tags SHOULD be included in at least 500 frames after 60
   seconds. For each tagged frame, the time at which the frame was
   fully transmitted (timestamp A) and the time at which the frame was
   received (timestamp B) MUST be recorded. The latency is timestamp B
   minus timestamp A as per the relevant definition from RFC 1242,
   namely latency as defined for store and forward devices or latency
   as defined for bit forwarding devices.

   We recommend to encode the identifying tag in the payload of the
   frame. To be more exact, the identifier SHOULD be inserted after the
   UDP header.

   From the resulted (at least 500) latencies, 2 quantities SHOULD be
   calculated. One is the typical latency, which SHOULD be calculated
   with the following formula:

   TL=Median(Li)

   Where: TL - the reported typical latency of the stream

   Li -the latency for tagged frame i

   The other measure is the worst case latency, which SHOULD be
   calculated with the following formula:

   WCL=L99.9thPercentile

   Where: WCL - The reported worst case latency

   L99.9thPercentile - The 99.9th Percentile of the stream measured
   latencies

The test MUST be repeated at least 20 times with the reported value being the median of the recorded values for TL and WCL.

Reporting Format: The report MUST state which definition of latency (from RFC 1242) was used for this test. The summarized latency results SHOULD be reported in the format of a table with a row for each of the tested frame sizes. There SHOULD be columns for the frame size, the rate at which the latency test was run for that frame size, for the media types tested, and for the resultant typical latency and worst case latency values for each type of data stream tested. To account for the variation, the 1st and 99th percentiles of the 20 iterations MAY be reported in two separated columns. For a fine grained analysis, the histogram (as exemplified in [RFC5481] Section 4.4) of one of the iterations MAY be displayed .

7.3. Packet Delay Variation

Considering two of the metrics presented in [RFC5481], Packet Delay Variation (PDV) and Inter Packet Delay Variation (IPDV), it is RECOMMENDED to measure PDV. For a fine grained analysis of delay variation, IPDV measurements MAY be performed.

7.3.1. PDV

Objective: To determine the Packet Delay Variation as defined in [RFC5481].

Procedure: As described by [RFC2544], first determine the throughput for the DUT at each of the listed frame sizes. Send a stream of frames at a particular frame size through the DUT at the determined throughput rate to a specific destination. The stream SHOULD be at least 60 seconds in duration. Measure the One-way delay as described by [RFC3393] for all frames in the stream. Calculate the PDV of the stream using the formula:

PDV=D99.9thPercentile - Dmin

Where: D99.9thPercentile - the 99.9th Percentile (as it was described in [RFC5481]) of the One-way delay for the stream

        Dmin - the minimum One-way delay in the stream

As recommended in [RFC2544], the test MUST be repeated at least 20 times with the reported value being the median of the recorded values. Moreover, the 1st and 99th percentiles SHOULD be calculated to account for the variation of the dataset.

   Reporting Format: The PDV results SHOULD be reported in a table with
   a row for each of the tested frame sizes and columns for the frame
   size and the applied frame rate for the tested media types. Two
   columns for the 1st and 99th percentile values MAY be displayed.
   Following the recommendations of [RFC5481], the RECOMMENDED units of
   measurement are milliseconds.

7.3.2. IPDV

   Objective: To determine the Inter Packet Delay Variation as defined
   in [RFC5481].

   Procedure: As described by [RFC2544], first determine the throughput
   for the DUT at each of the listed frame sizes. Send a stream of
   frames at a particular frame size through the DUT at the determined
   throughput rate to a specific destination. The stream SHOULD be at
   least 60 seconds in duration. Measure the One-way delay as described
   by [RFC3393] for all frames in the stream. Calculate the IPDV for
   each of the frames using the formula:

   $IPDV(i)=D(i) - D(i-1)$

   Where: $D(i)$ - the One-way delay of the i th frame in the stream

          $D(i-1)$ - the One-way delay of i-1 th frame in the stream

   Given the nature of IPDV, reporting a single number might lead to
   over-summarization. In this context, the report for each measurement
   SHOULD include 3 values: Dmin, Dmed, and Dmax

   Where: Dmin - the minimum IPDV in the stream

          Dmed - the median IPDV of the stream

          Dmax - the maximum IPDV in the stream

   The test MUST be repeated at least 20 times. To summarize the 20
   repetitions, for each of the 3 (Dmin, Dmed and Dmax) the median
   value SHOULD be reported.

   Reporting format: The median for the 3 proposed values SHOULD be
   reported. The IPDV results SHOULD be reported in a table with a row
   for each of the tested frame sizes. The columns SHOULD include the
   frame size and associated frame rate for the tested media types and
   sub-columns for the three proposed reported values. Following the
   recommendations of [RFC5481], the RECOMMENDED units of measurement
   are milliseconds.

7.4. Frame Loss Rate

Use Section 26.3 of [RFC2544] unmodified.

7.5. Back-to-back Frames

Use Section 26.4 of [RFC2544] unmodified.

7.6. System Recovery

Use Section 26.5 of [RFC2544] unmodified.

7.7. Reset

Use Section 4 of [RFC6201] unmodified.

8. Additional Benchmarking Tests for Stateful IPv6 Transition Technologies

This section describes additional tests dedicated to the stateful IPv6 transition technologies. For the tests described in this section, the DUT devices SHOULD follow the test setup and test parameters recommendations presented in [RFC3511] (Sections 5.2 and 5.3)

The following additional tests SHOULD be performed.

8.1. Concurrent TCP Connection Capacity

Use Section 5.2 of [RFC3511] unmodified.

8.2. Maximum TCP Connection Establishment Rate

Use Section 5.3 of RFC3511 unmodified.

9. DNS Resolution Performance

This section describes benchmarking tests dedicated to DNS64 (see [RFC6147]), used as DNS support for single translation technologies such as NAT64.

9.1. Test and Traffic Setup

The test setup in Figure 3 follows the setup proposed for single translation IPv6 transition technologies in Figure 1.
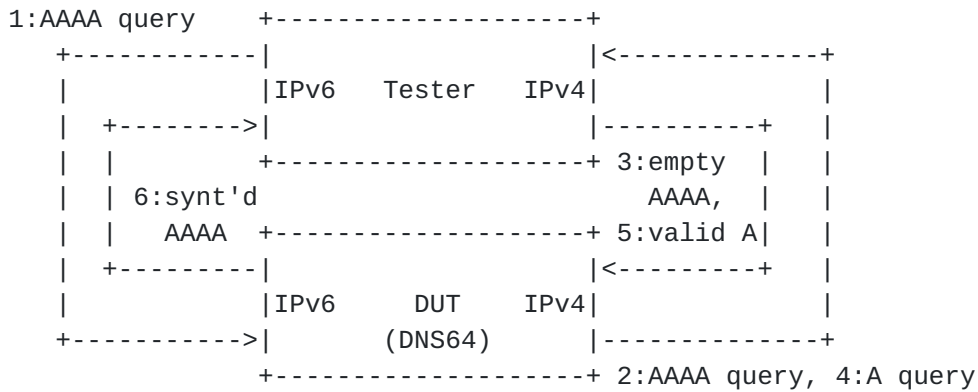
```
   1:AAAA query    +-------------------+
     +-----------|                   |<-------------+
     |           |IPv6   Tester   IPv4|             |
     |  +-------->|                   |----------+  |
     |  |         +-------------------+ 3:empty  |  |
     |  | 6:synt'd                     AAAA,     |  |
     |  |   AAAA  +-------------------+ 5:valid A|  |
     |  +---------|                   |<--------+   |
     |           |IPv6    DUT    IPv4|             |
     +----------->|       (DNS64)     |-------------+
                 +-------------------+ 2:AAAA query, 4:A query
                Figure 3. DNS64 test setup
```

The test traffic SHOULD follow the following steps.

1. Query for the AAAA record of a domain name (from client to DNS64 server)

2. Query for the AAAA record of the same domain name (from DNS64 server to authoritative DNS server)

3. Empty AAAA record answer (from authoritative DNS server to DNS64 server)

4. Query for the A record of the same domain name (from DNS64 server to authoritative DNS server)

5. Valid A record answer (from authoritative DNS server to DNS64 server)

6. Synthesized AAAA record answer (from DNS64 server to client)

The Tester plays the role of DNS client as well as authoritative DNS server. It MAY be realized as a single physical device, or alternatively, two physical devices MAY be used.

Please note that:

   - If the DNS64 server implements caching and there is a cache
     hit, then step 1 is followed by step 6 (and steps 2 through 5
     are omitted).
   - If the domain name has an AAAA record, then it is returned in
     step 3 by the authoritative DNS server; steps 4 and 5 are
     omitted, and the DNS64 server does not synthesizes an AAAA
     record, but returns the received AAAA record to the client.

      - As for the IP version used between the tester and the DUT, IPv6
        MUST be used between the client and the DNS64 server (as a
        DNS64 server provides service for an IPv6-only client), but
        either IPv4 or IPv6 MAY be used between the DNS64 server and
        the authoritative DNS server.

9.2. Benchmarking DNS Resolution Performance

   Objective: To determine DNS64 performance by means of the maximum
   number of successfully processed DNS requests per second.

   Procedure: Send a specific number of DNS queries at a specific rate
   to the DUT and then count the replies received in time (within a
   predefined timeout period from the sending time of the corresponding
   query, having the default value 1 second) and valid (contains an
   AAAA record) from the DUT. If the count of sent queries is equal to
   the count of received replies, the rate of the queries is raised and
   the test is rerun. If fewer replies are received than queries were
   sent, the rate of the queries is reduced and the test is rerun. The
   duration of each trial SHOULD be at least 60 seconds. This will
   reduce the potential gain of a DNS64 server, which is able to
   exhibit higher performance by storing the requests and thus
   utilizing also the timeout time for answering them. For the same
   reason, no higher timeout time than 1 second SHOULD be used. For
   further considerations, see [Lencse1].

   The maximum number of processed DNS queries per second is the
   fastest rate at which the count of DNS replies sent by the DUT is
   equal to the number of DNS queries sent to it by the test equipment.

   The test SHOULD be repeated at least 20 times and the median and 1st
   /99th percentiles of the number of processed DNS queries per second
   SHOULD be calculated.

   Details and parameters:

   1. Caching
   First, all the DNS queries MUST contain different domain names (or
   domain names MUST NOT be repeated before the cache of the DUT is
   exhausted). Then new tests MAY be executed with domain names, 20%,
   40%, 60%, 80% and 100% of which are cached. We note that ensuring a
   record being cached requires repeating it both "late enough" after
   the first query to be already resolved and be present in the cache
   and "early enough" to be still present in the cache.

   2. Existence of AAAA record
   First, all the DNS queries MUST contain domain names which do not
   have an AAAA record and have exactly one A record.

   Then new tests MAY be executed with domain names, 20%, 40%, 60%, 80%
   and 100% of which have an AAAA record.

   Please note that the two conditions above are orthogonal, thus all
   their combinations are possible and MAY be tested. The testing with
   0% cached domain names and with 0% existing AAAA record is REQUIRED
   and the other combinations are OPTIONAL. (When all the domain names
   are cached, then the results do not depend on what percentage of the
   domain names have AAAA records, thus these combinations are not
   worth testing one by one.)

   Reporting format: The primary result of the DNS64 test is the median
   of the number of processed DNS queries per second measured with the
   above mentioned "0% + 0% combination". The median SHOULD be
   complemented with the 1st and 99th percentiles to show the stability
   of the result. If optional tests are done, the median and the 1st
   and 99th percentiles MAY be presented in a two dimensional table
   where the dimensions are the proportion of the repeated domain names
   and the proportion of the DNS names having AAAA records. The two
   table headings SHOULD contain these percentage values.
   Alternatively, the results MAY be presented as the corresponding two
   dimensional graph, too. In this case the graph SHOULD show the
   median values with the percentiles as error bars. From both the
   table and the graph, one dimensional excerpts MAY be made at any
   given fixed percentage value of the other dimension. In this case,
   the fixed value MUST be given together with a one dimensional table
   or graph.

9.2.1. Requirements for the Tester

   Before a Tester can be used for testing a DUT at rate r queries per
   second with t seconds timeout, it MUST perform a self-test in order
   to exclude the possibility that the poor performance of the Tester
   itself influences the results. For performing a self-test, the
   tester is looped back (leaving out DUT) and its authoritative DNS
   server subsystem is configured to be able to answer all the AAAA
   record queries. For passing the self-test, the Tester SHOULD be able
   to answer AAAA record queries at 2*(r+delta) rate within 0.25*t
   timeout, where the value of delta is at least 0.1.

   Explanation: When performing DNS64 testing, each AAAA record query
   may result in at most two queries sent by the DUT, the first one of
   them is for an AAAA record and the second one is for an A record
   (the are both sent when there is no cache hit and also no AAAA
   record exists). The parameters above guarantee that the
   authoritative DNS server subsystem of the DUT is able to answer the
   queries at the required frequency using up not more than the half of
   the timeout time.

   Remark: a sample open-source test program, dns64perf++, is available
   from [Dns64perf] and it is documented in [Lencse2]. It implements
   only the client part of the Tester and it should be used together
   with an authoritative DNS server implementation, e.g. BIND, NSD or
   YADIFA. Its experimental extension for testing caching is available
   from [Lencse3] and it is documented in [Lencse4].

10. Overload Scalability

   Scalability has been often discussed; however, in the context of
   network devices, a formal definition or a measurement method has not
   yet been proposed. In this context, we can define overload
   scalability as the ability of each transition technology to
   accommodate network growth. Poor scalability usually leads to poor
   performance. Considering this, overload scalability can be measured
   by quantifying the network performance degradation associated with
   an increased number of network flows.

   The following subsections describe how the test setups can be
   modified to create network growth and how the associated performance
   degradation can be quantified.

10.1. Test Setup

   The test setups defined in Section 3 have to be modified to create
   network growth.

10.1.1. Single Translation Transition Technologies

   In the case of single translation transition technologies the
   network growth can be generated by increasing the number of network
   flows generated by the tester machine (see Figure 4).

```
                    +------------------------+
        +-----------|NF1                  NF1|<-------------+
        |  +--------|NF2      tester      NF2|<----------+  |
        |  |    ...|                         |           |  |
        |  |   +-----|NFn                  NFn|<------+   |  |
        |  |   |     +------------------------+       |   |  |
        |  |   |                                      |   |  |
        |  |   |     +------------------------+       |   |  |
        |  |   +---->|NFn                  NFn|-------+   |  |
        |  |     ...|           DUT           |           |  |
        |  +------->|NF2    (translator)   NF2|-----------+  |
        +---------->|NF1                  NF1|--------------+
                    +------------------------+
                    Figure 4. Test setup 3
```
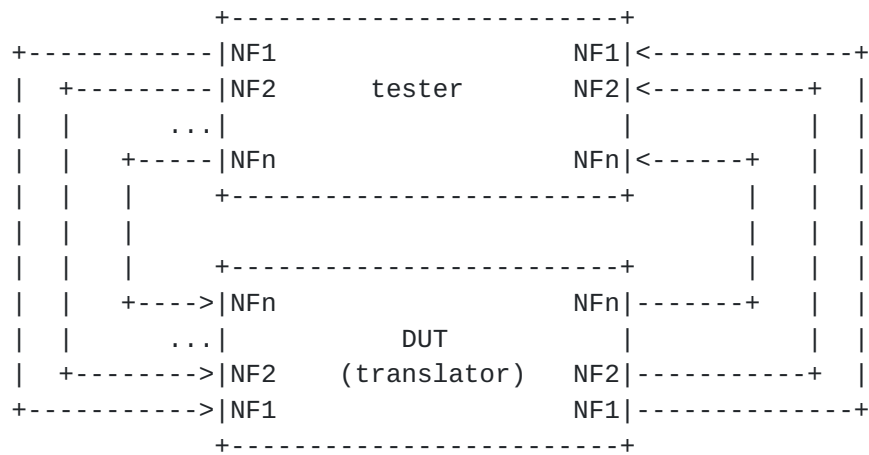
10.1.2. **Encapsulation/Double Translation Transition Technologies**

Similarly, for the encapsulation/double translation technologies a
multi-flow setup is recommended. Considering a multipoint-to-point
scenario, for most transition technologies, one of the edge nodes is
designed to support more than one connecting devices. Hence, the
recommended test setup is a n:1 design, where n is the number of
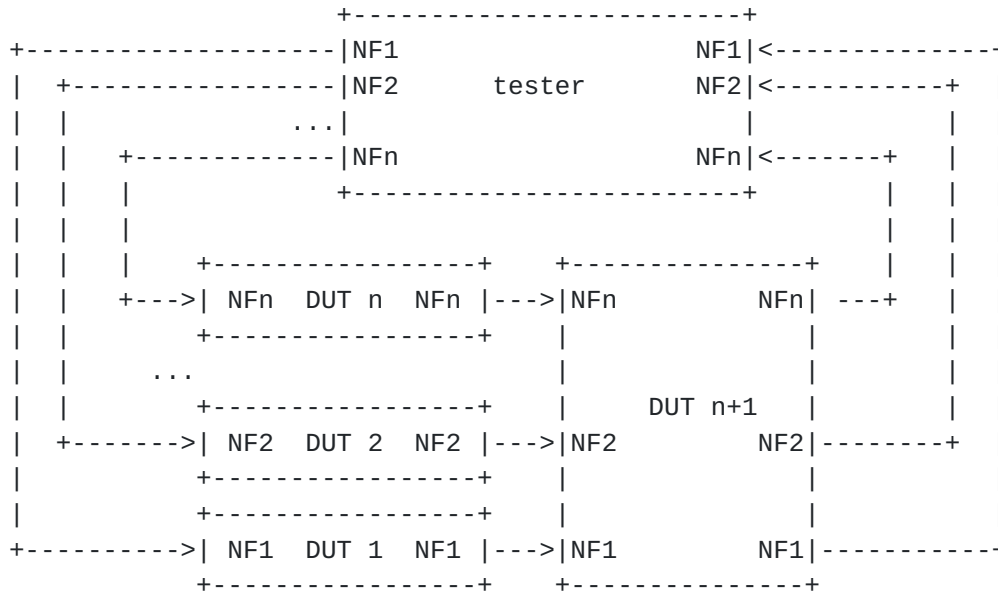client DUTs connected to the same server DUT (See Figure 5).

```
                           +------------------------+
    +--------------------|NF1                     NF1|<--------------+
    |   +----------------|NF2       tester        NF2|<-----------+  |
    |   |           ...|                            |            |  |
    |   |   +------------|NFn                     NFn|<-------+   |  |
    |   |   |           +------------------------+           |   |  |
    |   |   |                                                |   |  |
    |   |   |   +----------------+   +--------------+        |   |  |
    |   |   +--->| NFn   DUT n  NFn |--->|NFn         NFn| ---+   |  |
    |   |        +----------------+   |              |            |  |
    |   |     ...                     |              |            |  |
    |   |        +----------------+   |    DUT n+1   |            |  |
    |   +------->| NF2   DUT 2  NF2 |--->|NF2         NF2|--------+  |
    |            +----------------+   |              |               |
    |            +----------------+   |              |               |
    +---------->| NF1   DUT 1  NF1 |--->|NF1         NF1|-----------+
                +----------------+   +--------------+
                  Figure 5. Test setup 4
```

This test setup can help to quantify the scalability of the server
device. However, for testing the overload scalability of the client
DUTs additional recommendations are needed.
For encapsulation transition technologies, a m:n setup can be
created, where m is the number of flows applied to the same client
device and n the number of client devices connected to the same
server device.
For the translation based transition technologies, the client
devices can be separately tested with n network flows using the test
setup presented in Figure 4.

10.2. **Benchmarking Performance Degradation**

10.2.1. **Network performance degradation with simultaneous load**

Objective: To quantify the performance degradation introduced by n
parallel and simultaneous network flows.

Procedure: First, the benchmarking tests presented in Section 7 have
to be performed for one network flow.

   The same tests have to be repeated for n network flows, where the
   network flows are started simultaneously. The performance
   degradation of the X benchmarking dimension SHOULD be calculated as
   relative performance change between the 1-flow (single flow) results
   and the n-flow results, using the following formula:

            Xn - X1
       Xpd= ----------- * 100, where: X1 - result for 1-flow
               X1   Xn - result for n-flows

   This formula SHOULD be applied only for lower is better benchmarks
   (e.g. latency).
   For higher is better benchmarks (e.g. throughput), the following
   formula is RECOMMENDED.

            X1 - Xn
       Xpd= ----------- * 100, where: X1 - result for 1-flow
               X1   Xn - result for n-flows

   As a guideline for the maximum number of flows n, the value can be
   deduced by measuring the Concurrent TCP Connection Capacity as
   described by [RFC3511], following the test setups specified by
   Section 4.

   Reporting Format: The performance degradation SHOULD be expressed as
   a percentage. The number of tested parallel flows n MUST be clearly
   specified. For each of the performed benchmarking tests, there
   SHOULD be a table containing a column for each frame size. The table
   SHOULD also state the applied frame rate. In the case of benchmarks
   for which more than one value is reported (e.g. IPDV Section 7.3.2),
   a column for each of the values SHOULD be included.

10.2.2. Network performance degradation with incremental load

   Objective: To quantify the performance degradation introduced by n
   parallel and incrementally started network flows.

   Procedure: First, the benchmarking tests presented in Section 7 have
   to be performed for one network flow.

   The same tests have to be repeated for n network flows, where the
   network flows are started incrementally in succession, each after
   time t. In other words, if flow i is started at time x, flow i+1
   will be started at time x+t. Considering the time t, the time
   duration of each iteration must be extended with the time necessary
   to start all the flows, namely (n-1)xt. The measurement for the
   first flow SHOULD be at least 60 seconds in duration.

The performance degradation of the x benchmarking dimension SHOULD be calculated as relative performance change between the 1-flow results and the n-flow results, using the formula presented in Section 10.2.1. Intermediary degradation points for 1/4*n, 1/2*n and 3/4*n MAY also be performed.

Reporting Format: The performance degradation SHOULD be expressed as a percentage. The number of tested parallel flows n MUST be clearly specified. For each of the performed benchmarking tests, there SHOULD be a table containing a column for each frame size. The table SHOULD also state the applied frame rate and time duration T, used as increment step between the network flows. The units of measurement for T SHOULD be seconds. A column for the intermediary degradation points MAY also be displayed. In the case of benchmarks for which more than one value is reported (e.g. IPDV Section 7.3.2), a column for each of the values SHOULD be included.

11. NAT44 and NAT66

Although these technologies are not the primary scope of this document, the benchmarking methodology associated with single translation technologies as defined in Section 4.1 can be employed to benchmark NAT44 (as defined by [RFC2663] with the behavior described by [RFC7857]) implementations and NAT66 (as defined by [RFC6296]) implementations.

12. Summarizing function and variation

To ensure the stability of the benchmarking scores obtained using the tests presented in Sections 7 through 9, multiple test iterations are RECOMMENDED. Using a summarizing function (or measure of central tendency) can be a simple and effective way to compare the results obtained across different iterations. However, over-summarization is an unwanted effect of reporting a single number.

Measuring the variation (dispersion index) can be used to counter the over-summarization effect. Empirical data obtained following the proposed methodology can also offer insights on which summarizing function would fit better.

To that end, data presented in [ietf95pres] indicate the median as suitable summarizing function and the 1st and 99th percentiles as variation measures for DNS Resolution Performance and PDV. The median and percentile calculation functions SHOULD follow the recommendations of [RFC2330] Section 11.3.

   For a fine grained analysis of the frequency distribution of the
   data, histograms or cumulative distribution function plots can be
   employed.

13. Security Considerations

   Benchmarking activities as described in this memo are limited to
   technology characterization using controlled stimuli in a laboratory
   environment, with dedicated address space and the constraints
   specified in the sections above.

   The benchmarking network topology will be an independent test setup
   and MUST NOT be connected to devices that may forward the test
   traffic into a production network, or misroute traffic to the test
   management network.

   Further, benchmarking is performed on a "black-box" basis, relying
   solely on measurements observable external to the DUT/SUT. Special
   capabilities SHOULD NOT exist in the DUT/SUT specifically for
   benchmarking purposes. Any implications for network security arising
   from the DUT/SUT SHOULD be identical in the lab and in production
   networks.

14. IANA Considerations

   The IANA has allocated the prefix 2001:2::/48 [RFC5180] for IPv6
   benchmarking. For IPv4 benchmarking, the 198.18.0.0/15 prefix was
   reserved, as described in [RFC6890]. The two ranges are sufficient
   for benchmarking IPv6 transition technologies. Thus, no action is
   requested.

15. References

15.1. Normative References

   [RFC1242] Bradner, S., "Benchmarking Terminology for Network
             Interconnection Devices", RFC 1242, DOI 10.17487/RFC1242,
             July 1991, <http://www.rfc-editor.org/info/rfc1242>.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
             Requirement Levels", BCP 14, RFC 2119, DOI
             10.17487/RFC2119, March 1997, <http://www.rfc-
             editor.org/info/rfc2119>.

   [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis,
             "Framework for IP performance metrics", RFC 2330, DOI
             10.17487/RFC2330, May 1998, <http://www.rfc-
             editor.org/info/rfc2330>.

    [RFC2544] Bradner, S., and J. McQuaid, "Benchmarking Methodology for
              Network Interconnect Devices", RFC 2544, DOI
              10.17487/RFC2544, March 1999, <http://www.rfc-
              editor.org/info/rfc2544>.

    [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation
              Metric for IP Performance Metrics (IPPM)", RFC 3393, DOI
              10.17487/RFC3393, November 2002, <http://www.rfc-
              editor.org/info/rfc3393>.

    [RFC3511] Hickman, B., Newman, D., Tadjudin, S. and T. Martin,
              "Benchmarking Methodology for Firewall Performance", RFC
              3511, DOI 10.17487/RFC3511, April 2003, <http://www.rfc-
              editor.org/info/rfc3511>.

    [RFC5180] Popoviciu, C., Hamza, A., Van de Velde, G., and D.
              Dugatkin, "IPv6 Benchmarking Methodology for Network
              Interconnect Devices", RFC 5180, DOI 10.17487/RFC5180, May
              2008, <http://www.rfc-editor.org/info/rfc5180>.

    [RFC5481] Morton, A., and B. Claise, "Packet Delay Variation
              Applicability Statement", RFC 5481, DOI 10.17487/RFC5481,
              March 2009, <http://www.rfc-editor.org/info/rfc5481>.

    [RFC6201] Asati, R., Pignataro, C., Calabria, F. and C. Olvera,
              "Device Reset Characterization ", RFC 6201, DOI
              10.17487/RFC6201, March 2011, <http://www.rfc-
              editor.org/info/rfc6201>.

15.2. Informative References

    [RFC2663] Srisuresh, P., and M. Holdrege. "IP Network Address
              Translator (NAT) Terminology and Considerations", RFC2663,
              DOI 10.17487/RFC2663, August 1999, <http://www.rfc-
              editor.org/info/rfc2663>.

    [RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms
              for IPv6 Hosts and Routers", RFC 4213, DOI
              10.17487/RFC4213, October 2005, <http://www.rfc-
              editor.org/info/rfc4213>.

    [RFC4659] De Clercq, J., Ooms, D., Carugi, M., and F. Le Faucheur,
              "BGP-MPLS IP Virtual Private Network (VPN) Extension for
              IPv6 VPN", RFC 4659, September 2006, <http://www.rfc-
              editor.org/info/4659>.

   [RFC4798] De Clercq, J., Ooms, D., Prevost, S., and F. Le Faucheur,
              "Connecting IPv6 Islands over IPv4 MPLS Using IPv6
              Provider Edge Routers (6PE)", RFC 4798, February 2007,
              <http://www.rfc-editor.org/info/rfc4798>

   [RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4
              Infrastructures (6rd)", RFC 5569, DOI 10.17487/RFC5569,
              January 2010, <http://www.rfc-editor.org/info/rfc5569>.

   [RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for
              IPv4/IPv6 Translation", RFC 6144, DOI 10.17487/RFC6144,
              April 2011, <http://www.rfc-editor.org/info/rfc6144>.

   [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful
              NAT64: Network Address and Protocol Translation from IPv6
              Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146,
              April 2011, <http://www.rfc-editor.org/info/rfc6146>.

   [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van
              Beijnum, "DNS64: DNS Extensions for Network Address
              Translation from IPv6 Clients to IPv4 Servers", RFC 6147,
              DOI 10.17487/RFC6147, April 2011, <http://www.rfc-
              editor.org/info/rfc6147>.

   [RFC6219] Li, X., Bao, C., Chen, M., Zhang, H., and J. Wu, "The
              China Education and Research Network (CERNET) IVI
              Translation Design and Deployment for the IPv4/IPv6
              Coexistence and Transition", RFC 6219, DOI
              10.17487/RFC6219, May 2011, <http://www.rfc-
              editor.org/info/rfc6219>.

   [RFC6296] Wasserman, M., and F. Baker. "IPv6-to-IPv6 network prefix
              translation." RFC6296, DOI 10.17487/RFC6296, June 2011.

   [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-
              Stack Lite Broadband Deployments Following IPv4
              Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011,
              <http://www.rfc-editor.org/info/rfc6333>.

   [RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT:
              Combination of Stateful and Stateless Translation", RFC
              6877, DOI 10.17487/RFC6877, April 2013, <http://www.rfc-
              editor.org/info/rfc6877>.

   [RFC6890] Cotton, M., Vegoda, L., Bonica, R., and B. Haberman,
              "Special-Purpose IP Address Registries", BCP 153, RFC6890,
              DOI 10.17487/RFC6890, April 2013, <http://www.rfc-
              editor.org/info/rfc6890>.

   [RFC7596] Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I.
             Farrer, "Lightweight 4over6: An Extension to the Dual-
             Stack Lite Architecture", RFC 7596, DOI 10.17487/RFC7596,
             July 2015, <http://www.rfc-editor.org/info/rfc7596>.

   [RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S.,
             Murakami, T., and T. Taylor, Ed., "Mapping of Address and
             Port with Encapsulation (MAP-E)", RFC 7597, DOI
             10.17487/RFC7597, July 2015, <http://www.rfc-
             editor.org/info/rfc7597>.

   [RFC7599] Li, X., Bao, C., Dec, W., Ed., Troan, O., Matsushima, S.,
             and T. Murakami, "Mapping of Address and Port using
             Translation (MAP-T)", RFC 7599, DOI 10.17487/RFC7599, July
             2015, <http://www.rfc-editor.org/info/rfc7599>.

   [RFC7857] Penno, R., Perreault, S., Boucadair, M., Sivakumar, S.,
             and K. Naito "Updates to Network Address Translation (NAT)
             Behavioral Requirements" RFC 7857, DOI 10.17487/RFC7857,
             April 2016, <http://www.rfc-editor.org/info/rfc7857>.

   [RFC7915] LBao, C., Li, X., Baker, F., Anderson, T., and F. Gont,
             "IP/ICMP Translation Algorithm", RFC 7915, DOI
             10.17487/RFC7915, June 2016, <http://www.rfc-
             editor.org/info/rfc7915>.

   [Dns64perf] Bakai, D., "A C++11 DNS64 performance tester",
             available: https://github.com/bakaid/dns64perfpp

   [ietf95pres] Georgescu, M., "Benchmarking Methodology for IPv6
             Transition Technologies", IETF 95, Buenos Aires,
             Argentina, April 2016, available:
             https://www.ietf.org/proceedings/95/slides/slides-95-bmwg-
             2.pdf

   [Lencse1] Lencse, G., Georgescu, M. and Y. Kadobayashi,
             "Benchmarking Methodology for DNS64 Servers", unpublished,
             revised version is available:
             http://www.hit.bme.hu/~lencse/publications/ECC-2017-B-M-
             DNS64-revised.pdf

   [Lencse2] Lencse, G., Bakai, D, "Design and Implementation of a Test
             Program for Benchmarking DNS64 Servers", IEICE
             Transactions on Communications, to be published (vol.
             E100-B, no. 6. pp. -, June 2017.), advance publication is
             available: http://doi.org/10.1587/transcom.2016EBN0007
             revised version is freely available:
             http://www.hit.bme.hu/~lencse/publications/IEICE-2016-
             dns64perfpp-revised.pdf

   [Lencse3] http://www.hit.bme.hu/~lencse/dns64perfppc/

   [Lencse4] Lencse, G., "Enabling Dns64perf++ for Benchmarking the
             Caching Performance of DNS64 Servers", unpublished, review
             version is available:
             http://www.hit.bme.hu/~lencse/publications/IEICE-2016-
             dns64perfppc-for-review.pdf

16. Acknowledgements

Appendix A.   **Theoretical Maximum Frame Rates**

   This appendix describes the recommended calculation formulas for the
   theoretical maximum frame rates to be employed over Ethernet as
   example media. The formula takes into account the frame size
   overhead created by the encapsulation or the translation process.
   For example, the 6in4 encapsulation described in [RFC4213] adds 20
   bytes of overhead to each frame.

   Considering X to be the frame size and O to be the frame size
   overhead created by the encapsulation on translation process, the
   maximum theoretical frame rate for Ethernet can be calculated using
   the following formula:

$$\frac{\text{Line Rate (bps)}}{(8\text{bits/byte})*(X+O+20)\text{bytes/frame}}$$

   The calculation is based on the formula recommended by RFC5180 in
   Appendix A1. As an example, the frame rate recommended for testing a
   6in4 implementation over 10Mb/s Ethernet with 64 bytes frames is:

$$\frac{10,000,000(\text{bps})}{(8\text{bits/byte})*(64+20+20)\text{bytes/frame}} = 12,019 \text{ fps}$$

   The complete list of recommended frame rates for 6in4 encapsulation
   can be found in the following table:

Frame size (bytes)	10 Mb/s (fps)	100 Mb/s (fps)	1000 Mb/s (fps)	10000 Mb/s (fps)
64	12,019	120,192	1,201,923	12,019,231
128	7,440	74,405	744,048	7,440,476
256	4,223	42,230	422,297	4,222,973
512	2,264	22,645	226,449	2,264,493
678	1,740	17,409	174,094	1,740,947
1024	1,175	11,748	117,481	1,174,812
1280	947	9,470	94,697	946,970
1518	802	8,023	80,231	802,311
1522	800	8,003	80,026	800,256
2048	599	5,987	59,866	598,659
4096	302	3,022	30,222	302,224
8192	152	1,518	15,185	151,846
9216	135	1,350	13,505	135,048

Authors' Addresses

   Marius Georgescu
   RCS&RDS
   Strada Dr. Nicolae D. Staicovici 71-75
   Bucharest 030167
   Romania

   Phone: +40 31 005 0979
   Email: marius.georgescu@rcs-rds.ro

   Liviu Pislaru
   RCS&RDS
   Strada Dr. Nicolae D. Staicovici 71-75
   Bucharest 030167
   Romania

   Phone: +40 31 005 0979
   Email: liviu.pislaru@rcs-rds.ro

   Gabor Lencse
   Szechenyi Istvan University
   Egyetem ter 1.
   Gyor
   Hungary

   Phone: +36 20 775 8267
   Email: lencse@sze.hu