Description
3. Measuring Internet access service quality
The aim of this chapter is to specify the measurement methodology best practices with the combined goal of maximising measurement accuracy and to ensure that the measurement results are comparable between
Results of these measurements can be also used for the following purposes:
Empowering the end user to validate the commitments made to them from their IAS provider.
Monitoring the general IAS quality and confirming that the performance of IAS is developing sufficiently over time when taking into account technological evolution.
To support the detection of any prioritisation and/or throttling of selected applications compared to other applications running over IAS.
NRAs may also use the data to increase transparency (e.g. interactive maps showing performance in a
According to BEREC NN guidelines [3] paragraph 166, speed should be calculated “based on IP packet payload, e.g. using TCP as transport layer protocol” and according to the NN guidelines paragraph 140, an ISP should define the speed on the basis of the IP packet payload or transport layer protocol payload.
This methodology is targeted to measure IAS quality in both the upload and download direction. It is worth noting that IAS speed is just one component of the performance experienced by the end users, since different applications have different protocol overheads and different requirements related to IAS delay,
For both measurement tasks - IAS as a whole and individual applications using IAS - the fundamental precondition is that measurements are performed at the edge of the network which provides the IAS (i.e. end user premises for fixed access or via the radio access for Mobile IAS).
Where measurements are performed against a test server, this server should be located outside the IAS network. It should have adequate connectivity between the server and the IAS provider to avoid influencing measurements. In general it is recommended that the measurement server should be located at the national Internet exchange point (IXP) unless there is specific reason for its placement elsewhere.
Monitoring mechanisms should mitigate, to the extent possible, confounding factors which are internal to the user environment. Examples of these factors include existing cross-traffic and the usage of Wi-Fi based interfaces. This topic is discussed separately in chapter 5.

The assessment of measurement results is discussed further in chapter 6 and the certified monitoring mechanism is further discussed in chapter 7.

3.1 IAS speed measurements

3.1.1 Speed measurement overall methodology

To maximise compatibility in a real-world environment, it is recommended to measure upload/download speeds based on the time to execute a set of controlled file transfers over HTTP.

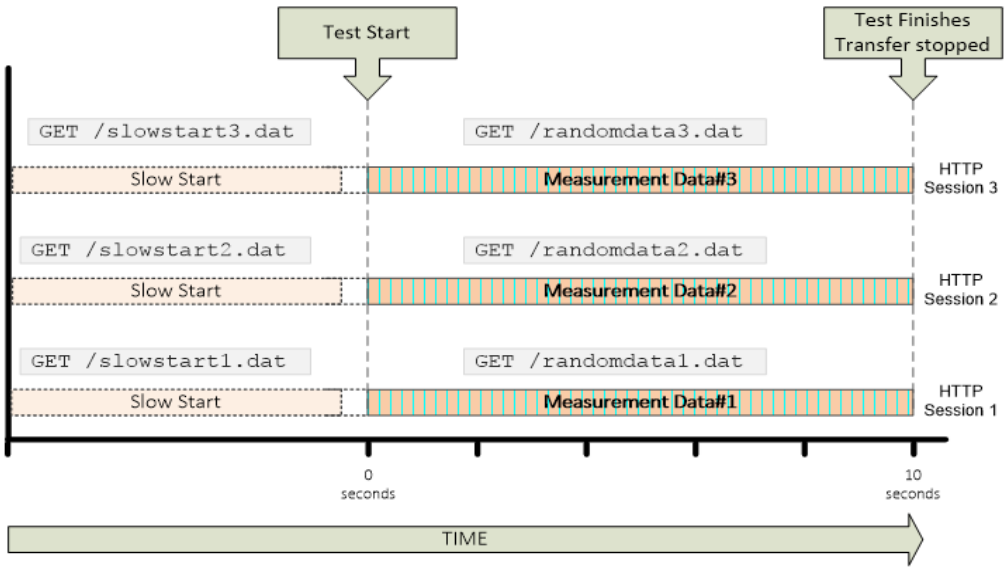
This methodology is supported by the broadest range of platforms, and can be implemented within a web browser or within the restricted sandbox of an on-device app. As such it is the best compromise between the competing demands of accuracy, platform agnosticism, ease of implementation and transparency.

Another reason to recommend the use of HTTP is to mitigate any firewall based restrictions which may result from the choice of a less commonly used protocol/port. The use of HTTPS also prevents

In order to saturate the path, it is recommended to use 3-5 HTTP connections. Furthermore, these connections should all have completed the TCP slow start phase to maximise throughput and ensure that the measurement is as representative as possible. The test is stopped after a pre-defined interval and the transfer speed is calculated by the recipient based on the data transferred over that interval.

BEREC recognises that packet loss and packet retransmission has a negative impact on the throughput of each TCP connection and hence the IAS speed.

The following diagram illustrates a download test based on 3 HTTP connections.



The following points should be noted in relation to the diagram above:

3 persistent HTTP connections are started at the same time;

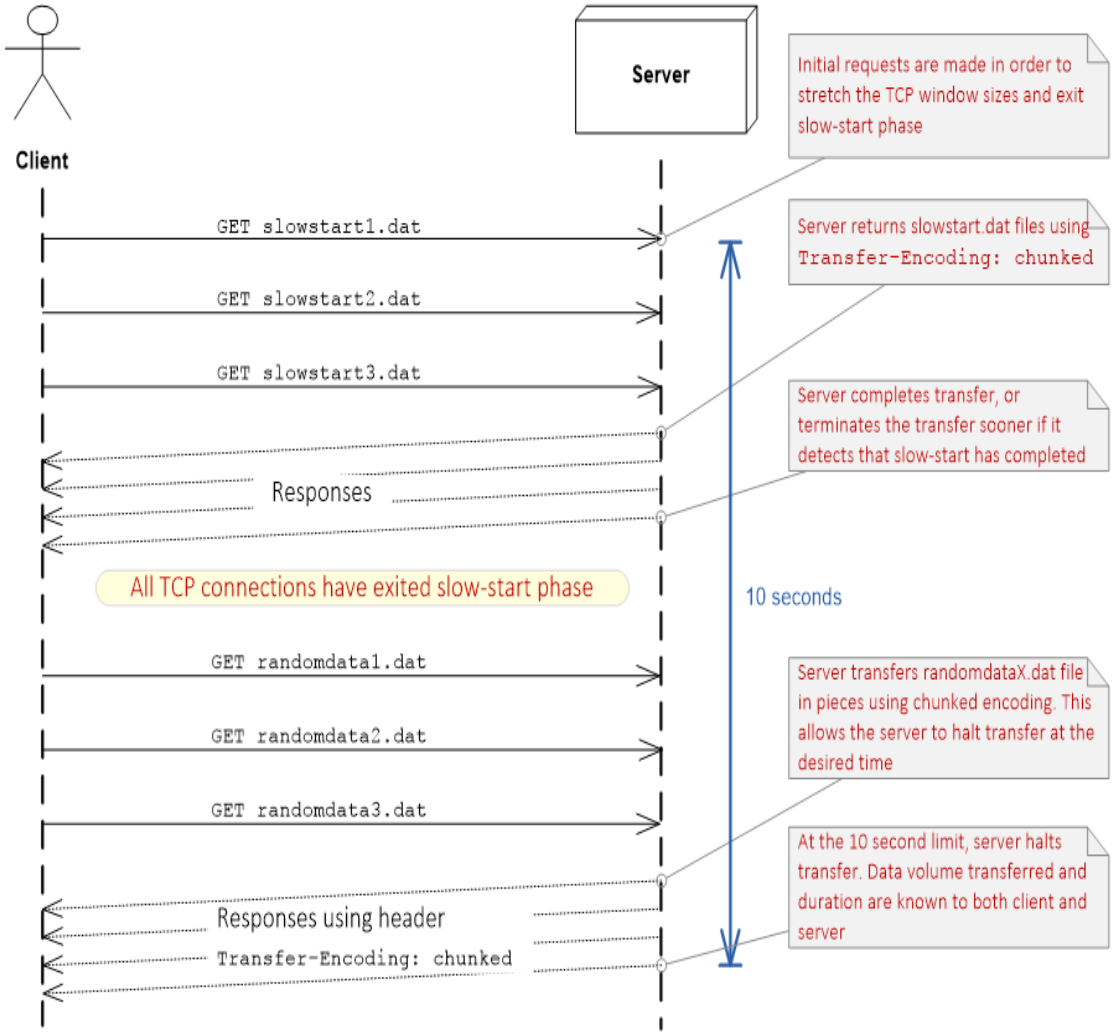
The duration of the test in this example is 10 seconds, however a longer fixed duration could be

To mitigate the effect of TCP slow-start, an initial retrieval of a slowstartX.dat file is made to maximise subsequent throughput; once the transfer of slowstartX.dat file is finished for each connection, the real test commences (using same TCP socket) without delay; all .dat files referred to above contain random data, which cannot be compressed and the test is stopped after a total of 10 seconds, and the valid measured upload/download volume is based on the total transferred volumes in Measurement Data#1,

It is recommended that the HTTP transfers are made using chunked transfer encoding to enable the sending side to stop the transfer at the appropriate time.

The diagram below shows the HTTP transfers in more detail in a message sequence chart format.

Diagram



The following sections discuss the ways to calculate the actual data transferred under this methodology.

3.1.2 Calculating speed based on TCP payload

Calculating the TCP payload is relatively straightforward as compared to calculating the IP Payload (see section 3.1.3). For a given HTTP connection, both the client and the server are mutually aware of the data volume transferred. This data volume will vary for each connection due to the recommendation that the measurement test duration is fixed.

Note that the amount of data transferred will also include the HTTP headers, so it is recommended that in cases where the exact size of these headers is not known, that a fixed 500 byte value is added to the total file size as an approximation.

The error introduced by this approximation is considered to be negligible except in cases where the test is run very briefly or on extremely slow links.

3.1.3 Calculating speed based on IP packet payload

Calculating speed based on IP packet payload is more complex due to the fact that most platforms don't allow clients to access this information directly, so it must be calculated based on assumptions, and the

Since the measurement client can only be guaranteed to know the TCP payload volume (i.e. the size of the file transferred), it would be necessary to calculate the number of packets required to transfer this TCP payload and then use this number to calculate the volume of TCP headers.

However the number of packets is a function of the TCP Maximum Segment Size (MSS), which is itself a function of the Maximum Transmission Unit size (MTU).

In addition to the above, the potential presence of TCP options introduces the possibility that the TCP header size is not fixed which further complicates the calculation.

The result of these factors is that it is impossible to accurately calculate the IP Payload volume from the TCP Payload volume. Therefore an adequate safety margin should be taken into account. Example calculations for the overhead are shown in the following tables for various sample values of MTU and TCP header size for

IPv4 (no IP options)	MTU	
Average TCP Header Size	**1500 Octets**	**1280 Octets**
20 Octets (no TCP options)	**1.37%**	1.61%
40 Octets (Average 50% of max)	**2.78%**	3.28%
60 Octets (Max TCP options)	4.23%	5.00%

IPv6 (no IP options)	MTU	
Average TCP Header Size	**1500 Octets**	**1280 Octets**
20 Octets (no TCP options)	**1.39%**	1.64%
40 Octets (Average 50% of max)	**2.82%**	3.34%
60 Octets (Max TCP options)	4.29%	5.09%

Note that these tables are intended to provide an illustration of the potential impact of these variables; however it's expected that in practise the MTU will generally be very close to 1500 octets and the average TCP header size will be close to 20 bytes. Only the two percentage values highlighted in each table are

Therefore 3% TCP header overhead can be considered to include an adequate safety margin and it can be used in calculating the IP packet payload. The IP packet payload calculation is done by adding this 3 % value to the speed calculated based on TCP packet payload.

However it should be noted that BEREC considers that TCP payload volume is the most reliable one to use when calculating the upload/download speed.

3.1.4 Miscellaneous Details

It should be possible to run measurements both over IPv4 and IPv6.

Both download and upload speeds should be measured in the same manner and reported in bits/second (e.g. Kbit/s or Mbit/s). Note that conversion factors between mega and kilo shall be base-10 rather than base-2 (i.e. 1KB = 1000 Bytes rather than 1024 bytes)
TCP/HTTP characteristics and options
Where possible, it is recommended to mitigate the effect of the following inherent HTTP and TCP characteristics which could otherwise introduce error to speed measurements:
TCP connection speed limit: As the bandwidth of an individual TCP connection is limited to the bandwidth-delay product of the path in question, it is necessary to utilise multiple TCP connections in
HTTP considerations: Generally the use of HTTPS is recommended. When using plain HTTP, either the appropriate HTTP headers to prevent caching should be used, or unique URIs should be used.
3.2 Delay and delay variation measurements
In principle, any kind of IP packet could be used for latency measurements (e.g. ICMP, UDP or TCP). However the following considerations should be taken into consideration:
Operating systems normally require administrator (root) privileges for sending ICMP packets. Also ICMP packets are often blocked by firewalls and antivirus software and hence they cannot be relied upon.
TCP packets (after connection setup) are subject to flow control
In a web browser environment it is difficult or even impossible to setup UDP-based connections.
It is recommended that delay is measured using:
UDP with TCP as fall back option
at least 10 measurements
and calculated as an average of recorded round-trip time values (typically expressed in milliseconds)
The measurement setup should be insensitive to (user) clock changes during the measurement.
It is also recommended that the delay variation (jitter) is calculated as mean deviation based on the samples collected for the delay measurement.
Calculation shall be based on the algorithms used in the Linux ping utility which is based on 4.3BSD

Drawing
For example:
`--- qostest.eu ping statistics ---`
`10 packets transmitted, 10 received, 0% packet loss, time 9014ms`
`rtt min/avg/max/mdev = 5.317/5.442/5.727/0.121 ms`

3.3 Packet loss measurements
If a packet is not received back within a certain timeout (e.g. 3 seconds), it is considered as lost for packet
Because of packet loss on TCP connections, even the low level of packet loss observed in modern networks[1] can result in significant performance degradation. Therefore, it is evident that 10 or even 100 measurements may not yield packet loss accurately. It is therefore recommended to send a large number of IP packets (e.g. at least 1000). The number of IP packets should be based on access technology
Delay and packet loss measurements are typically performed over a longer period of time in order to allow for the time varying nature of network performance in packet-switched networks.
However, the principle of running measurements of long duration conflicts with the crowdsourced user-initiated measurement concept. End users will not accept an extended waiting time for the presentation
Conversely, short duration tests can only provide an indication of whether the measurement was done during stable network conditions; whereas long measurement intervals are preferred for the meaningful
While long duration tests are preferred for delay and packet loss measurements, this likely introduces the need for a measurement client running permanently in the background.

Compliancy
Noted
Noted. Audit related
Noted. Audit related
Noted. Audit related
Noted. Audit related
As per BoR14_117, The data transmission speed is defined as the data transmission rate that is achieved separately for downloading and uploading specified test files between a remote web site and a user's computer. MedUX solution has specific tests for uplink and downlink and it considers only the Payload at Application Layer. If Headers are required, they could be included
Noted. Upload and Download directions will be measured
For Fixed Solution, probes will be located at CPE For Mobile Solution, probes are Mobile devices simulating the Mobile User
As per Chapter 3 of Berec consultation and indicated on BoR14_117, Monitoring mechanisms should mitigate, to the extent possible, confounding factors such as cross-traffic. In this sense, Case solution: 1. Depending on the Technology and the Contracted Commercial Plan, MedUX builds a specific Tests Suite 2. During the tuning phase, MedUX launches the Tests Suite to verify the Busy and Off-Peak timeframes. Then, depending on the results and the Customer preferences, it can be possible to run two different Tests Suites: during the Busy hours a Busy Tests Suite is executed with with light traffic tests and during the Off-Peak hours a Off-Peak Tests Suite is run with intensive traffic tests sequences 3. In addition, Tests Suites are prepared in a data burst (high stress in a short period of time) manner with a time period (to be defined by the Customer) between Tests, to minimize to the extent possible the cross-traffic 4. By other side, tests can be done limited in time to have a representation of the results. In such way, a quality sample is achieved that is not preempting the Customer BW or any Service degradation 5. It must be highlighted that MedUX aim is to measure the Service in a 7*24 schedule, without any discrimination in Tests Suites, being the probe considered as an additional User of a specific CPE. If points 1 to 4 are followed, the probe will not affect more than other user at CPE With more than 2850 reusable probes installed, Case experience has demonstrated that less than 1% of the Customers have perceived any type of degradation. The added value of this method is that in a proactive way permits to Regulators and/or Operators to have such type amount of samples that will permit to characterize the real situation/technology/region/commercial packages or any other method as per the Customer needs It is important to mention that MedUX solution is based on Statistical representation of the Confidence Level, an error permitted and a p value. In this sense, a specific number of tests will need to be executed to

Noted. Audit related
HTTP testing target is to verify how long it takes to a file to be downloaded/uploaded to sense in an objective way, how the Customer Experience would be whereas downloading/uploading a file. SpeedTest purpose is to stress the channel in order to verify the Latency and Data Rate, to understand the channel capacity
http test is supported on HW probes. Speedtest is supported on both: HW probes and APP
Noted. https expected to be ready on 17Q04
MedUX-SpeedTest uses four threads over http using TCP during the download and upload portions of the test Number of threads is hardcoded, then, in order to modify it, it is required to estimate and plan the change in both the Code. Not included into the standard proposals
Noted
Noted
Noted
By default, default value is 20 seconds
Slow Start expected to be ready on 17Q04

HTTP transfers are made using chunked transfer encoding. In "chunked transfer encoding", the data are sent as a series of "chunks". The sender of chunked data must know the current chunk size but not the overall data size. Thus, dynamically generated content may be transmitted. The chunked keyword in the Transfer-Encoding header is used to indicate chunked transfer. Each chunk is preceded by its size. The transmission ends when a zero-length chunk is encountered. More details can be seen on: https://en.wikipedia.org/wiki/Chunked_transfer_encoding

As our test aim is to measure the CE, it is expected that network could support Customer Throughtput. For this reason, Speed is calculated based on Application payload only. Speed is obtained as follows:

* Threats are triggered for DL/UL
* Start Time is established

* Total number of DL/UL Bytes are calculated as part of the Request message. Headers are not taken into account for such calculation * Each 200 msec. the actual speed is calculated as well as the number of data Rx/Tx and the Total Time required to complete the action If required, headers can be added
Noted
Noted
Noted
Noted
Noted
Noted
Noted
Noted. If required, headers can be added
Partially Compliant. Current Case solution is based on ipV4, however, it is compliant with any ipV6 network, however, if there is a Market needs on End User/Applications for ipV6, a proper Tests Suites validation needs to be done to certify a proper working.

Functionality already developed in other tests. It can be included upon request
Compliant for http. https expected to be ready on 17Q04
MedUX PING test operates by sending Internet Control Message Protocol (ICMP) Echo Request packets to the target Host and waiting for an ICMP Response.
Noted. NA as own development
Noted. TCP Packets require tuning to properly work from OS & Application to establish a correct TCP session and to achieve the max. speed, mainly on those long distances where Windows need to be tuned
Noted as FW most possibly will reject the UDP packets
iperf can be added upon Request
The number of measurements is a Test Suite's parameter, it is configurable
Per probe, all Ping results are taken into consideration to estimate the Average of the measurement
As probes are not managed by End Users, measurement setup is not sensitive to any Clock changes during the measurement
Jitter: Standard Deviation of the Mean as Standard deviation Time of received packets (msec.). Standard DV is estimated as per ping command results and average of a bunch of samples

fping can cope with this requirement. It can be included upon request
The number of packets is a configurable parameter
The period of time is a configurable parameter
Noted
Noted
Noted